Clustering, Classification and the Multi–Facility Location Problem

- Adi Ben-Israel (Rutgers University)
Clustering, Classification and the Multi–Facility Location Problem

- Adi Ben-Israel (Rutgers University)
- Cem Iyigun (Middle East Technical University)
Clustering, Classification and the Multi-Facility Location Problem

- Adi Ben-Israel (Rutgers University)
- Cem Iyigun (Middle East Technical University)

INFORMS, Rutgers University, October 22, 2009
LOCATION
The Fermat–Weber location problem

Given \( \{x_i: i \in \{1,N\}\} \) and weights \( w_i > 0 \), find \( c \) minimizing

\[
f(c) = \sum_{i=1}^{N} w_i \|x_i - c\|, \quad \| \cdot \| \text{ Euclidean}.\]
The Fermat–Weber location problem

Given \( \{x_i : i \in \overline{1,N}\} \) and weights \( w_i > 0 \), find \( c \) minimizing

\[
f(c) = \sum_{i=1}^{N} w_i \|x_i - c\|, \quad \|\cdot\| \text{ Euclidean.}
\]

The \textit{gradient} (wherever existing)

\[
\nabla f(c) = - \sum_{i=1}^{N} w_i \frac{x_i - c}{\|x_i - c\|}
\]
The Fermat–Weber location problem

Given \( \{x_i : i \in \overline{1,N}\} \) and weights \( w_i > 0 \), find \( c \) minimizing

\[
f(c) = \sum_{i=1}^{N} w_i \|x_i - c\|, \quad \| \cdot \| \text{ Euclidean.}
\]

The gradient (wherever existing)

\[
\nabla f(c) = - \sum_{i=1}^{N} w_i \frac{x_i - c}{\|x_i - c\|}
\]

vanishes for

\[
c = \sum_{i=1}^{N} \lambda_i x_i, \quad \lambda_i = \frac{w_i}{\sum_{j=1}^{N} \frac{w_j}{\|x_j - c\|}}
\]

expressing \( c \) as a convex combination of \( \{x_i : i \in \overline{1,N}\} \).
A mechanical solution

The point $c$ is tied to the weights $w_i$ through holes in $x_i$. $c$ is free to move, will come to rest at the optimal center.
A mechanical solution

The point $\mathbf{c}$ may settle at one of the holes, say $\mathbf{c} = \mathbf{x}_1$, if the weight $w_1$ is greater than the resultant of the other weights.
The Varignon frame

Pierre Varignon (1654–1722)
The Weiszfeld method (1937)

Given $X = \{x_i : i \in \overline{1, N}\}$ and weights $w_i > 0$, solve

$$\min_c \sum_{i=1}^{N} w_i \|x_i - c\|$$
The Weiszfeld method (1937)

Given $X = \{x_i : i \in 1,N\}$ and weights $w_i > 0$, solve

$$\min_c \sum_{i=1}^N w_i \|x_i - c\|$$

The Weiszfeld method is the iteration

$$c_+ = T(c) := \begin{cases} \sum_{i=1}^N \left( \frac{w_i}{\sum_{j=1}^N \frac{\|x_i - c\|}{w_j}} \right) x_i , & c \notin X; \\ c , & c \in X. \end{cases}$$
The Weiszfeld method (1937)

Given \( X = \{x_i : i \in \{1, N\}\} \) and weights \( w_i > 0 \), solve

\[
\min_c \sum_{i=1}^{N} w_i \|x_i - c\|
\]

The **Weiszfeld method** is the iteration

\[
c_+ = T(c) := \begin{cases} 
\sum_{i=1}^{N} \left( \frac{w_i}{\sum_{j=1}^{N} \|x_j - c\|} \right) x_i, & c \notin X; \\
c, & c \in X.
\end{cases}
\]

E. Weiszfeld, Sur le point par lequel la somme des distances de n points donnés est minimum, *Tohoku Math. J.* **43**(1937) 355-386

Endre Weiszfeld, Andrew Vazsonyi (1916–2003)
Modified gradient $\nabla f(c) := -R(c)$

$X = \{x_1, \ldots, x_N\}$, $f(c) = \sum_{i=1}^{N} w_i \|x_i - c\|

(a) If $c \not\in X$, then $R(c) := -\nabla f(c)$

(b) If $c \in X$, say $c = x_j$, then

$$R(c) := \max \{0, \|R^j\| - w_j\} \frac{R^j}{\|R^j\|}$$

where $R^j := \sum_{i \neq j} \frac{w_i}{\|x_i - x_j\|} (x_i - x_j)$

the resultant force of $N - 1$ forces $w_i \frac{(x_i - x_j)}{\|x_i - x_j\|}$, $i \neq j$. 
Modified gradient $\nabla f(c) := -R(c)$

$X = \{x_1, \ldots, x_N\}$, $f(c) = \sum_{i=1}^{N} w_i \|x_i - c\|

(a) If $c \notin X$, then $R(c) := -\nabla f(c)$

(b) If $c \in X$, say $c = x_j$, then

$$R(c) := \max \{0, \|R^j\| - w_j\} \frac{R^j}{\|R^j\|}$$

where $R^j := \sum_{i \neq j} \frac{w_i}{\|x_i - x_j\|} (x_i - x_j)$

the resultant force of $N - 1$ forces $w_i \frac{(x_i - x_j)}{\|x_i - x_j\|}, i \neq j$.

Theorem: A center $c$ is optimal if and only if $R(c) = 0$. 
Convergence of $c := T(c)$ to optimal $c^*$

Kuhn (1973)

(a) $c = c^* \iff R(c) = 0$.

(b) $c^* \in \text{conv } X$ (the convex hull of $X$).

(c) $c = c^* \implies T(c) = c$. Conversely, $c \not\in X, T(c) = c \implies c = c^*$.

(d) $T(c) \neq c \implies f(T(c)) < f(c)$.

(e) $x^*_j = c^* \iff w_j \geq \|R^j\|$.

(f) If $x^*_j \neq c^*$, then direction of $-\nabla f(x^*_j)$ is $R^j/\|R^j\|$.

(g) If $x^*_j \neq c^*$ there exists $\delta > 0$ such that

$$0 < \|c - x^*_j\| \implies \|T^s(c) - x^*_j\| > \delta$$

for some $s$.

(h) $\lim_{c \to x^*_j} \frac{\|T(c) - x^*_j\|}{\|c - x^*_j\|} = \frac{\|R^j\|}{w_j}$.

(i) For any $c_0$, if all $T^r(c_0) \not\in X$, then $\lim_{r \to \infty} T^r(c_0) = c^*$.
A bad example for the Weiszfeld method

\[ x_i = (\pm 1, \pm 1), i \in \overline{1, 4}, \quad x_5 = (100, 0) \]

\[ w_i = 1, i \in \overline{1, 4}, \quad w_5 = 4 \]

\[ x_1 = (80.0, 0), \ldots, x_{11} = (80.0037, 0), \quad x_{12} = (80.00387991, 0), \ldots \]

\[ x_{17} = (80.00430047, 0), \quad x_{18} = (80.00436990, 0), \]

\[ x_{19} = (80.00443568, 0), \quad x_{20} = (80.00449821, 0), \ldots \]
A multi–facility location problem

Given \{x_i : i \in \overline{1,N}\} \subset \mathbb{R}^n,
weights \ w_i > 0, \ and
an integer \ K \geq 1,
find \{c_k : k \in \overline{1,K}\} \text{ minimizing}

\[ f(c_1, \ldots, c_K) = \sum_{k=1}^{K} \sum_{x_i \in C_k} w_i \|x_i - c_k\|, \]

where \ C_k = \{x : x \text{ assigned to } c_k\}. 
A multi–facility location problem

Given \( \{ \mathbf{x}_i : i \in \overline{1,N} \} \subset \mathbb{R}^n \),
weights \( w_i > 0 \), and
an integer \( K \geq 1 \),
find \( \{ \mathbf{c}_k : k \in \overline{1,K} \} \) minimizing

\[
    f(\mathbf{c}_1, \ldots, \mathbf{c}_K) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} w_i \| \mathbf{x}_i - \mathbf{c}_k \|,
\]

where \( \mathcal{C}_k = \{ \mathbf{x} : \mathbf{x} \text{ assigned to } \mathbf{c}_k \} \).

