❖ Simulated Annealing

- It guarantees (under certain conditions) in probability, the computation of the globally optimal solution of the problem at hand via the minimization of a cost function J.
- It may escape from local minima since it allows moves that temporarily may increase the value of J.

Definitions

- An important parameter of the algorithm is the "temperature" T, which starts at a high value and reduces gradually.

Notation

- T_{max} is the initial value of the temperature T.
- $\mathbf{C}_{init}$ is the initial clustering.
- $\mathbf{C}$ is the current clustering.
- t is the current sweep.

- The algorithm:
  - Set $T=T_{max}$ and $\mathbf{C}=\mathbf{C}_{init}$.
  - $t=0$
  - Repeat
    - $t=t+1$
    - Repeat
      - Compute $J(\mathbf{C})$
      - Produce a new clustering, $\mathbf{C}'$, by assigning a randomly chosen vector from X to a different cluster.
      - Compute $J(\mathbf{C}')$
      - If $\Delta J=J(\mathbf{C}')-J(\mathbf{C})<0$ then
        - * (A) $\mathbf{C}=\mathbf{C}'$
      - Else
        - * (B) $\mathbf{C}=\mathbf{C}'$, with probability $P(\Delta J)=e^{-\Delta J/T}$.
      - End if
    - Until an equilibrium state is reached at this temperature.
    - $T=f(T_{max},t)$
  - Until a predetermined value T_{min} for T is reached.

❖ Simulated Annealing (cont.)

➢ Remarks:

- For $T \to \infty$, $p(\Delta J) \approx 1$. Thus almost all movements of vectors between clusters are allowed.
- For lower values of T fewer moves of type (B) (from lower to higher cost clusterings) are allowed.
- As $T \to 0$ the probability of moves of type (B) tends to zero.

- We assume that the equilibrium state has been reached

  "If for k successive random reassignments of vectors, $\mathbf{C}$ remains unchanged."

- A schedule for lowering T that guarantees convergence to the global minimum with probability 1, is

$$T = T_{max} / \ln(1+t)$$

❖ Clustering using genetic algorithms (GA)

A few hints concerning genetic algorithms

- They have been inspired by the natural selection mechanism (Darwin).
- They consider a population of solutions of the problem at hand and they perform certain operators to this, so that the new population is improved compared to the previous one (in terms of a criterion function J).
- The solutions are coded and the operators are applied on the coded versions of the solutions.

The most well-known operators are:

- Reproduction:
  - It ensures that, in probability, the better a solution in the current population is, the more replicates it has in the next population.
- Crossover:
  - It applies to the temporary population produced after the application of the reproduction operator.
  - It selects pairs of solutions randomly, splits them at a random position and exchanges their second parts.

❖ Clustering using genetic algorithms (cont.)

- Mutation:
  - It applies after the reproduction and the crossover operators.
  - It selects randomly an element of a solution and alters it with some probability.
  - It may be viewed as a way out of getting stuck in local minima.

Aspects/Parameters that affect the performance of the algorithm

- The coding of the solutions.
- The number of solutions in a population, p.
- The probability with which we select two solutions for crossover.
- The probability with which an element of a solution is mutated.

❖ Clustering using genetic algorithms (cont.)

GA Algorithmic scheme

- $t=0$
- Choose an initial population $\wp_t$ of solutions.
- Repeat
  - Apply reproduction on $\wp_t$ and let $\wp_t'$ be the resulting temporary population.
  - Apply crossover on $\wp_t'$ and let $\wp_t''$ be the resulting temporary population.
  - Apply mutation on $\wp_t''$ and let $\wp_{t+1}$ be the resulting population.
  - $t=t+1$
- Until a termination condition is met.
- Return
  - either the best solution of the last population,
  - or the best solution found during the evolution of the algorithm.

❖ Clustering using genetic algorithms (cont.)

Genetic Algorithms in Clustering

The characteristics of a simple GA hard clustering algorithm suitable for compact clusters, whose number m is fixed, is discussed next.

- A (not unique) way to code a solution is via the cluster representatives.

$$[\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_m]$$

- The cost function in use is

where $$J = \sum_{i=1}^{N} u_{ij} d(\underline{x}_i, \underline{w}_j)$$

$$u_{ij} = \begin{cases} 1, & if \;\; d(\underline{x}_i, \underline{w}_j) = \min_{k=1,\ldots m} d(\underline{x}_i, \underline{w}_k) \\ 0, & \text{otherwise} \end{cases}, \quad i = 1,\ldots, N$$

- The allowable cut points for the crossover operator are between different representatives.
- The mutation operator selects randomly a coordinate and decides randomly to add a small random number to it.

