DBSCAN

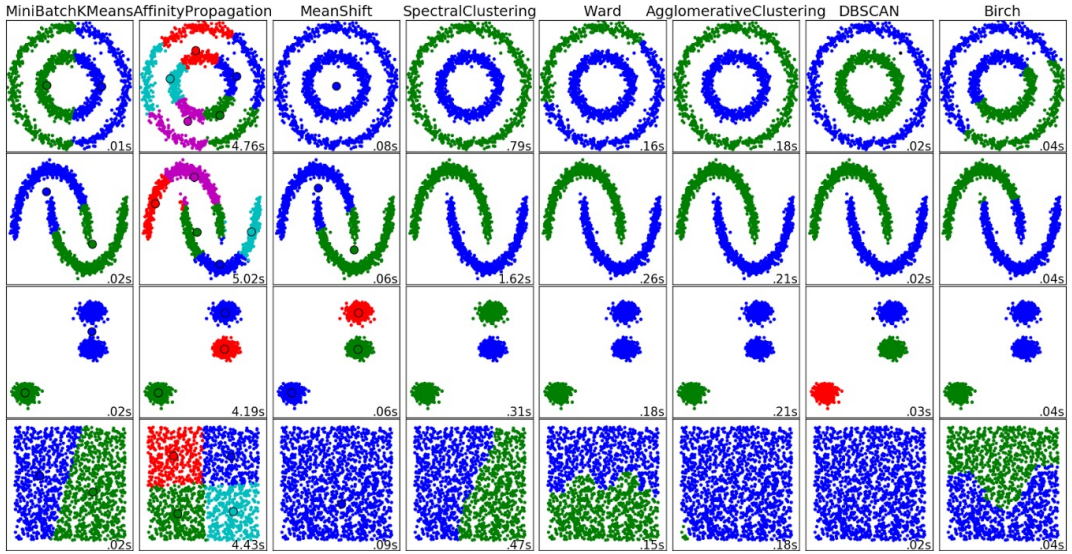
Density-based spatial clustering of applications with noise

Klustring

- Klustring är unsupervised machine learning metoder som separerar datapunkterna i flera olika grupper (kluster)
- Datapunkterna i samma kluster har liknande egenskaper
- Datapunkter i olika kluster har olika egenskaper i någon mening
- Får en intuition av mönster i datan
- Finns många olika modeller som har olika sätt att mäta distans mellan datapunkterna
- *Ex: K-Means (distance between points), Affinity propagation (graph distance), Mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centers), Spectral clustering (graph distance)*

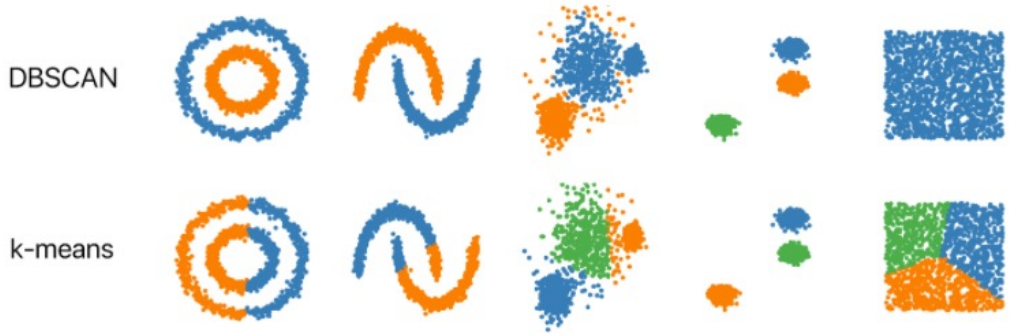
DBSCAN

- DBSCAN är en klustringsalgoritm
- Den grupperar data baserad på
  - Datapunkter som är nära varandra baserad på ett distansmått (euclidean)
  - Ett minimum antal punkter som måste ingå i klustret
- Den markerar datapunkter som är i områden med låg förekomst av andra datapunkter som outliers



DBSCAN VS K-Means

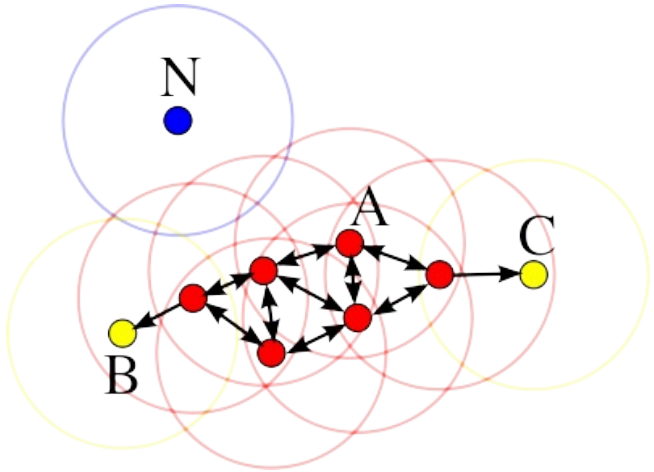
- K-Means kommer klustra alla datapunkter
- Trots datapunkten är långt ifrån klustren kommer den fortfarande tilldelas ett kluster
- Klustringsmetoder beror av genomsnittet av datapunkterna som igår i klustret
- Ett datapunkt långt ifrån alla andra datapunkter kan påverka utfallet av klustringen
- DBSCAN hanterar detta lite bättre än K-Means
- K-Means måste vi bestämma antal klusters K. Detta måste vi inte i DBSCAN



