K-means

Algorytm K-means jest metodą kwantyzacji wektorów która jest często używana w analizie klastrów. Zadaniem algorytmu jest podzielenie **n** danych na **k** klastrów, gdzie każda dana należy do klastra z najbliższym środkiem. Jest on jednym z algorytmów stosowanych w grupowaniu(klasteryzacji). Działanie algorytmu można opisać w 4 krokach

1. Wybór ilości klastrów, z góry musimy założyć ile będziemy ich chcieli mieć. Następnie umieszczamy zadaną ilość punktów w losowy sposób w przestrzeni.
2. Wyliczamy średnią odległość poszczególnych punktów i przypisujemy im najbliższą centroidę
3. Wyliczamy średnią arytmetyczną z punktów w poszczególnych grupach, następnie wynik zapisujemy jako nowe położenie centroidy.
4. Powtarzam krok 2 i 3 tak długo aż przynależność punktów do grup nie zmienia się wraz z iteracjami

Zaletami algorytmu K-means jest na pewno prostota w działaniu co przekłada się na wydajność. Dodatkowo przy dużej ilości danych i małej ilości grup algorytm będzie najszybciej zwracał wynik od innych algorytmów. Są dwie główne wady owego algorytmu, pierwsza to brak pomocy w określeniu ilości grup (K) a drugi poważniejszy to różne wyniki w zależności od wartości początkowych. Dodatkowo przez losowo dobierane centroidy, wynik będzie inny na tym samym zestawie danych. Tak jak pisałem k-means przyjmuje losowe wartości początkowe, więc przy pechowym doborze początkowym wyniki nie będą zadowalające, a często mogą być błędne. W tym momencie z pomocą przychodzi nam drugi algorytm k-means++. Jest to algorytm do wybierania początkowych wartości centroidów. Zasada działa k-means++ może być opisana w 5 krokach

1. Wybieramy 1 centroidę spośród danych w sposób losowy.
2. Dla każdej danej wyliczamy odległość pomiędzy daną a centroidą.
3. Wybieramy następną centroidę spośród danych aby prawdopodobieństwo wyboru punktu jako centroidy było proporcjonalne do dystansu do najbliższego, poprzednio wybranej centroidy. Zazwyczaj jest to punkt który jest najdalej oddalony od poprzedniej centroidy.
4. Powtarzamy 2 i 3 punkt aż nie otrzymamy K centroid.
5. Przechodzimy do algorytmu k-means pomijając pkt 1.

Przez swoją złożoność przy wyborze centroid k-means++ jest wolniejszy. Natomiast k-means++ gwarantuje nam znalezienie wyniku który ma złożoność O(log *k*) do optymalnego wyniku k-means.