❖ Clustering using genetic algorithms (cont.)

➢ Remark:

- An alternative to the above scheme results if prior to the application of the reproduction operator, the hard clustering algorithm (GHAS), described in the previous chapter, runs p times, each time using a different solution of the current population as the initial state. The p resulting solutions constitute the population on which the reproduction operator will be applied.

DENSITY-BASED ALGORITHMS FOR LARGE DATA SETS

These algorithms:

- Consider clusters as regions in the l-dimensional space that are "dense" in points of X.
- Have, in principle, the ability to recover arbitrarily shaped clusters.
- Handle efficiently outliers.
- Have time complexity less than $O(N^2)$.

Typical density-based algorithms are:

- DBSCAN algorithm
- DBCLASD algorithm
- DENCLUE algorithm

❖ The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Algorithm.

The "density" around a point $\underline{x}$ is estimated as the number of points in X that fall inside a certain region of the l-dimensional space surrounding $\underline{x}$.

Notation

- $V_\varepsilon(\underline{x})$ is the hypersphere of radius ε centered at $\underline{x}$.
- $N_\varepsilon(\underline{x})$ the number of points of X in $V_\varepsilon(\underline{x})$.
- q is the minimum number of points of X that must be contained in $V_\varepsilon(\underline{x})$, in order for $\underline{x}$ to be considered an "interior" point of a cluster.

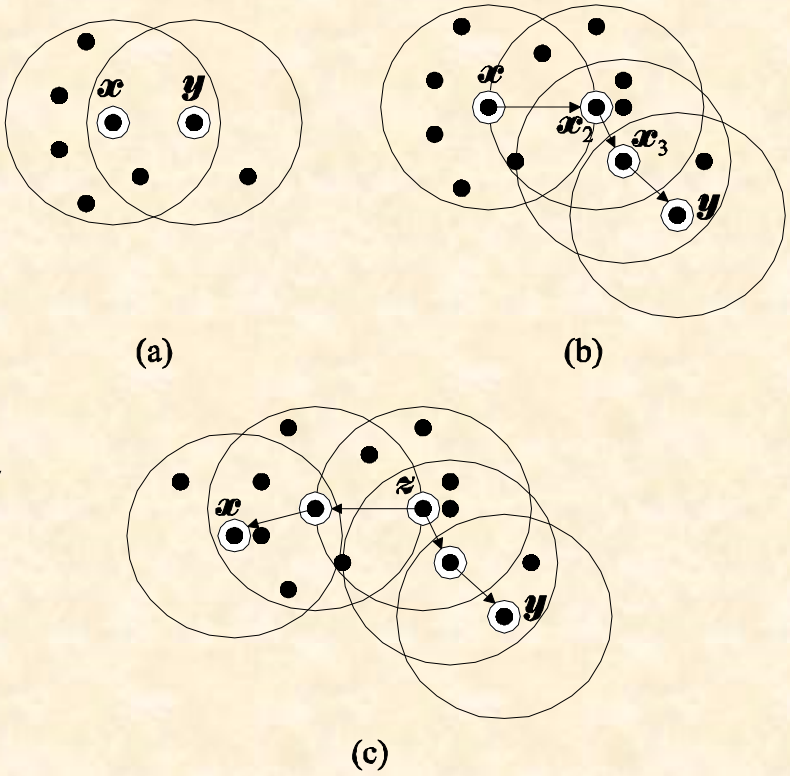
Definitions

1. A point $\underline{y}$ is directly density reachable from a point $\underline{x}$ if
   (i) $\underline{y} \in V_\varepsilon(\underline{x})$
   (ii) $N_\varepsilon(\underline{x}) \geq q$ (fig. (a)).
2. A point $\underline{y}$ is density reachable from a point $\underline{x}$ in X if there is a sequence of points $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p \in X$, with $\underline{x}_1 = \underline{x}$, $\underline{x}_p = \underline{y}$, such that $\underline{x}_{i+1}$ is directly density reachable from $\underline{x}_i$ (fig. (b)).

❖ The DBSCAN Algorithm (cont.)

3. A point x is density connected to a point $y \in X$ if there exists $z \in X$ such that both x and y are density reachable from z (fig. (c)).

➢Example 4: Assuming that $q=5$, (a) y is directly density reachable from x, but not vice versa, (b) y is density reachable from x, but not vice versa, and (c) x and y are density connected (in addition, y is density reachable from x, but not vice versa).



❖ The DBSCAN Algorithm (cont.)