The problem is \textbf{NP hard}:

Location–assignment iterations

The multi–facility location problem

\[
\min f(c_1, \ldots, c_K) = \sum_{k=1}^{K} \sum_{x_i \in C_k} w_i \|x_i - c_k\|,
\]

can be solved iteratively as follows:

0. **Initialization**: Assign the \(N\) points \(\{x_i\}\) to \(K\) clusters \(C_k\)
Location–assignment iterations

The multi–facility location problem

\[ \min f(c_1, \ldots, c_K) = \sum_{k=1}^{K} \sum_{x_i \in C_k} w_i \| x_i - c_k \|, \]

can be solved iteratively as follows:

0. **Initialization**: Assign the \( N \) points \( \{x_i\} \) to \( K \) clusters \( C_k \)

1. **Location**: For each cluster \( C_k \) solve the Fermat–Weber problem

\[ \min_{c_k} \sum_{x_i \in C_k} w_i \| x_i - c_k \| \]
Location–assignment iterations

The multi–facility location problem

\[ \min f(c_1, \ldots, c_K) = \sum_{k=1}^{K} \sum_{x_i \in C_k} w_i \|x_i - c_k\|, \]

can be solved iteratively as follows:

0. Initialization: Assign the \( N \) points \( \{x_i\} \) to \( K \) clusters \( C_k \)

1. Location: For each cluster \( C_k \), solve the Fermat–Weber problem

\[ \min_{c_k} \sum_{x_i \in C_k} w_i \|x_i - c_k\| \]

2. Assignment: Assign each point to the cluster with nearest center, if different from its current cluster. Stop when done. Go to 1.
Possible to get stuck at a non–optimal ("local optimal") solution

\[ C_1 = \{x_1, x_4\}, \quad C_2 = \{x_2, x_3\} \]
\[ c_1 \in [x_1, x_4], \quad c_2 \in [x_2, x_3] \]

objective = 2

\[ C_1 = \{x_1, x_2, x_4\}, \quad C_2 = \{x_3\} \]
\[ c_1 = \left(\frac{1}{3+\sqrt{3}}, \frac{1}{3+\sqrt{3}}\right), \quad c_2 = x_3 \]

objective = 1.9318

The left solution is stable
CLUSTERING
Data
Clustering

Clustering = partitioning a set $\mathcal{D}$ into clusters, i.e. disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of similar elements, with elements of different clusters dissimilar.
Clustering

Clustering = partitioning a set $\mathcal{D}$ into clusters, i.e. disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of similar elements, with elements of different clusters dissimilar.

The number of clusters, $K$, is in general unknown.
Clustering

Clustering = partitioning a set $\mathcal{D}$ into clusters, i.e. disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of similar elements, with elements of different clusters dissimilar.

The number of clusters, $K$, is in general unknown.

Distance clustering: Similar means close in some metric sense. For example, $x \in \mathcal{C}_k$ if $d(x, \mathcal{C}_k) < d(x, \mathcal{C}_j)$ for $j \neq k$
Clustering

Clustering = partitioning a set \( D \) into clusters, i.e. disjoint subsets \( C_1, \ldots, C_K \) of similar elements, with elements of different clusters dissimilar.

The number of clusters, \( K \), is in general unknown.

Distance clustering: Similar means close in some metric sense. For example, \( x \in C_k \) if \( d(x, C_k) < d(x, C_j) \) for \( j \neq k \)

Probabilistic (soft) clustering uses (subjective) probabilities, or cluster membership functions,

\[
p_k(x) := \text{Prob} \{ x \in C_k \}, \quad p_k(x) \geq 0, \quad \sum_k p_k(x) = 1
\]
Clustering

Clustering = partitioning a set $\mathcal{D}$ into clusters, i.e. disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of similar elements, with elements of different clusters dissimilar.

The number of clusters, $K$, is in general unknown.

Distance clustering: Similar means close in some metric sense. For example, $x \in \mathcal{C}_k$ if $d(x, \mathcal{C}_k) < d(x, \mathcal{C}_j)$ for $j \neq k$.

Probabilistic (soft) clustering uses (subjective) probabilities, or cluster membership functions,

$$p_k(x) := \text{Prob}\{x \in \mathcal{C}_k\}, \quad p_k(x) \geq 0, \quad \sum_k p_k(x) = 1$$

- $p_k(x)$ measures the strength of belief that $x \in \mathcal{C}_k$
- Deterministic (hard) clustering: $\forall x, p_k(x) = 1$ for some $k$
Distances

Data points are vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$

A distance function $d(\mathbf{x}, \mathbf{y})$ is given in $\mathbb{R}^n$,

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$
Distances

Data points are vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$

A distance function $d(\mathbf{x}, \mathbf{y})$ is given in $\mathbb{R}^n$,

$$d(\mathbf{x}, \mathbf{y}) = \| \mathbf{x} - \mathbf{y} \|, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

For example, $\| \cdot \|$ is an elliptic norm, defined by

$$\| \mathbf{u} \| = \langle \mathbf{u}, Q \mathbf{u} \rangle^{1/2}, \text{ with } Q \text{ a positive definite matrix.}$$
Distances

Data points are vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$

A distance function $d(\mathbf{x}, \mathbf{y})$ is given in $\mathbb{R}^n$,

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

For example, $\| \cdot \|$ is an elliptic norm, defined by

$$\|\mathbf{u}\| = \langle \mathbf{u}, Q\mathbf{u} \rangle^{1/2}, \text{ with } Q \text{ a positive definite matrix.}$$

In particular, $Q = I$ gives the Euclidean norm,

$$\|\mathbf{u}\| = \langle \mathbf{u}, \mathbf{u} \rangle^{1/2},$$
Distances

Data points are vectors $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$

A distance function $d(x, y)$ is given in $\mathbb{R}^n$,

$$d(x, y) = \|x - y\|, \ \forall x, y \in \mathbb{R}^n$$

For example, $\| \cdot \|$ is an elliptic norm, defined by

$$\|u\| = \langle u, Qu \rangle^{1/2}, \text{ with } Q \text{ a positive definite matrix.}$$

In particular, $Q = I$ gives the Euclidean norm,

$$\|u\| = \langle u, u \rangle^{1/2},$$

and the Mahalanobis distance corresponds to $Q = \Sigma^{-1}$, where $\Sigma$ is the covariance matrix of the data involved.
CLASSIFICATION
Classification rule
Classification rule

![Classification rule diagram](image)
Classification

A population is partitioned into several disjoint classes. Each element is of the form \((x, y)\), where \(x\) is a vector of observations, and \(y\), the label, is the class where \(x\) belongs.
Classification

A population is partitioned into several disjoint classes. Each element is of the form \((x, y)\), where \(x\) is a vector of observations, and \(y\), the label, is the class where \(x\) belongs.

In general, the number of classes \(K\) is known,

- the vector \(x\) is easily observable,
- the label \(y\) is unknown, but depends somehow on \(x\).
Classification

A population is partitioned into several disjoint classes. Each element is of the form \((x, y)\), where \(x\) is a vector of observations, and \(y\), the label, is the class where \(x\) belongs.

In general, the number of classes \(K\) is known,

- the vector \(x\) is easily observable,
- the label \(y\) is unknown, but depends somehow on \(x\).

A classification rule is a function \(y = f(x)\) that assigns class values \(y\) to the observations \(x\).

Classification (also called supervised learning) is the process of determining a classification rule, from a sample data (the training set) with known labels.
Classification

A population is partitioned into several disjoint classes. Each element is of the form \((x, y)\), where \(x\) is a vector of observations, and \(y\), the label, is the class where \(x\) belongs.

In general, the number of classes \(K\) is known,

- the vector \(x\) is easily observable,
- the label \(y\) is unknown, but depends somehow on \(x\).

A classification rule is a function \(y = f(x)\) that assigns class values \(y\) to the observations \(x\).

Classification (also called supervised learning) is the process of determining a classification rule, from a sample data (the training set) with known labels.

The rule is tested on a testing set, also with known labels.
Application: Medical diagnosis

\( \mathcal{D} \) is a medical data set with elements \((x, y)\), one for each patient,

- \( x \) is a vector of test results on a patient,
- \( y \) denotes the medical status, say

\[
y = \begin{cases} 
1, & \text{if the patient is sick;} \\
0, & \text{if healthy.}
\end{cases}
\]
**Application: Medical diagnosis**

$\mathcal{D}$ is a medical data set with elements $(x, y)$, one for each patient,

- $x$ is a vector of test results on a patient,
- $y$ denotes the medical status, say

$$y = \begin{cases} 
1, & \text{if the patient is sick;} \\
0, & \text{if healthy.}
\end{cases}$$

The classification rule $y = f(x)$ may result in error, say

$$f(x) = \begin{cases} 
0, & \text{if the patient is sick, false negative;} \\
1, & \text{if healthy, false negative.}
\end{cases}$$

Typically, these classification errors have different consequences, and we may want to reduce one at the cost of increasing the other.
PROBABILITIES, DISTANCES, & CLUSTER SIZES
Distances and cluster sizes

Let the data set $\mathcal{D}$ be partitioned into clusters $C_1, \ldots, C_K$, with cluster sizes $q_1, \ldots, q_K$ and centers $c_1, \ldots, c_K$, respectively.
Distances and cluster sizes

Let the data set $\mathcal{D}$ be partitioned into clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$, with cluster sizes $q_1, \ldots, q_K$ and centers $c_1, \ldots, c_K$, respectively.

