# 算法实行思路

## 一、k-medoid:

The most common realisation of *k*-medoid clustering is the **Partitioning Around Medoids (PAM)** algorithm. It works as follows:

1. Initialize: select[ *k* of the *n* data points as the medoids
2. Associate each data point to the closest medoid.
3. While the cost of the configuration decreases:
   1. For each medoid *m*, for each non-medoid data point *o*:
      1. Swap *m* and *o*, recompute the cost (sum of distances of points to their medoid)
      2. If the total cost of the configuration increased in the previous step, undo the swap

## 二、Affinity propagation：

Let *x*1 through *xn* be a set of data points, with no assumptions made about their internal structure, and let *s* be a function that quantifies the similarity between any two points, such that *s*(*xi*, *xj*) > *s*(*xi*, *xk*)  if *xi* is more similar to *xj* than to *xk*.

The algorithm proceeds by alternating two message passing steps, to update two matrices:[[1]](https://en.wikipedia.org/w/index.php?title=Affinity_propagation&oldid=716583354#cite_note-science-1)

* The "responsibility" matrix **R** has values *r*(*i*, *k*) that quantify how well-suited *xk* is to serve as the exemplar for *xi*, relative to other candidate exemplars for *xi*.
* The "availability" matrix **A** contains values *a*(*i*, *k*) represents how "appropriate" it would be for *xi* to pick *xk* as its exemplar, taking into account other points' preference for *xk* as an exemplar.

Both matrices are initialized to all zeroes, and can be viewed as [log-probability](https://en.wikipedia.org/wiki/Log-probability) tables. The algorithm then performs the following updates iteratively:

* First, responsibility updates are sent around:
* Then, availability is updated per
* and

## 三、agglomerative hierarchical clustering

Basic process:

* Assign each object to a separate cluster.
* Evaluate all pair-wise distances(similarity) between clusters
* Construct a distance(similarity) matrix using the distance values.
* Look for the pair of clusters with the shortest distance(similarity).
* Remove the pair from the matrix and merge them.
* Evaluate all distances from this new cluster to all other clusters, and update the matrix.
* Repeat until the distance matrix is reduced to a single element.

A very convenient formulation, in similarity terms, which embraces all the hierarchical methods mentioned so far, is the *Lance–Williams dissimilarity update formula*

different AHC method will choose different .

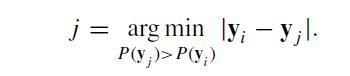
## 四、mean shift and quick shift:

In mean shift algorithm, we need to know the kernel density function( at first. Then we calculate the weighted mean of the density in the window:

where {\displaystyle N(x)}is the neighborhood of {\displaystyle x}x, a set of points for which {\displaystyle K(x\_{i})\neq 0}

The difference {\displaystyle m(x)-x} is called *mean shift* in Fukunaga and Hostetler. The *mean-shift algorithm* now sets {\displaystyle x\leftarrow m(x)}, and repeats the estimation until m(x) {\displaystyle m(x)}converges.

Quick shift algorithm is more easier:



You just need to find the most probability point with is closest to original cluster point at last step.