4. A cluster C in DBSCAN is defined as a nonempty subset of X satisfying the following conditions:
   - If $\underline{x}$ belongs to C and $\underline{y} \in X$ is density reachable from $\underline{x}$, then $\underline{y} \in C$.
   - For each pair $(\underline{x},\underline{y}) \in C$, $\underline{x}$ and $\underline{y}$ are density connected.

5. Let $C_1, \ldots, C_m$ be the clusters in X. The set of points that are not connected in any of the $C_1, \ldots, C_m$ is known as noise.

6. A point $\underline{x}$ is called a core point if it has at least q points in its neighborhood. A noncore point may be either
   - a border point of a cluster (that is, density reachable from a core point) or
   - a noise point (that is, not density reachable from other points in X).

❖ The DBSCAN Algorithm (cont.)

Proposition 1: If x is a core point and D is the set of points in X that are density reachable from x, then D is a cluster.

Proposition 2: If C is a cluster and x is a core point in C, then C equals to the set of the points $y \in X$ that are density reachable from x.

Therefore: A cluster is uniquely determined by any of its core points.

Notation

- X_{un} is the set of points in X that have not been considered yet.
- m denotes the number of clusters.

❖ The DBSCAN Algorithm (cont.)

DBSCAN Algorithm

- Set $X_{un}=X$
- Set $m=0$
- While $X_{un} \neq \varnothing$ do
  - Arbitrarily select a $\underline{x} \in X_{un}$
  - If $\underline{x}$ is a noncore point then
    - Mark $\underline{x}$ as noise point
    - $X_{un}=X_{un}-\{\underline{x}\}$
  - End if
  - If $\underline{x}$ is a core point then
    - $m=m+1$
    - Determine all density-reachable points in X from $\underline{x}$.
    - Assign $\underline{x}$ and the previous points to the cluster C_m. The border points that may have been marked as noise are also assigned to C_m.
    - $X_{un}=X_{un}-C_m$
  - End {if}
- End {while}

❖ The DBSCAN Algorithm (cont.)

➢ Important notes:

- If a border point y of a cluster C is selected, it will be marked initially as a noise point. However, when a core point x in C will be selected later on, y will be identified as a density-reachable point from x and it will assigned to C.
- If a noise point y is selected, it will be marked as such and since it is not density reachable by any of the core points in X, its "noise" label will remain unaltered.

➢ Remarks:

- The parameters ε and q influence significantly the performance of DBSCAN. These should selected such that the algorithm is able to detect the least "dense" cluster (experimentation with several values for ε and q should be carried out).
- Implementation of DBSCAN using R*-tree data structure can achieve time complexity of $O(Nlog_2N)$ for low-dimensional data sets.
- DBSCAN is not well suited for cases where clusters exhibit significant differences in density as well as for applications of high-dimensional data.

❖ The DENsity-based CLUstEring (DENCLUE) Algorithm

Definitions

The influence function $f^{\underline{y}}(\underline{x})$ for a point $\underline{y} \in X$ is a positive function that decays to zero as $\underline{x}$ "moves away" from $\underline{y}$ ($d(\underline{x},\underline{y}) \to \infty$). Typical examples are:

$$f^{\underline{y}}(\underline{x}) = \begin{cases} 1, & \text{if } d(\underline{x},\underline{y}) < \sigma \\ 0, & \text{otherwise} \end{cases}, \qquad f^{\underline{y}}(\underline{x}) = e^{-\frac{d(\underline{x},\underline{y})^2}{2\sigma^2}}$$

where σ is a user-defined function.

The density function based on X is defined as (cf. the Parzen windows):

$$f^{X}(\underline{x}) = \sum_{i=1}^{N} f^{\underline{x}_i}(\underline{x})$$

The Goal:

(a) Identify all "significant" local maxima, $\underline{x}_j^*$, $j=1,\ldots,m$ of $f^X(\underline{x})$

(b) Create a cluster C_j for each $\underline{x}_j^*$ and assign to C_j all points of X that lie within the "region of attraction" of $\underline{x}_j^*$.

❖ The DENCLUE Algorithm (cont.)

Two clarifications

- The region of attraction of $\underline{x}_j^*$ is defined as the set of points in $\underline{x} \in \Re^l$ such that if a "hill-climbing" (such as the steepest ascent) method is applied, initialized by $\underline{x}$, it will terminate arbitrarily close to $\underline{x}_j^*$.
- A local maximum is considered as significant if $f^X(\underline{x}_j^*) \geq \xi$ (ξ is a user-defined parameter).