For any data point $x \in \mathcal{D}$ denote

$$d_k(x) := d_k(x, c_k), \text{ distance from the center } c_k$$

$$p_k(x) := \text{probability of belonging to } \mathcal{C}_k.$$
Distances and cluster sizes

Let the data set $\mathcal{D}$ be partitioned into clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$, with cluster sizes $q_1, \ldots, q_K$ and centers $c_1, \ldots, c_K$, respectively.

For any data point $x \in \mathcal{D}$ denote

$$d_k(x) := d_k(x, c_k), \quad \text{distance from the center } c_k$$

$$p_k(x) := \text{probability of belonging to } \mathcal{C}_k.$$

The distance functions $d_k(\cdot)$ depend in general on the cluster, e.g.,

$$d_k(x, c) = \langle (x - c), \Sigma_k^{-1}(x - c) \rangle^{1/2}$$

where $\Sigma_k$ is the covariance of the $k_{\text{th}}$ cluster.
Distances and cluster sizes

Let the data set $D$ be partitioned into clusters $C_1, \ldots, C_K$, with cluster sizes $q_1, \ldots, q_K$ and centers $c_1, \ldots, c_K$, respectively.

For any data point $x \in D$ denote

$$d_k(x) := d_k(x, c_k), \text{ distance from the center } c_k$$

$$p_k(x) := \text{probability of belonging to } C_k.$$ 

The distance functions $d_k(\cdot)$ depend in general on the cluster, e.g.,

$$d_k(x, c) = \langle (x - c), \Sigma_k^{-1}(x - c) \rangle^{1/2}$$

where $\Sigma_k$ is the covariance of the $k_{th}$ cluster.

The cluster sizes $q_k$ are known in some applications, and parameters to be estimated in others.
Probabilities, distances and cluster sizes

Let the cluster distances $d_k(x)$ and the cluster sizes $q_k$ be given. Assume:

Cluster membership is more probable the closer is the data point to the cluster, and the bigger the cluster.
Probabilities, distances and cluster sizes

Let the cluster distances $d_k(x)$ and the cluster sizes $q_k$ be given. Assume:

Cluster membership is more probable the closer is the data point to the cluster, and the bigger the cluster.

**Principle:** For all $x \in D$, the probabilities $p_k(x)$ satisfy,

$$\frac{p_k(x) \cdot d_k(x)}{q_k} = D(x), \text{ a constant, depending on } x.$$  \(\heartsuit\)
Probabilities, distances and cluster sizes

Let the cluster distances $d_k(x)$ and the cluster sizes $q_k$ be given. Assume:

**Cluster membership is more probable the closer is the data point to the cluster, and the bigger the cluster.**

**Principle:** For all $x \in D$, the probabilities $p_k(x)$ satisfy,

$$\frac{p_k(x) d_k(x)}{q_k} = D(x), \text{ a constant, depending on } x. \quad (\heartsuit)$$

If there is no need to treat $q_k$ separately, e.g., if the cluster sizes are similar, or can be ignored, consider the ratio

$$\frac{d_k(x)}{q_k} \text{ as a weighted distance}$$

and denote it by $d_k(x)$. 
Probabilities

From \( p_k(x) \cdot d_k(x) = D(x) \), and \( \sum_k p_k(x) = 1 \), it follows that,

\[
p_k(x) = \frac{\prod_{j \neq k} d_j(x)}{\sum_i \prod_{j \neq i} d_j(x)} , \quad k \in \{1, K\}.
\]
**Probabilities**

From \( p_k(x) \cdot d_k(x) = D(x) \), and \( \sum_k p_k(x) = 1 \), it follows that,

\[
p_k(x) = \frac{\prod_{j \not= k} d_j(x)}{\sum_{i=1}^{K} \prod_{j \not= i} d_j(x)}, \quad k \in \overline{1,K}.
\]

In particular, for \( K = 2 \),

\[
p_1(x) = \frac{d_2(x)}{d_1(x) + d_2(x)}, \quad p_2(x) = \frac{d_1(x)}{d_1(x) + d_2(x)},
\]

and for \( K = 3 \),

\[
p_1(x) = \frac{d_2(x)d_3(x)}{d_1(x)d_2(x) + d_1(x)d_3(x) + d_2(x)d_3(x)}, \quad \text{etc.}
\]
An extremal principle

The principle

\[ p_1(x)d_1(x) = p_2(x)d_2(x), \]  

is an **optimality condition** for the problem

\[
\begin{align*}
\min & \quad d_1(x) p_1^2 + d_2(x) p_2^2 \\
\text{s.t.} & \quad p_1 + p_2 = 1 \\
& \quad p_1, p_2 \geq 0
\end{align*}
\]  

(P)
An extremal principle

The principle

\[ p_1(x)d_1(x) = p_2(x)d_2(x), \]  

(\heartsuit)

is an **optimality condition** for the problem

\[
\min d_1(x) p_1^2 + d_2(x) p_2^2 \quad (P)
\]

s.t. \[ p_1 + p_2 = 1 \]

\[ p_1, p_2 \geq 0 \]

as shown by differentiating the **Lagrangian**, \[ L(p_1, p_2, \lambda) = d_1(x) p_1^2 + d_2(x) p_2^2 - \lambda(p_1 + p_2 - 1) \]

with respect to \( p_1, p_2, \)
An extremal principle

The principle

\[ p_1(x)d_1(x) = p_2(x)d_2(x), \quad (\heartsuit) \]

is an **optimality condition** for the problem

\[
\begin{align*}
\min \quad & d_1(x)p_1^2 + d_2(x)p_2^2 \\
\text{s.t.} \quad & p_1 + p_2 = 1 \\
& p_1, p_2 \geq 0
\end{align*}
\]

as shown by differentiating the **Lagrangian**, \( L(p_1, p_2, \lambda) = d_1(x)p_1^2 + d_2(x)p_2^2 - \lambda(p_1 + p_2 - 1) \)

with respect to \( p_1, p_2, \)

\[
\frac{\partial L}{\partial p_1} = 2p_1d_1(x) - \lambda = 0, \text{ etc.}
\]
$p^2$ ?
The problem

\[ \min \{ p_1^2 d_1 + p_2^2 d_2 : p_1 + p_2 = 1 , \ p_1, p_2 \geq 0 \} \]

is a \textit{smoothed} version of

\[ \min \{ p_1 d_1 + p_2 d_2 : p_1 + p_2 = 1 , \ p_1, p_2 \geq 0 \} \implies \min \{ d_1, d_2 \} . \]
The problem

$$\min \{ p_1^2 d_1 + p_2^2 d_2 : p_1 + p_2 = 1, \, p_1, p_2 \geq 0 \}$$

is a **smoothed** version of

$$\min \{ p_1 d_1 + p_2 d_2 : p_1 + p_2 = 1, \, p_1, p_2 \geq 0 \} \implies \min \{ d_1, d_2 \}.$$  

$p^2$? Yes

The problem

$$\min \{ p_1^2 d_1 + p_2^2 d_2 : p_1 + p_2 = 1 , \ p_1, p_2 \geq 0 \}$$

is a smoothed version of

$$\min \{ p_1 d_1 + p_2 d_2 : p_1 + p_2 = 1 , \ p_1, p_2 \geq 0 \} \implies \min \{ d_1, d_2 \} .$$


Other schemes include entropic smoothing

$$\min \{ p_1 d_1 + p_2 d_2 + p_1 \log p_1 + p_2 \log p_2 : p_1 + p_2 = 1 , \ p_1, p_2 \geq 0 \}$$

resulting in the principle

$$p_k(x) e^{d_k(x)} = \text{independent of } k .$$
An electrical analogy: Parallel circuit

$$\min \{p_1^2d_1 + p_2^2d_2 : p_1 + p_2 = 1, p_1, p_2 \geq 0\}$$

$$p_1d_1 = p_2d_2$$
An electrical analogy: Parallel circuit

\[
\min \{p_1^2 d_1 + p_2^2 d_2 : p_1 + p_2 = 1, p_1, p_2 \geq 0\}
\]

\[
p_1 d_1 = p_2 d_2
\]

\[
\min \{I_1^2 R_1 + I_2^2 R_2 : I_1 + I_2 = I\}
\]

\[
I_1 R_1 = I_2 R_2 = \text{voltage drop from } A \text{ to } B
\]
Probabilistic assignment