DBSCAN

Parameter

- En stor fördel med DBSCAN är man inte behöver ange antal klustre i förväg.
- Den är dock känslig på val av hyperparameter till modellen
- DBSCAN har två viktiga hyperparameter:
  - **epsilon (eps)** – anger hur nära datapunkter ska vara för att räknas som ett kluster. Är avståndet mellan två punkter lägre än epsilon räknas punkterna som grannar
  - **min_samples** – minimum antal datapunkter för att skapa ett kluster



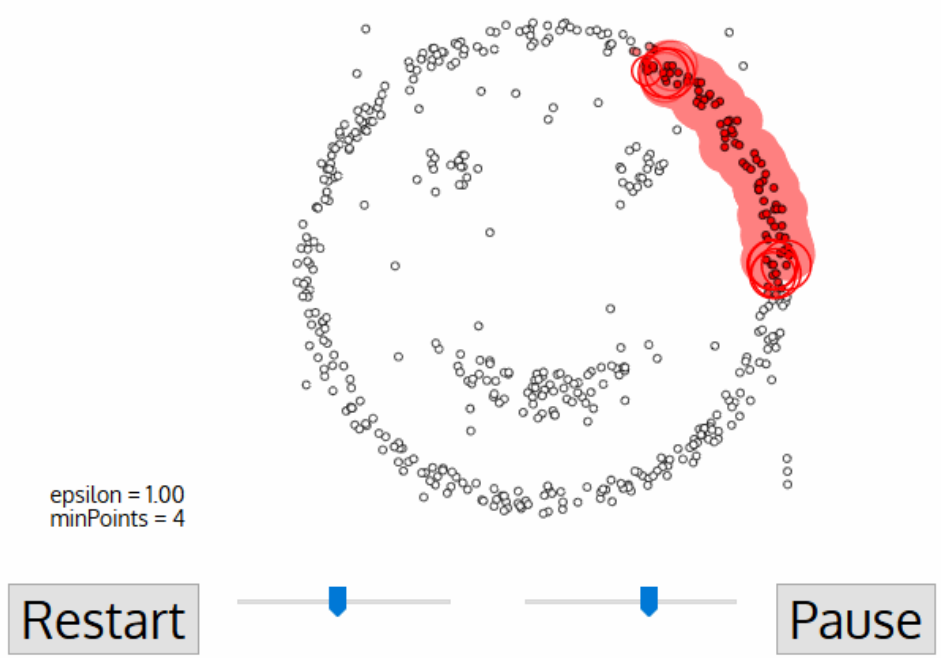
DBSCAN

Parameter – Hur tolka?

- epsilon
  - Om värdet är för litet kommer en stor del av datan inte klustras och räknas som outliers
  - Om värdet är för stort blir klustren för stora och växa ihop och de flesta datapunkter kommer hamna i samma kluster
  - Default är 0.5. Kom ihåg att standardisera datan innan klustring så att mean=0 och standard deviation=1
  - Generellt är det att föredra att ha små värden på epsilon
- min samples
  - Kan väljas efter dimensionen (D) till datasetet (antal features)
  - $min\_samples \leq D + 1$
  - Större värde är generellt bättre när det är mycket brus i datan. Välj större värde om det finns många datapunkter

Steg i DBSCAN

- Algoritmen godtyckligt väljer en punkt i datasetet (tills alla punkter har besökts)
- Om det finns åtminstone *min_samples* punkter inom en radie av *epsilon* till punkten, betraktar vi alla dessa punkter som en del av samma kluster
- Klustren utökas sedan genom att rekursivt upprepa beräkningen om grannar för varje grannpunkt



Evaluation metrics för klustring

- **Silhouette score**

$$s = \frac{b - a}{\max(a, b)}$$

- a genomsnittsavstånd mellan ett datapunkt och alla andra datapunkter i klustret
- b genomsnittsavstånd mellan ett datapunkt och alla andra datapunkter i nästa närmaste kluster
- Värde mellan -1 och 1
- -1 betyder felaktig klustering och 1 betyder tät klustring och kluster långt ifrån varandra (bra). 0 betyder överlappande kluster

Evaluation metrics för klustring

- **Davies-Bouldin score**
- Baserad på förhållandet mellan ”within-cluster” och ”between-cluster” avstånd
- $DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}$
- $R_{ij} = \frac{s_i + s_j}{d_{ij}}$
- R_{ij} är ”within-to-between cluster distance ratio” för *i* och *j* kluster
- s_i genomsnittsavstånd mellan varje datapunkt i kluster *i* och centroiden (mittpunkt i klustret)
- d_{ij} Euclidean avstånd mellan centroids till de två klustrerna
- Om två kluster är nära (liten) men har stor spridning blir ration stor som betyder att klustrenrna inte är så tydliga