Approximation of $f^X(\underline{x})$

$$\hat{f}^X(\underline{x}) = \sum_{\underline{x}_i \in Y(\underline{x})} f^{\underline{x}_i}(\underline{x})$$

where $Y(\underline{x})$ is the set of points in X that lie "close" to $\underline{x}$.

The above framework is used by the DENCLUE algorithm.

❖ The DENCLUE Algorithm (cont.)

DENCLUE algorithm

- *Preclustering stage* (identification of regions dense in points of X)
  - Apply an l-dimensional grid of edge-length 2σ in the $\mathcal{R}^l$ space.
  - Determine the set D_p of the hypercubes that contain at least one point of X.
  - Determine the set $D_{sp} (\subset D_p)$ that contains the "highly populated" cubes of D_p (Cubes that contain at least $\xi_c > 1$ points of X).
  - For each $c \in D_{sp}$ define a connection with all neighboring cubes c_j in D_p for which $d(\underline{m}_c, \underline{m}_{c_j})$ is no greater than 4σ, where $\underline{m}_c$, $\underline{m}_{c_j}$ are the means of c and c_j, respectively.
- *Main stage*
  - Determine the set D_r that contains:
    - the highly populated cubes and
    - the cubes that have at least one connection with a highly populated cube.
  - For each point $\underline{x}$ in a cube $c \in D_r$ determine $Y(\underline{x})$ as the set of points that belong to cubes c_j in D_r such that the mean values of c_js lie at distance less than $\lambda\sigma$ from $\underline{x}$ (typically $\lambda = 4$).

❖ The DENCLUE Algorithm (cont.)

DENCLUE algorithm (cont.)

- For each point $\underline{x}$ in a cube $c \in D_r$
  - Apply a hill climbing method starting from $\underline{x}$ and let $\underline{x}^*$ be the local maximum to which the method converges.
  - If $\underline{x}^*$ is a significant local maximum ($f^X(\underline{x}^*) \geq \xi$) then
    - If a cluster C associated with $\underline{x}^*$ has already been created then
      - $\underline{x}$ is assigned to C
    - Else
      - Create a cluster C associated with $\underline{x}^*$
      - Assign $\underline{x}$ to C
    - End if
  - End if
- End for

❖ The DENCLUE Algorithm (cont.)

➢ Remarks:

- Shortcuts allow the assignment of points to clusters, without having to apply the hill-climbing procedure.
- DENCLUE is able to detect arbitrarily shaped clusters.
- The algorithm deals with noise very satisfactory.
- The worst-case time complexity of DENCLUE is $O(Nlog_2N)$.
- Experimental results indicate that the average time complexity is $O(log_2N)$.
- It works efficiently with high-dimensional data.

CLUSTERING ALGORITHMS FOR HIGH-DIMENSIONAL DATA SETS

- What is a high-dimensionality space?

  Dimensionality l of the input space with

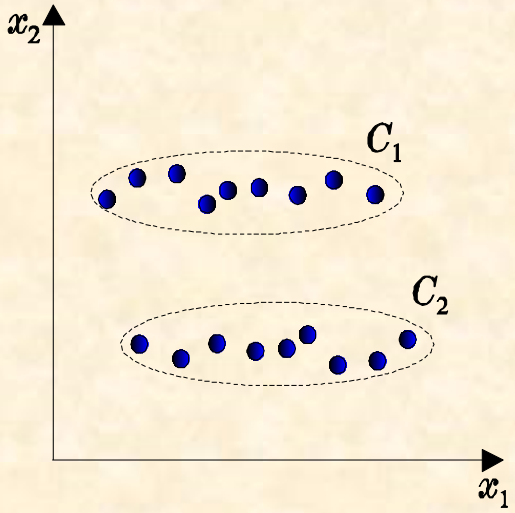
  $$20 \leq l \leq \text{few thousands}$$

  indicate high-dimensional data sets.

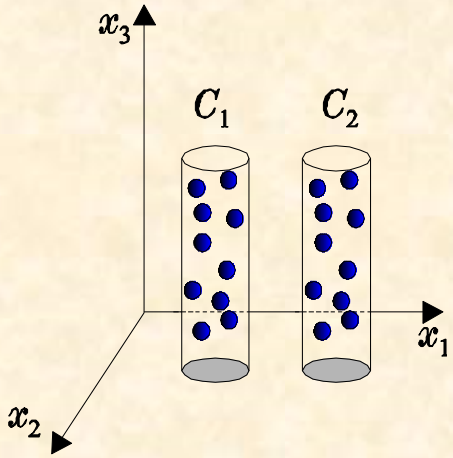
- Problems of considering simultaneously all dimensions in high-dimensional data sets:
  - "Curse of dimensionality". As a fixed number of points spread out in high-dimensional spaces, they become almost equidistant (that is, the terms similarity and dissimilarity tend to become meaningless).
  - Several dimensions may be irrelevant to the identification of the clusters (that is, the clusters usually are identified in subspaces of the original feature space).