Given a point $\mathbf{x}$ and a cluster $C_k$ with center $c_k$, let

$$p_k(\mathbf{x}) := \text{Prob} \{ \mathbf{x} \in C_k \}$$

with $p_k(\mathbf{x}) \geq 0$, $\sum_{k=1}^{K} p_k(\mathbf{x}) = 1$

$$d_k(\mathbf{x}) := d(\mathbf{x}, c_k)$$
Probabilistic assignment

Given a point \( x \) and a cluster \( C_k \) with center \( c_k \), let

\[
p_k(x) := \text{Prob} \{ x \in C_k \}
\]

with \( p_k(x) \geq 0, \sum_{k=1}^{K} p_k(x) = 1 \)

\[
d_k(x) := d(x, c_k)
\]

The membership probabilities \( \{p_k(x) : k \in \overline{1,K}\} \) of a point \( x \) depend only on the distances \( \{d_k(x) : k \in \overline{1,K}\} \),

\[
p(x) = f(d(x))
\]

where \( p(x) \in \mathbb{R}^K \) is the vector of probabilities \( (p_k(x)) \), and \( d(x) \) is the vector of distances \( (d_k(x)) \).
Desired properties of $p(x) = f(d(x))$

For any $x$, $i, j, k \in 1, K$, and any permutation matrices $Q$,

\[ d_i(x) < d_j(x) \implies p_i(x) > p_j(x), \quad (a) \]
\[ f(\lambda d(x)) = f(d(x)), \text{ for any } \lambda > 0 \quad (b) \]
\[ Q p(x) = f(Q d(x)), \quad (c) \]
\[ f \text{ is continuous}, \quad (d) \]
Desired properties of \( p(x) = f(d(x)) \)

For any \( x, \ i, j, k \in 1,K \), and any permutation matrices \( Q \),

\[
d_i(x) < d_j(x) \implies p_i(x) > p_j(x), \quad (a)
\]

\[
f(\lambda d(x)) = f(d(x)), \text{ for any } \lambda > 0 \quad (b)
\]

\[
Q p(x) = f(Q d(x)), \quad (c)
\]

\( f \) is continuous, \quad (d)

\[
p_k(x) = p_k(x|S) p_S(x), \quad (e)
\]

where \( p_S(x) := \sum_{s \in S} p_s(x) , \quad \forall S \subset 1,K \)

and \( p_k(x|S) \) is the **conditional probability** of \( x \in C_k \) given \( x \in S \).
Desired properties of $p(x) = f(d(x))$

For any $x$, $i, j, k \in \overline{1,K}$, and any permutation matrices $Q$,

\begin{align*}
    d_i(x) < d_j(x) & \implies p_i(x) > p_j(x), \quad (a) \\
    f(\lambda d(x)) &= f(d(x)), \text{ for any } \lambda > 0 \quad (b) \\
    Qp(x) &= f(Qd(x)), \quad (c) \\
    f \text{ is continuous,} \quad (d) \\
    &p_k(x) = p_k(x|S)p_S(x), \quad (e)
\end{align*}

where $p_S(x) := \sum_{s \in S} p_s(x)$, $\forall S \subset \overline{1,K}$

and $p_k(x|S)$ is the conditional probability of $x \in C_k$ given $x \in S$.

Property (e) is the choice axiom of Luce.

The choice axiom of Luce

\[ p_k(x) = p_k(x|S) p_S(x), \quad \forall \ k \in 1,K, \ S \subset 1,K \]  

(e)

is equivalent to

\[ p_k(x|S) = \frac{v_k(x)}{\sum_{s \in S} v_s(x)} \]

where \( v_k(x) \) is a scale function,
The choice axiom of Luce

\[ p_k(x) = p_k(x|S) p_S(x), \quad \forall \, k \in \overline{1,K}, \; S \subset \overline{1,K} \]  \quad (e)

is equivalent to

\[ p_k(x|S) = \frac{v_k(x)}{\sum_{s \in S} v_s(x)} \]

where \( v_k(x) \) is a scale function, in particular,

\[ p_k(x) = \frac{v_k(x)}{\sum_{s \in \overline{1,K}} v_s(x)}. \]
The choice axiom of Luce

\[ p_k(x) = p_k(x|S) p_S(x), \quad \forall k \in \overline{1,K}, \ S \subset \overline{1,K} \quad (e) \]

is equivalent to

\[ p_k(x|S) = \frac{v_k(x)}{\sum_{s \in S} v_s(x)} \]

where \( v_k(x) \) is a scale function, in particular,

\[ p_k(x) = \frac{v_k(x)}{\sum_{s \in \overline{1,K}} v_s(x)} . \]

Therefore, for all \( k \in \overline{1,K} \),

\[ p_k(x) v_k^{-1}(x) = D(x), \]

a function of \( x \), independent of \( k \).
Probabilities and distances

From

\[ d_i(x) < d_j(x) \implies p_i(x) > p_j(x), \]

and

\[ p_k(x) = \frac{v_k(x)}{\sum_{s \in 1, K} v_s(x)}. \]

it follows that \( v_k(x) \) is a decreasing function of \( d_k(x) \).
Probabilities and distances

From

\[ d_i(x) < d_j(x) \implies p_i(x) > p_j(x), \]

and

\[ p_k(x) = \frac{v_k(x)}{\sum_{s \in 1,K} v_s(x)}. \]

it follows that \( v_k(x) \) is a decreasing function of \( d_k(x) \), in particular,

\[ v_k(x) = \frac{1}{d_k(x)} \]

and

\[ p_k(x) v_k^{-1}(x) = D(x) \]

becomes

\[ p_k(x) d_k(x) = D(x) \]

independent of \( k \). This is our working principle.
CONTOUR APPROXIMATION
Joint distance function

From $\sum_k p_k(x) = 1$ and

$$p_k(x) = \frac{D(x)}{d_k(x, c_k)}, \quad k = 1, \ldots, K$$
Joint distance function

From \( \sum_k p_k(x) = 1 \) and

\[
p_k(x) = \frac{D(x)}{d_k(x, c_k)}, \quad k = 1, \ldots, K
\]

we get

\[
D(x) = \frac{\prod_{j=1}^{K} d_j(x)}{\sum_{i=1}^{K} \prod_{j \neq i} d_j(x)},
\]

called the joint distance function (JDF) of \( x \).
Joint distance function

From $\sum_k p_k(x) = 1$ and

$$p_k(x) = \frac{D(x)}{d_k(x, c_k)} , \ k = 1, \ldots, K$$

we get

$$D(x) = \frac{\prod_{j=1}^{K} d_j(x, c_j)}{\sum_{i=1}^{K} \prod_{j \neq i} d_j(x, c_j)} ,$$

called the joint distance function (JDF) of $x$.

$D(x)$ is (up to a constant) the harmonic mean of the distances of $x$ from all centers,

$$D(x) = \frac{1}{\frac{1}{d_1(x, c_1)} + \frac{1}{d_2(x, c_2)} + \cdots + \frac{1}{d_K(x, c_K)}}$$
Special cases

The joint distance function

\[ D(x) = \frac{\prod_{j=1}^{K} d_j(x, c_j)}{\sum_{i=1}^{K} \prod_{j \neq i} d_j(x, c_j)} , \]
Special cases

The joint distance function

\[ D(x) = \frac{\prod_{j=1}^{K} d_j(x)}{\sum_{i=1}^{K} \prod_{i \neq j} d_j(x)} , \]

for \( K = 2, \)

\[ D(x) = \frac{d_1(x)d_2(x)}{d_1(x) + d_2(x)} = \frac{1}{1/d_1(x) + 1/d_2(x)} \]

and for \( K = 3, \)

\[ D(x) = \frac{d_1(x)d_2(x)d_3(x)}{d_1(x)d_2(x) + d_1(x)d_3(x) + d_2(x)d_3(x)} \]
The JDF as optimal value

The optimal value of

$$\min \quad d_1(x) p_1^2 + d_2(x) p_2^2$$

$$\text{s.t.} \quad p_1 + p_2 = 1$$

$$p_1, p_2 \geq 0$$
The JDF as optimal value

The optimal value of

$$\begin{align*}
\min & \quad d_1(x, c_1) p_1^2 + d_2(x, c_2) p_2^2 \\
\text{s.t.} & \quad p_1 + p_2 = 1 \\
& \quad p_1, p_2 \geq 0
\end{align*}$$

(P)

obtained by substituting the optimal solution

$$p_1 = \frac{d_2(x, c_2)}{d_1(x, c_1) + d_2(x, c_2)} , \quad p_2 = \frac{d_1(x, c_1)}{d_1(x, c_1) + d_2(x, c_2)} ,$$
The JDF as optimal value

