# Classification vs Clustering

|  |  |
| --- | --- |
| Classification | Clustering |
| Data have labels | Data don’t have label |
| Find a model to label data | Find an optimal cluster |

## Clustering

* Hierarchical clustering: bottom-up and top-down
* Distance between cluster (min,max,avg,centroid)

# K-means (Clustering)

## Algorithm

* Given:
* Initial: defined initial clusters
* Loop:
  + Compute mean of cluster :
  + For each (1)
  + Move to cluster m(i): (2)
  + Iterate until stopping criterion

## Proof of convergence:

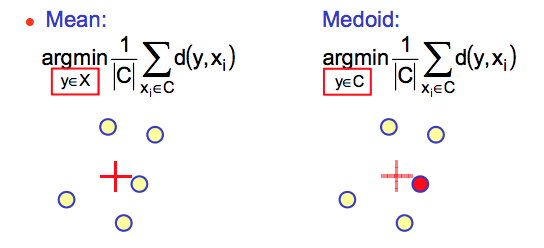
* : sum of distances of all nodes to their belonged cluster.
* (1)
* because (2)
* is decreasing during the iterations => reach **local minimum** after finite number of step
* toping criteria: no element move, after number of iterations or value

## Comment:

* Strength: fast
* Limits: require computation of mean (non-scalar data), specific k and sensitive with outliner

## K-medoids

* Use the most central object of cluster, robust to outliner

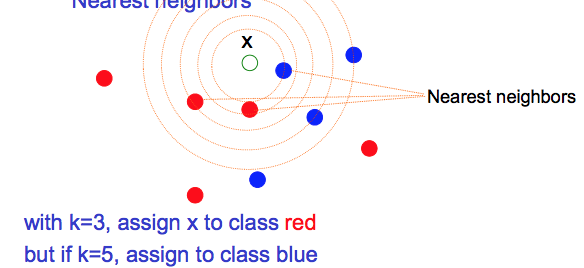


## Minimum Description Length

* Increase k, then decrease => not mean it will be better
* Minimum Description Length principle: penalty based on transmitting centroid information (regularization)
* MDK for k-means:

# K-Nearest Neighbors

* NN rule: assign x to class
* K-NN: assign x to most frequent class among k-NN



* Pros and Cons
  + Pros: flexible, simple, Good performance (large data), regression ( average instead of vote)
  + Cons: Needs lot of data, Intensive Computation, Hard to speed up (high dimension), difficult to choose weight/distance

# Bayesian Classification

* + P(C|x): posterior probability of class C given X
  + P(C): prior probability of class C
* Modeling
  + is generative model and is discrimative model
* Generative model
  + Can model classes independently
  + Easy to add one class

|  |
| --- |
| Bayesian |
| Input |
| Output Map point x to class y |
|  |
| Training \theta Maximum likelihood => find general distribution of dimension ( p(x\_i),p(y\_i)) |

<http://www.cs.columbia.edu/~mcollins/courses/6998-2012/lectures/lec6.1.pdf>

|  |
| --- |
| Gaussian Model |
| Input continuous valued attribute |
| P(x|C) = is calculated by data in that class |

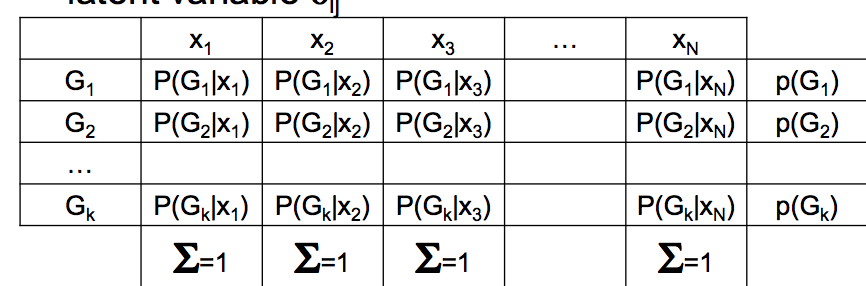
|  |
| --- |
| Gaussian Mixture Model |
| K Gaussian distribution |
| : Parameter need to optimize |
|  |

# EM- Algorithm

## Ideas

* Given and we want
* Using latent (non-observable) variable
* Algorithm
  + E-step: find expectation of latent variables values
  + M-step: compute maximum likelihood estimate of parameter value
  + Iterate

## EM in GMM



* E-step:
* -step: update \theta