- A way out: *Work on subspaces of dimension lower than l.*
  - Main approaches:
    - Dimensionality reduction clustering approach.
    - Subspace clustering approach.

An example:



(a)



(b)

❖ Dimensionality Reduction Clustering Approach

Main idea

- Identify an appropriate l'-dimensional space $H_{l'}$ ($l' < l$).
- Project the data points in X into $H_{l'}$.
- Apply a clustering algorithm on the projections of the points of X into $H_{l'}$.

Identification of $H_{l'}$ may be carried out using either by:

- Feature generation methods,
- Feature selection methods,
- Random projections.

❖ Dimensionality Reduction Clustering Approach (cont.)

Feature generation methods

- In general, these methods preserve the distances between the points in the high-dimensional space, when these are mapped to the lower-dimensional space.
- They are very useful in producing compact representations of the original high-dimensional feature space.

- They are useful in cases where a significant number of features contributes to the identification of the clusters.
- Typical Methods in this category are:
  - Principal component analysis (PCA).
  - Singular value decomposition (SVD).
  - Nonlinear PCA.
  - Independent component analysis (ICA).

❖ Dimensionality Reduction Clustering Approach (cont.)

Feature selection methods

- They identify the features that are the main contributors to the formation of the clusters.
- The criteria used to evaluate the "goodness" of a specific subset of features follow either the
  - Wrapper model (The clustering algorithm is first chosen and a set of features F_i is evaluated through the results obtained from the application of the algorithm to X, where for each point only the features in F_i are taken into account).
  - Filter model (The evaluation of a subset of features is carried out using intrinsic properties of the data, prior to the application of the clustering algorithm).
- They are useful when all clusters lie in the same subspace of the feature space.

❖ Dimensionality Reduction Clustering Approach (cont.)

Clustering using Random Projections:

Here $H_{l'}$ is identified in a random manner.

Note: The projection of an l-dimensional space to an l'-dimensional space ($l' < l$) is uniquely defined via an $l' \times l$ projection matrix A.

Issues to be addressed:

(a) The proper estimate of l'. Estimates of l' guarantee that the distances between the points of X, in the original data set will be preserved (with some distortion) after the projection to a randomly chosen l'-dimensional space, whose projection matrix is constructed via certain probabilistic rules

(b) The definition of the projection matrix A. Possible rules for constructing A are:

1. Set each of its entries equal to a value stemming from an i.i.d. zero mean, unit variance Gaussian distribution and then normalize each row to the unit length.
2. Set each entry of A equal to -1 or $+1$, with probability 0.5.
3. Set each entry of A equal to $+\sqrt{3}$, $-\sqrt{3}$ or 0, with probabilities $1/6$, $1/6$, $2/3$, respectively.

❖ Dimensionality Reduction Clustering Approach (cont.)

Having defined A:

- Project the points of X into $H_{l'}$
- Perform a clustering algorithm on the projections of the points of X into $H_{l'}$.

Problem: Different random projections may lead to totally different results.

Solution:

- Perform several random projections $H_{l'}$.
- Apply a clustering algorithm on the projections of X to each $H_{l'}$.
- Combine the clustering results and produce the final clustering.

A method in the above spirit is described next ($O(N^2)$).

Clustering using Random Projections

- Select l´.
- Generate $A_1,\ldots,A_r$ different projection matrices using the (b.1) rule given above.
- For s=1 to r
  - Run GMDAS for the s-th random projection of X.
  - Compute the probability that $\underline{x}_i$ belongs to the j-th cluster of the s projection, $P(C_j^s|\underline{x}_i)$, i=1,…,N, j=1,…,m_s.
  - Create the similarity matrix P^s, where $P_{ij}^{\ s}$ is the probability that $\underline{x}_i$ and $\underline{x}_j$ belong to the same cluster,

$$P_{ij}^s = \sum_{q=1}^{m_s} P(C_q^s \mid \underline{x}_i) P(C_q^s \mid \underline{x}_j)$$

- End for
- Compute the average proximity matrix P, so that P_{ij} is the average of $P_{ij}^{\ s}$ s, s=1,…,r.
- Apply GAS (actually its complete link version) on P.
- Plot the similarity between the closest pair of clusters at each iteration versus the number of iterations.
- Select the clustering that corresponds to the most abrupt decrease in the plot.