The optimal value of

\[
\min \quad d_1(x, c_1) p_1^2 + d_2(x, c_2) p_2^2 \\
\text{s.t.} \quad p_1 + p_2 = 1 \\
\quad p_1, p_2 \geq 0
\]

obtained by substituting the optimal solution

\[
p_1 = \frac{d_2(x)}{d_1(x) + d_2(x)} , \quad p_2 = \frac{d_1(x)}{d_1(x) + d_2(x)} ,
\]

is the joint distance function

\[
D(x) = \frac{d_1(x) d_2(x)}{d_1(x) + d_2(x)}
\]
An electrical analogy: Parallel resistance

\[
\min \{ p_1^2 d_1 + p_2^2 d_2 : p_1 + p_2 = 1, p_1, p_2 \geq 0 \} = \frac{d_1 d_2}{d_1 + d_2}
\]

\[
\begin{align*}
\min \{ I_1^2 R_1 + I_2^2 R_2 : I_1 + I_2 = I \} &= I^2 R \\
R &= \frac{1}{1/R_1 + 1/R_2} = \frac{R_1 R_2}{R_1 + R_2}
\end{align*}
\]
The JDF of the dataset

The sum of the joint distance functions of all $N$ data points

$$f(c_1, c_2) = \sum_{i=1}^{N} \frac{d_1(x_i, c_1) \cdot d_2(x_i, c_2)}{d_1(x_i, c_1) + d_2(x_i, c_2)}$$

a function of the cluster centers $c_k$, and distance functions $d_k$
The JDF of the dataset

The sum of the joint distance functions of all $N$ data points

$$f(c_1, c_2) = \sum_{i=1}^{N} \frac{d_1(x_i, c_1) d_2(x_i, c_2)}{d_1(x_i, c_1) + d_2(x_i, c_2)}$$

a function of the cluster centers $c_k$, and distance functions $d_k$

The function $f(c_1, c_2)$ is **not convex**, not even quasi–convex, so the best one could hope for is a **stationary point**.
Properties and significance of the JDF

\[ D(\mathbf{x}) = \frac{d_1(\mathbf{x}, \mathbf{c}_1)d_2(\mathbf{x}, \mathbf{c}_2)}{d_1(\mathbf{x}, \mathbf{c}_1) + d_2(\mathbf{x}, \mathbf{c}_2)} \]

- \( D(\mathbf{x}) = 0 \iff \mathbf{x} = \mathbf{c}_1, \mathbf{c}_2, \) and otherwise \( D(\mathbf{x}) > 0 \)
- \( D(\mathbf{x}) \) is not convex, not even quasi convex, and may have other stationary points that are necessarily saddle points.
- \( D(\mathbf{x}) \) is a measure of the **classifiability** of the point \( \mathbf{x} \),
  - easier [more difficult] for small [large] values of \( D(\mathbf{x}) \).
- If the distances are equal, \( d_1(\mathbf{x}, \mathbf{c}_1) = d_2(\mathbf{x}, \mathbf{c}_2) = d \), then
  \[ D(\mathbf{x}) = \frac{d}{2} \]
  corresponding to \( p_1(\mathbf{x}) = p_2(\mathbf{x}) = \frac{1}{2} \) (indifference).
Contour approximation of data
Contour approximation: 3 clusters
A contour approximation of data

M. Arav, Contour approximation of data and the harmonic mean, 
*J. of Mathematical Inequalities* **2**(2008), 161–167

Given a **data set** $D$ in $\mathbb{R}^n$ with **$K$ clusters**, their **centers** $c_k$ and their **distance functions** $d_k(\cdot, c_k)$, a **contour approximation** of $D$ is a function $D(x) = F(d_1(x, c_1), \cdots, d_K(x, c_K))$ that captures the data in its lower level sets: $D(x) \leq d_k(x, c_k)$, $\forall k$. 
Desirable properties

for a function \( F(d_1, d_2, \cdots, d_K) : \mathbb{R}_+^K \rightarrow \mathbb{R}_+ \) to be a contour approximation:

(a) \( F \) is a differentiable function of \( d_1, \cdots, d_K \).

(b) \( F \) is symmetric in \( d_1, \cdots, d_K \).

(c) \( F(d, d_2, \cdots, d_K) \) is an increasing function of \( d \).

(d) \( F(\lambda d_1, \lambda d_2, \cdots, \lambda d_K) = \lambda F(d_1, d_2, \cdots, d_K), \forall \lambda > 0 \).

(e) \( F(d_1, d_2, \cdots, d_K) \leq \min\{d_1, d_2, \cdots, d_K\} \).

(f) \( F(d_1, d_2, \cdots, d_K) \leq \frac{1}{K} \max\{d_1, d_2, \cdots, d_K\} \),

with equality only if \( d_1 = d_2 = \cdots = d_K \).

(g) \( \frac{\partial F(0, d_2, \cdots, d_K)}{\partial d_1} = 1 \).
Harmony in the animal kingdom

Kenneth R. Dixon and Joseph A. Chapman,

Harmonic mean measure of animal activity areas


A new method of calculating centers and areas of animal activity is presented based on the harmonic mean of an areal distribution. The center of activity is located in the area of greatest activity; in fact more than one “center” may exist.
Harmony in the animal kingdom

Kenneth R. Dixon and Joseph A. Chapman,
Harmonic mean measure of animal activity areas

A new method of calculating centers and areas of animal activity is presented based on the harmonic mean of an areal distribution. The center of activity is located in the area of greatest activity; in fact more than one “center” may exist.

... The calculation of home range allows for heterogeneity of any habitat and is illustrated with data collected near Corvallis, Oregon, on the brush rabbit (Sylvilagus bachmani)
Home range of black bear (ID)

Location data (49 points)  Areas of concentrated use,

M.D. Samuel, D.J. Pierce and E.O. Garton, Identifying areas of concentrated use within the home range, *J. Animal Ecology* 54(1985), 711–719
Contour approximation of home range

Contours of the JDF (2 centers)    Contours of the JDF (3 centers)

M.D. Samuel, D.J. Pierce and E.O. Garton, **Identifying areas of concentrated use within the home range**, *J. Animal Ecology* **54**(1985), 711–719
CENTERS
Centers

Consider the problem

$$\min \sum_{i=1}^{N} d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2 \quad (P)$$

s.t. \quad p_1(x_i) + p_2(x_i) = 1, \ i = 1, \ldots, N$
$$p_1(x_i), p_2(x_i) \geq 0$$
Consider the problem

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{N} d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2 \\
\text{s.t.} & \quad p_1(x_i) + p_2(x_i) = 1, \quad i = 1, \ldots, N \\
& \quad p_1(x_i), p_2(x_i) \geq 0
\end{align*}
\]

(P)

with \textbf{elliptic distances}

\[
d_k(x, c_k) = \langle (x - c_k), Q_k(x - c_k) \rangle^{1/2}, \quad k = 1, 2,
\]

and given \textbf{probabilities} \( p_k(x_i), \quad i = 1, \ldots, N, \quad k = 1, 2. \)
Centers

Consider the problem

$$\min \sum_{i=1}^{N} d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2$$  \hspace{1cm} (P)

s.t. \quad p_1(x_i) + p_2(x_i) = 1, \quad i = 1, \ldots, N

$$p_1(x_i), \quad p_2(x_i) \geq 0$$

with elliptic distances

$$d_k(x, c_k) = \langle (x - c_k), Q_k(x - c_k) \rangle^{1/2}, \quad k = 1, 2,$$

and given probabilities \( p_k(x_i), \quad i = 1, \ldots, N, \quad k = 1, 2. \)

**Problem:** Find the points \( c_1, c_2 \) minimizing

$$f(c_1, c_2) = \sum_{i=1}^{N} (d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2)$$
Centers (cont’d)

**Theorem:** The minimizers of $f(c_1, c_2)$ are

$$c_k = \sum_{i=1}^{N} \left( \frac{u_k(x_i)}{\sum_{j=1}^{N} u_k(x_j)} \right) x_i,$$

where

$$u_k(x_i) = \frac{p_k(x_i)^2}{d_k(x_i, c_k)},$$

for $k = 1, 2,$
Centers (cont’d)

Theorem: The minimizers of \( f(c_1, c_2) \) are

\[ c_k = \sum_{i=1}^{N} \left( \frac{u_k(x_i)}{\sum_{j=1}^{N} u_k(x_j)} \right) x_i, \]

where

\[ u_k(x_i) = \frac{p_k(x_i)^2}{d_k(x_i)}, \]

for \( k = 1, 2 \), or equivalently,

\[ u_1(x_i) = \frac{d_2(x_i)^2}{d_1(x_i)(d_1(x_i) + d_2(x_i))^2}, \]

\[ u_2(x_i) = \frac{d_1(x_i)^2}{d_2(x_i)(d_1(x_i) + d_2(x_i))^2}. \]
Centers (cont’d)

Proof: Let $d_k(x_i, c_k) = \langle (x_i - c_k), Q_k(x_i - c_k) \rangle^{1/2}$.

