Welke methode is optimaal voor het vinden van vergelijkbare patiënten in medische data?

Voorwaarden:

* Er kan gefilterd worden op input data (e.g. selecteer specifieke kolommen, selecteer een specifieke patiënt).
* Er kan gemeten worden hoe ver patiënten van elkaar afliggen in vergelijkbaarheid.
* Er kunnen n vergelijkbare patiënten worden opgevraagd

Omdat er relatief weinig te vinden is over wanneer welke methode geschikt is wordt er soms verwezen naar onbetrouwbare bronnen. Dit zijn dus niet per se harde conclusies en verder onderzoek kan waardevol zijn.

# Clustering vs klassieke statistische methodes

Quinn is op zoek gegaan naar verschillende clusteringsmethodes om vergelijkbare patiënten in een set van medische data te vinden. Clustering is hier, naar mijn mening, niet optimaal voor en ik stel voor om te kijken naar veel eenvoudigere statistische methodes [[3]](https://doi.org/10.1137/1.9781611972788.22)[[4]](https://www.iiisci.org/journal/pdv/sci/pdfs/GS315JG.pdf).

Wat zijn mogelijke **voordelen** van statistische methodes over clustering?

* Geen parameter voor het aantal clusters
* Sommige methodes (zoals bijv. Jaccard of Hamming similarity) zijn in staat om een similarity measure tussen patiënten op te stellen. Op deze manier zou je dus alle patiënten kunnen rangschikken op de vergelijkbaarheid met één specifieke patiënt. Het opvragen van n vergelijkbare patiënten is een stuk lastiger bij clustering wanneer n > de grootte van de cluster waarin de patiënt zich bevindt.
* Het zijn statistische methodes; geen ML-modellen. Er hoeft dus niks getraind te worden waar er voor clustering eventueel meerdere modellen getraind zouden moeten worden om om te gaan met missende data en met kolom filters (clustering modellen zijn ook erg gevoelig voor data imputation als je dat als alternatief zou gebruiken).
* **Je hebt de optie om te kiezen voor statistische methodes die positieve en negatieve matches in eigenschappen niet even zwaar laten meewegen**. “The inclusion or exclusion of negative matches, in the binary similarity measures have been an ongoing issue…

Sokal et al. argued that the negative matches do not mean necessarily any similarity between two objects. This is because an almost infinite number of attributes is possibly lacking in two objects.” [[4]](https://www.iiisci.org/journal/pdv/sci/pdfs/GS315JG.pdf)

* De berekeningen zelf zijn intuïtief en eenvoudig.

Wat zijn mogelijke **nadelen** van statistische methodes over clustering?

* De berekening moet live worden uitgevoerd (schaalt mogelijk slecht met veel data, maar kan deels worden verholpen door het gebruiken van een random subset en eventueel multithreading) of vooraf moet elke combinatie in een database worden opgeslagen.

# Problemen met het clusteringsonderzoek

Quinn zegt:

“Er is gekeken naar de volgende modellen: Kmeans, Affinity Propagation, Spectral clustering, Agglomorative Clustering, DBSCAN, OPTICS, BIRCH en Gaussian Mixture”

“Na onderzoek te doen naar de gekozen modellen zijn de volgende problemen gevonden:

* AffinityPropagation -- werkt niet op grote datasets
* SpectralClustering -- werkt niet met veel clusters
* AgglomerativeClustering -- geen mogelijkheid voor predict nieuwe data
* DBSCAN -- geen mogelijkheid voor predict nieuwe data
* OPTICS -- geen mogelijkheid voor predict nieuwe data

Dus blijven alleen **KMeans**, **BIRCH** en **GausianMixture** over”

Wat hier niet is meegenomen:

* GaussianMixture – “One of the main disadvantages of clustering with gaussian mixture models is that it is difficult to incorporate categorical variables. Gaussian mixture models operate under the assumption that all of your features are normally distributed, so they are not easily adapted to categorical data.” [[0]](https://crunchingthedata.com/when-to-use-gaussian-mixture-models/) Onze dataset bestaat echter grotendeels (op het moment zelfs volledig) uit non-gaussian categorical data. Nominale data is nooit normaal verdeeld.
* KMeans – “The standard k-means algorithm isn't directly applicable to categorical data, for various reasons. The sample space for categorical data is discrete and doesn't have a natural origin. A Euclidean distance function on such a space isn't really meaningful.” [[1]](https://datascience.stackexchange.com/questions/22/k-means-clustering-for-mixed-numeric-and-categorical-data)[[2]](https://www.daylight.com/meetings/mug04/Bradshaw/why_k-modes.html) Hier hebben we wederom hetzelfde probleem dat de methode niet compatible is met nominale data, deze keer omdat het de Euclidean distance tussen deze nominale variabelen, wat berekend moet worden voor het creëren van clusters, betekenisloos is.
* BIRCH – “However, BIRCH has one major drawback – it can only process metric attributes. A metric attribute is any attribute whose values can be represented in Euclidean space i.e., no categorical attributes should be present” [[5]](https://www.geeksforgeeks.org/ml-birch-clustering/)

**Geen van de onderzochte methodes zijn dus geschikt voor onze situatie**

Overige problemen:

* Er is niet gekeken of de clusters logisch zijn/ kloppen**, metrics zijn dus mogelijk gebaseerd op betekenisloze clusters. Waardoor getrokken conclusies ook niet zouden kloppen.**
* Er is niet gekeken naar k-modes/k-prototypes (vergelijkbaar met kmeans maar geschikt voor mixed/nominale data)
* Er is niet nagedacht over het omgaan met filters op de input (imputation? Multiple models?)
* Geen onderbouwing voor het aantal clusters (waarom geen silhouette score elleboogplot om overfitting tegen te gaan?)

Hier voorspellen we de clusters voor verschillende patiënten met KMeans en k-modes. Zoals je kan zien leidt dit tot een verschil in resultaten.

A screenshot of a black screen

Description automatically generated with low confidence

# Aanbevelingen

1. Bepaal welke distance measure geschikt is voor onze situatie specifiek. Beschouw je negatieve matches (e.g. twee patiënten hebben alle twee geen kanker) als een vorm van similarity? Dan zou je Hamming distance kunnen gebruiken. Wil je dit niet, dan kan je kijken naar bijvoorbeeld Jaccard distance.
2. Implementeer een statistische distance measure om patient similarity te bepalen. Dit lost praktisch alle problemen op behalve \*mogelijk\* de computational time required, dit vergt verder onderzoek.
3. Implementeer k-modes of k-prototypes (mixed datatypes) om similarity clusters te maken. Dit lost een deel van de problemen op maar de meeste problemen blijven.

# Functie voor patient similarity

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De gemaakte functie maakt gebruik van de scipy spatial distance module[6] om de similarity tussen een specifieke patiënt en elke andere patiënt te berekenen. De resultaten hiervan worden (in dictionaries) in een lijst opgeslagen en gesorteerd teruggegeven. De unique only bool zorgt ervoor dat er alleen wordt gekeken naar patiënten met verschillende eigenschappen. In de dictionaries wordt ook opgeslagen hoeveel patiënten met dezelfde eigenschappen voorkomen.



# Bronnen

[0] <https://crunchingthedata.com/when-to-use-gaussian-mixture-models/>

[1] <https://datascience.stackexchange.com/questions/22/k-means-clustering-for-mixed-numeric-and-categorical-data>

[2] <https://www.daylight.com/meetings/mug04/Bradshaw/why_k-modes.html>

[3] <https://doi.org/10.1137/1.9781611972788.22>

[4] <https://www.iiisci.org/journal/pdv/sci/pdfs/GS315JG.pdf>

[5] <https://www.geeksforgeeks.org/ml-birch-clustering/>

[6] https://docs.scipy.org/doc/scipy/reference/spatial.distance.html