$$\nabla_c (d_k(x, c)) = -\frac{Q_k(x - c)}{d_k(x, c)}$$
Proof: Let $d_k(x_i, c_k) = \langle (x_i - c_k), Q_k(x_i - c_k) \rangle^{1/2}$.

$$\nabla_c \ (d_k(x, c)) = -\frac{Q_k(x - c)}{d_k(x, c)}$$

$$\therefore \nabla_{c_1} \sum_{i=1}^{N} \left( d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2 \right)$$

$$= - \sum_{i=1}^{N} \left( \frac{Q_1(x_i - c_1)}{d_1(x_i, c_1)} p_1(x_i)^2 \right)$$
Centers (cont’d)

Proof: Let \( d_k(x_i, c_k) = \langle (x_i - c_k), Q_k(x_i - c_k) \rangle^{1/2} \).

\[
\nabla_c (d_k(x, c)) = -\frac{Q_k(x - c)}{d_k(x, c)}
\]

\[
\therefore \nabla_{c_1} \sum_{i=1}^{N} \left( d_1(x_i, c_1) p_1(x_i)^2 + d_2(x_i, c_2) p_2(x_i)^2 \right)
\]

\[
= - \sum_{i=1}^{N} \left( \frac{Q_1(x_i - c_1)}{d_1(x_i, c_1)} p_1(x_i)^2 \right)
\]

Zeroing and summing like terms gives

\[
\sum_{i=1}^{N} \left( \frac{p_1(x_i)^2}{d_1(x_i, c_1)} \right) x_i = \left( \sum_{i=1}^{N} \frac{p_1(x_i)^2}{d_1(x_i, c_1)} \right) c_1 .
\]
A PD Clustering Algorithm

| Initialization: | given data set $\mathcal{D}$ with $N$ points, |
|                | any two centers $c_1, c_2$, |
|                | $\epsilon > 0$ |

| Iteration: |
| Step 1    | compute distances from $c_1, c_2$ for all $x \in \mathcal{D}$ |
| Step 2    | update the centers $c_1^+, c_2^+$ |
| Step 3    | update the covariances $\Sigma_1, \Sigma_2$ |
| Step 4    | if $\|c_1^+ - c_1\| + \|c_2^+ - c_2\| < \epsilon$ stop |
|           | return to step 1 |

(a) Step 3 absent for location problems.
(b) Probabilities not needed explicitly, except perhaps at the end.
EXAMPLES
$N = 15, \ K = 3$ (Cooper, 1964)
$N = 15, \ K = 3$ (Cooper, 1964)
$N = 50, \ K = 5$ (Eilon, 1971)
$N = 50, \ K = 5$ (Eilon, 1971)
$N = 287, \ K = 3$ (Bongartz, 1994)
$N = 287, \; K = 2, \cdots, 5$ (Bongartz, 1994)

<table>
<thead>
<tr>
<th>$K$</th>
<th>Optimal value*</th>
<th>Best value found</th>
<th>Ave. value found</th>
<th>Ave. CPU</th>
<th>Ave. no. iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14427</td>
<td>14481</td>
<td>14841</td>
<td>0.0344</td>
<td>38.9</td>
</tr>
<tr>
<td>3</td>
<td>12095</td>
<td>12124</td>
<td>12577</td>
<td>0.801</td>
<td>30.3</td>
</tr>
<tr>
<td>4</td>
<td>10661</td>
<td>10682</td>
<td>11598</td>
<td>0.6754</td>
<td>20.6</td>
</tr>
<tr>
<td>5</td>
<td>9715</td>
<td>9743</td>
<td>10787</td>
<td>1.47</td>
<td>18.6</td>
</tr>
</tbody>
</table>

Results of 200 runs from random initials

Evolution of the joint distance function:
2 clusters
Convergence to true centers from any starting point
Evolution of the joint distance function:
3 clusters
VALIDATION
Validation

A difficult problem in clustering is to determine the “right” number $K$ of clusters that fits a data set with $N$ points.
Validation

A difficult problem in clustering is to determine the “right” number $K$ of clusters that fits a data set with $N$ points.

In dichotomous situations the unambiguous answer is $K = 2$, but in general, the answer lies between the two extremes of $K = 1$ (one cluster fits all), and $K = N$ (each point is a cluster.)
Validation

A difficult problem in clustering is to determine the “right” number $K$ of clusters that fits a data set with $N$ points.

In dichotomous situations the unambiguous answer is $K = 2$, but in general, the answer lies between the two extremes of $K = 1$ (one cluster fits all), and $K = N$ (each point is a cluster.)

The joint distance function offers a practical solution. Indeed, the JDF decreases monotonically with $K$, the number of clusters, and the decrease is precipitous until the “right” number is reached, and after that the rate of decrease is small.
Validation

A difficult problem in clustering is to determine the “right” number $K$ of clusters that fits a data set with $N$ points.

In dichotomous situations the unambiguous answer is $K = 2$, but in general, the answer lies between the two extremes of $K = 1$ (one cluster fits all), and $K = N$ (each point is a cluster.)

The joint distance function offers a practical solution. Indeed, the JDF decreases monotonically with $K$, the number of clusters, and the decrease is precipitous until the “right” number is reached, and after that the rate of decrease is small.

Recipe: Use $K = 1, 2, \cdots$ and stop when the JDF stops decreasing.
Example: 2 clusters
Example: 3 clusters
Example: 4 clusters
What if there is no structure?
What is the correct number of clusters?
UNCERTAINTY
The uncertainty of classification

The JDF \( D(x) \) has the dimension of distance. Normalizing it, we get the dimensionless function

\[
E(x) = K D(x) / \left( \prod_{j=1}^{K} d_j(x) \right)^{1/K},
\]

with \( 0/0 \) interpreted as zero, or

\[
E(x) = K \left( \prod_{j=1}^{K} p_j(x) \right)^{1/K}.
\]

In particular, for \( K = 2 \),

\[
E(x) = 2 \frac{\sqrt{d_1(x)d_2(x)}}{d_1(x) + d_2(x)} = 2 \sqrt{p_1(x)p_2(x)}. 
\]
The uncertainty of classification

The function $E(x) = K D(x) / \left( \prod_{j=1}^{K} d_j(x) \right)^{1/K}$ is the harmonic mean of the distances $\{d_j(x)\}$ divided by their geometric mean.
The uncertainty of classification

The function $E(x) = K D(x) / \left( \prod_{j=1}^{K} d_j(x) \right)^{1/K}$ is the harmonic mean of the distances $\{d_j(x)\}$ divided by their geometric mean.

Therefore $0 \leq E(x) \leq 1$, with

$E(x) = 0 \iff d_j(x) = 0, \exists j$

$E(x) = 1 \iff p_k(x) = \frac{1}{K}, \forall k$

$E(x)$ is called the classification uncertainty function (CUF) at $x$. 
The uncertainty of classification

The function \( E(x) = K D(x) / \left( \prod_{j=1}^{K} d_j(x) \right)^{1/K} \) is the harmonic mean of the distances \( \{d_j(x)\} \) divided by their geometric mean. Therefore \( 0 \leq E(x) \leq 1 \), with

\[
E(x) = 0 \iff d_j(x) = 0, \ \exists \ j
\]

\[
E(x) = 1 \iff p_k(x) = \frac{1}{K}, \ \forall \ k
\]

\( E(x) \) is called the classification uncertainty function (CUF) at \( x \).

The CUF of the data set \( \mathcal{D} = \{x_i : i \in \overline{1,N}\} \) is

\[
E(\mathcal{D}) := \frac{1}{N} \sum_{i=1}^{N} E(x_i).
\]
The Csiszár divergence

Let $\mathbb{P}^K$ be the set of $K$–dimensional probability vectors, denoted $\mathbf{p} = (p_i)$, $\mathbf{q} = (q_i)$. Given a convex function $\phi : \mathbb{R}^+ \to \mathbb{R}$, the Csiszár $\phi$–divergence, defined by

$$I_\phi(\mathbf{p}, \mathbf{q}) := \sum_{i=1}^{K} q_i \phi \left( \frac{p_i}{q_i} \right), \quad \text{with} \quad 0 \phi \left( \frac{0}{0} \right) := 0,$$

is a distance function on $\mathbb{P}^K$, a generalized measure of entropy.
The Csiszár divergence

Let $\mathbb{P}^K$ be the set of $K$–dimensional probability vectors, denoted $p = (p_i)$, $q = (q_i)$. Given a convex function $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$, the Csiszár $\phi$–divergence, defined by

$$I_\phi(p, q) := \sum_{i=1}^{K} q_i \phi \left( \frac{p_i}{q_i} \right), \quad \text{with} \quad 0 \phi \left( \frac{0}{0} \right) := 0,$$

is a distance function on $\mathbb{P}^K$, a generalized measure of entropy.

For the special case

$$\phi(t) := t \log t, \quad t > 0,$$

$(I_\phi)$ gives

$$I_{KL}(p, q) = \sum_{i=1}^{K} p_i \log \left( \frac{p_i}{q_i} \right),$$

the Kullback–Leibler distance.
The uncertainty of classification

\[ E(x) = K \left( \prod_{j=1}^{K} p_j(x) \right)^{1/K} = \left( \prod_{j=1}^{K} \left( \frac{p_j(x)}{1/K} \right) \right)^{1/K} \]

\[ \therefore - \log E(x) = \sum_{i \in 1,K} (1/K) \log \left( \frac{1/K}{p_i(x)} \right) = I_{KL} (p(x), \frac{1}{K} \mathbf{1}) , \]

the Kullback–Leibler distance between the distributions

\[ p(x) = (p_1(x), p_2(x), \cdots, p_K(x)) \quad \text{and} \quad \frac{1}{K} \mathbf{1} = (\frac{1}{K}, \frac{1}{K}, \cdots, \frac{1}{K}) . \]

Therefore

\[ E(x) = \exp \{ -I_{KL} (p(x), \frac{1}{K} \mathbf{1}) \} \]

is an entropic measure of the uncertainty of classification, a monotone increasing function of the uncertainty.
Contour approximation and classification uncertainty

Contours of the JDF

Contours of the CUF
Football and pizza
CLUSTER SIZES
Probabilistic clustering adjusted for cluster size

Two kinds of problems

1) Cluster sizes are given (e.g., \textit{capacitated multi–facility location problems}), or otherwise can be ignored.
Probabilistic clustering adjusted for cluster size

Two kinds of problems

1) Cluster sizes are given (e.g., capacitated multi–facility location problems), or otherwise can be ignored.

2) Cluster sizes are unknown, e.g., uncappeditate multi–facility location problems, or mixtures of distributions, where weights and other parameters have to be estimated (by, say, the EM method).
Principle: Let $q_k$ be the size of cluster $k$.

$$\frac{p_k(x) d_k(x)}{q_k} = \text{constant, depending on } x.$$
Adjusting for cluster size

**Principle:** Let \( q_k \) be the size of cluster \( k \).

\[
\frac{p_k(x) \cdot d_k(x)}{q_k} = \text{constant, depending on } x.
\]

**Probabilities:**

\[
p_1(x) = \frac{d_2(x)/q_2}{d_1(x)/q_1 + d_2(x)/q_2}, \quad p_2(x) = \frac{d_1(x)/q_1}{d_1(x)/q_1 + d_2(x)/q_2},
\]
Principle: Let $q_k$ be the size of cluster $k$.

\[
\frac{p_k(x) d_k(x)}{q_k} = \text{constant, depending on } x.
\]

Probabilities:

\[
p_1(x) = \frac{d_2(x)/q_2}{d_1(x)/q_1 + d_2(x)/q_2}, \quad p_2(x) = \frac{d_1(x)/q_1}{d_1(x)/q_1 + d_2(x)/q_2},
\]

Joint distance function:

\[
D(x) = \frac{d_1(x) d_2(x)/q_1q_2}{d_1(x)/q_1 + d_2(x)/q_2}.
\]
An extremal principle

\[
\min \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right)
\]

s.t. \( q_1 + q_2 = N \)
An extremal principle

\[
\min \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right)
\]

s.t. \( q_1 + q_2 = N \)

\( p_1(x_i) + p_2(x_i) = 1 \)

\( p_1(x_i), p_2(x_i) \geq 0 \)
An extremal principle

\[
\begin{align*}
\min \quad & \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right) \\
\text{s.t.} \quad & q_1 + q_2 = N \\
& p_1(x_i) + p_2(x_i) = 1 \\
& p_1(x_i), p_2(x_i) \geq 0
\end{align*}
\]

where the minimization w.r.t. \( p_1, p_2 \) assumes \( q_1, q_2 \) fixed, and vice versa.
Cluster sizes

Taking the \textbf{cluster sizes} as variables in the extremal principle

\[
\min \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right)
\]

s.t. \(q_1 + q_2 = N\)

\(q_1, q_2 \geq 0\)
Cluster sizes

Taking the **cluster sizes** as variables in the extremal principle

\[
\min \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right)
\]

s.t. \( q_1 + q_2 = N \)

\( q_1, q_2 \geq 0 \)

with \( p_1(x_i), p_2(x_i) \) assumed known, we have the **Lagrangian**

\[
L(q_1, q_2, \lambda) = \sum_{i=1}^{N} \left( \frac{d_1(x_i) p_1(x_i)^2}{q_1} + \frac{d_2(x_i) p_2(x_i)^2}{q_2} \right) + \lambda(q_1 + q_2 - N)
\]
Zeroing the partials $\partial L / \partial q_k$ gives,

\[ q_k^2 = \frac{1}{\lambda} \left( \sum_{i=1}^{N} d_k(x_i) p_k(x_i)^2 \right), \quad k = 1, 2, \]
Zeroing the partials $\partial L / \partial q_k$ gives,

$$q_k^2 = \frac{1}{\lambda} \left( \sum_{i=1}^{N} d_k(x_i) p_k(x_i)^2 \right), \quad k = 1, 2,$$

and since $q_1 + q_2 = N$,

$$\frac{q_1}{N} = \frac{\left( \sum_{i=1}^{N} d_1(x_i) p_1(x_i)^2 \right)^{1/2}}{\left( \sum_{i=1}^{N} d_1(x_i) p_1(x_i)^2 \right)^{1/2}} + \frac{\left( \sum_{i=1}^{N} d_2(x_i) p_2(x_i)^2 \right)^{1/2}}{\left( \sum_{i=1}^{N} d_2(x_i) p_2(x_i)^2 \right)^{1/2}},$$

$$q_2 = N - q_1.$$
The PDQ Algorithm

Initialization: given data set $\mathcal{D}$ with $N$ points,
any two centers $c_1, c_2$,
any two cluster sizes $q_1, q_2$, $q_1 + q_2 = N$,
$\epsilon > 0$

Iteration:

Step 1 \textbf{compute} distances from $c_1, c_2$ for all $x \in \mathcal{D}$
Step 2 \textbf{update} the cluster sizes $q_1^+, q_2^+$
Step 3 \textbf{update} the centers $c_1^+, c_2^+$
Step 4 \textbf{if} $\|c_1^+ - c_1\| + \|c_2^+ - c_2\| < \epsilon$ \textbf{stop}
\textbf{return} to step 1
The PDQ Algorithm: Comments

(a) Step 2 is absent if \( q_1, q_2 \) are given, or do not matter.

(b) If Step 2 is omitted the algorithm tends to generate clusters of equal size.

(c) If the Mahalanobis distance is used, the covariance matrix \( \Sigma_k \) of the \( k \)-th cluster, can be estimated at each iteration by

\[
\Sigma_k = \frac{\sum_{i=1}^{N} u_k(x_i)(x_i - c_k)(x_i - c_k)^T}{\sum_{i=1}^{N} u_k(x_i)}
\]
Mixture models

Given observations from a density $\phi(x)$, that is itself a mixture of two densities,

$$\phi(x) = \pi \phi_1(x) + (1 - \pi) \phi_2(x),$$

it is required to estimate the weight $\pi$, and the relevant parameters of the distributions $\phi_1$ and $\phi_2$. 
Mixture models

Given observations from a density \( \phi(x) \), that is itself a mixture of two densities,

\[
\phi(x) = \pi \phi_1(x) + (1 - \pi) \phi_2(x),
\]

it is required to estimate the weight \( \pi \), and the relevant parameters of the distributions \( \phi_1 \) and \( \phi_2 \).

A common situation is Gaussian mixture, i.e., the distribution \( \phi \) is a mixture of normal distributions \( \phi_k \), each with its mean \( c_k \) and covariance \( \Sigma_k \) that need to be estimated.
Mixture models

Given observations from a density $\phi(x)$, that is itself a mixture of two densities,

$$\phi(x) = \pi \phi_1(x) + (1 - \pi) \phi_2(x),$$

it is required to estimate the weight $\pi$, and the relevant parameters of the distributions $\phi_1$ and $\phi_2$.

A common situation is Gaussian mixture, i.e., the distribution $\phi$ is a mixture of normal distributions $\phi_k$, each with its mean $c_k$ and covariance $\Sigma_k$ that need to be estimated.

The EM method is commonly used in the applications.
Comparison of PDQ & EM

(a) The EM Algorithm is based on maximum likelihood, and therefore depends on the density functions in the mix, requiring different computations for different densities.
Comparison of PDQ & EM

(a) The EM Algorithm is based on maximum likelihood, and therefore depends on the density functions in the mix, requiring different computations for different densities.

(b) In each EM iteration the density functions must be evaluated, requiring in Step 1 $K N$ function evaluations where $K$ is the number of densities in the mixture.
**Comparison of PDQ & EM**

(a) The EM Algorithm is based on maximum likelihood, and therefore depends on the density functions in the mix, requiring different computations for different densities.

(b) In each EM iteration the density functions must be evaluated, requiring in Step 1 $K N$ function evaluations where $K$ is the number of densities in the mixture.

(c) Because the EM iterations are costly, it is common to use another method, e.g., the $K$–means method, as a pre–processor, to get closer to the centers before starting EM.
Comparison of PDQ & EM

(a) The EM Algorithm is based on maximum likelihood, and therefore depends on the density functions in the mix, requiring different computations for different densities.

(b) In each EM iteration the density functions must be evaluated, requiring in Step 1 $KN$ function evaluations where $K$ is the number of densities in the mixture.

(c) Because the EM iterations are costly, it is common to use another method, e.g., the $K$–means method, as a pre–processor, to get closer to the centers before starting EM.

(d) If the mixing distributions are correctly assumed, the EM method gives precise estimates of the parameters.
Example: Unequal clusters
Comparison of PDQ and EM
Density estimation

Contours of mixture density

Contours of the JDF
SEMI–SUPERVISED CLUSTERING
**“Good” and “bad” data sets**

<table>
<thead>
<tr>
<th>Name of Data Set</th>
<th>% Correct Predictions</th>
<th>% Errors</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>96.5</td>
<td>100</td>
<td>93.1</td>
</tr>
<tr>
<td>Liver</td>
<td>63.2</td>
<td>79.3</td>
<td>49.7</td>
</tr>
<tr>
<td>Diabetes</td>
<td>74.7</td>
<td>79.9</td>
<td>65.7</td>
</tr>
<tr>
<td>Voting</td>
<td>92.0</td>
<td>98.78</td>
<td>82.3</td>
</tr>
<tr>
<td>Wine</td>
<td>93.7</td>
<td>100</td>
<td>82.35</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>86.03</td>
<td>96.42</td>
<td>71.43</td>
</tr>
</tbody>
</table>

Semi–supervised clustering

Clustering is modeled by the optimization problem

$$\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) p_k(x)^2$$
Semi–supervised clustering

Clustering is modeled by the optimization problem

$$\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) p_k(x)^2$$

Assume probabilistic labels (given)

$$r_k(x) = \text{Prob} \{ x \in C_k \}, \ k \in \overline{1, K}.$$ 

Classification corresponds to the (trivial) optimization problem

$$\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) (p_k(x) - r_k(x))^2$$
Semi–supervised clustering

Clustering is modeled by the optimization problem

$$
\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) p_k(x)^2
$$

Assume probabilistic labels (given)

$$
r_k(x) = \text{Prob}\{x \in C_k\}, \ k \in 1, K.
$$

Classification corresponds to the (trivial) optimization problem

$$
\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) (p_k(x) - r_k(x))^2
$$

Semi–supervised clustering is the parameterized problem

$$
\min \sum_{i=1}^{N} \sum_{k=1}^{K} d_k(x) ((1 - \theta) p_k(x)^2 + \theta (p_k(x) - r_k(x))^2)
$$

where $\theta = \text{reliability of prior information}$. 
Semi–supervised clustering

For any \( x \) in the training set \( \mathcal{T} \) consider the problem,

\[
\min_{p_1, p_2} \quad (1 - \theta) \left( d_1 p_1^2 + d_2 p_2^2 \right) + \theta \left( (p_1 - r_1)^2 d_1 + (p_2 - r_2)^2 d_2 \right)
\]

s.t. \( p_1 + p_2 = 1 \)

\( p_1, p_2 \geq 0 \)

where

\( r_i = r_i(x) \) (the label at \( x \)), and the parameter

\( \theta \in [0, 1] \) indicates the reliability of the prior information.
Semi–supervised clustering

For any $x$ in the training set $T$ consider the problem,

$$\min_{p_1, p_2} \quad (1 - \theta) (d_1 p_1^2 + d_2 p_2^2) + \theta ((p_1 - r_1)^2 d_1 + (p_2 - r_2)^2 d_2)$$

s.t. \quad p_1 + p_2 = 1

$p_1, p_2 \geq 0$

where

$r_i = r_i(x)$ (the label at $x$), and the parameter

$\theta \in [0, 1]$ indicates the reliability of the prior information.

The Lagrangian of this problem is

$$L(p_1, p_2, \lambda) = (1 - \theta) (d_1 p_1^2 + d_2 p_2^2) + \theta ((p_1 - r_1)^2 d_1 + (p_2 - r_2)^2 d_2)$$

$$+ \lambda (1 - p_1 - p_2)$$
Zeroing the gradient (with respect to $p_1$, $p_2$) we get

\[
2(1 - \theta)p_1d_1 + 2\theta(p_1 - r_1) = \lambda, \\
2(1 - \theta)p_2d_2 + 2\theta(p_2 - r_2) = \lambda,
\]

\[
\therefore p_1 = \frac{\lambda + 2\theta r_1d_1}{2d_1}, \quad p_2 = \frac{\lambda + 2\theta r_2d_2}{2d_2},
\]

Since the probabilities $p_1, p_2$ add to 1, we get

\[
\frac{\lambda + 2\theta r_1d_1}{2d_1} + \frac{\lambda + \theta r_2d_2}{2d_2} = 1,
\]

\[
\therefore \lambda = 2(1 - \theta)\frac{d_1d_2}{d_1 + d_2}.
\]

\[
\therefore p_1 = (1 - \theta)\frac{d_2}{d_1 + d_2} + \theta r_1, \quad p_2 = (1 - \theta)\frac{d_1}{d_1 + d_2} + \theta r_2.
\]
For $x \in T$ (labelled points) the extremal problem is

$$
\min_{p_1, p_2} \ (1 - \theta) \sum_x (d_1 p_1^2 + d_2 p_2^2) + \theta \sum_x ((p_1 - r_1)^2 d_1 + (p_2 - r_2)^2 d_2)
$$

$$
-\nabla_{c_1} = (1 - \theta) \sum_x p_1^2 \frac{x - c_1}{d_1} + \theta \sum_x (p_1 - r_1)^2 \frac{x - c_1}{d_1}
$$

$$
c_1 = \sum_{i=1}^N \left( \frac{u_1(x_i)}{\sum_{j=1}^N u_1(x_j)} \right) x_i
$$

where $u_1(x_i) = (1 - \theta) \frac{p_1^2}{d_1} + \theta \frac{(p_1 - r_1)^2}{d_1}$, in particular,

$$
u_1(x_i) = \begin{cases} 
\left( \frac{d_2}{d_1 + d_2} \right)^2 / d_1, & \theta = 0, \\
\left( \frac{r_1^2}{d_1} \right), & \theta \to 1
\end{cases}
$$
Comparison of datasets for $\theta \in [0, 1]$

Wisconsin breast–cancer dataset (L):

% of correct classifications is insensitive to $\theta$.
Classification uncertainty is monotone in $\theta$.

Diabetes dataset (R): The opposites hold.
REFERENCES
References

References

• A. B–I and C. Iyigun, **Probabilistic distance clustering**, *Journal of Classification* **25**(2008), 5-26

• C. Iyigun and A. B–I, **Probabilistic distance clustering adjusted for cluster size**, *Probability in Engineering and Informational Sciences* **22**(2008), 1-19
References

• A. B–I and C. Iyigun, Probabilistic distance clustering, *Journal of Classification* 25(2008), 5-26

• C. Iyigun and A. B–I, Probabilistic distance clustering adjusted for cluster size, *Probability in Engineering and Informational Sciences* 22(2008), 1-19

References

• A. B–I and C. Iyigun, Probabilistic distance clustering, *Journal of Classification* 25(2008), 5-26

• C. Iyigun and A. B–I, Probabilistic distance clustering adjusted for cluster size, *Probability in Engineering and Informational Sciences* 22(2008), 1-19


References

• A. B–I and C. Iyigun, Probabilistic distance clustering, *Journal of Classification* **25**(2008), 5-26

• C. Iyigun and A. B–I, Probabilistic distance clustering adjusted for cluster size, *Probability in Engineering and Informational Sciences* **22**(2008), 1-19


References downloaded at
http://benisrael.net/DATA-SETS.html
References downloaded at
http://benisrael.net/DATA-SETS.html

Thanks for your kind attention
References downloaded at
http://benisrael.net/DATA-SETS.html

Thanks for your kind attention

Questions?
References downloaded at
http://benisrael.net/DATA-SETS.html

Thanks for your kind attention

Questions?

adi.benisrael@gmail.com
The clusters must be “nice”