FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Introdução

Este segundo projecto insere-se na área de aprendizagem não supervisionada. Este tipo de aprendizagem pretende ajustar um modelo à estrutura geral dos dados sem que este seja definido pelas *labels* dos exemplos. Neste caso, os algoritmos estudados serão de *clustering* que consiste em agrupar os dados em *clusters* de forma a maximizar a semelhança entre exemplos do mesmo grupo e minimizar entre grupos diferentes. Sendo que, no exemplo estudado, esta medida de semelhança será a distância, logo a relação será a inversa, pretendendo-se minimizar a distância entre exemplos do mesmo grupo e maximizar a de grupos diferentes.

Em relação ao problema apresentado, o objectivo será avaliar a performance de três algoritmos de *clustering* diferentes (K-Means, DBSCAN, Gaussian Mixture Models) com o intuito de agrupar eventos sísmicos. Este *dataset* contém informação dos sismos de uma magnitude superior a 6.5 nos últimos 100 anos sendo que foi adicionada uma nova coluna *fault* que indica a falha tectónica mais próximo de cada evento sísmico. É importante referir que o valor desta é -1, no caso de não existir uma falha significativamente próxima e que para uma falha ser considerada relevante terá que ter pelo menos 30 eventos sísmicos.

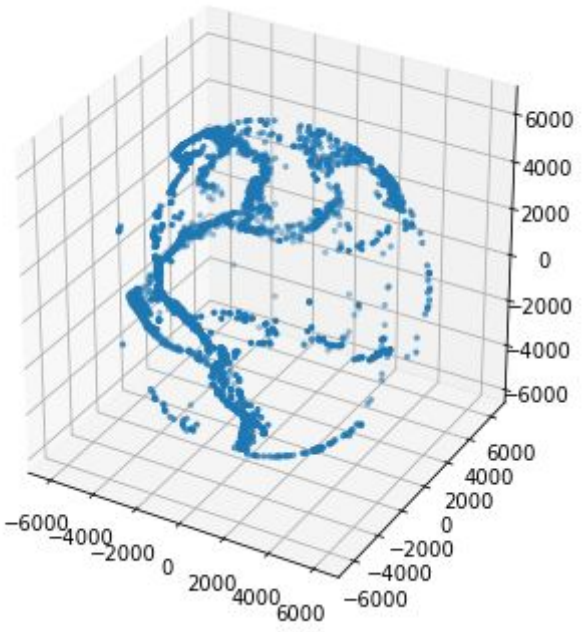


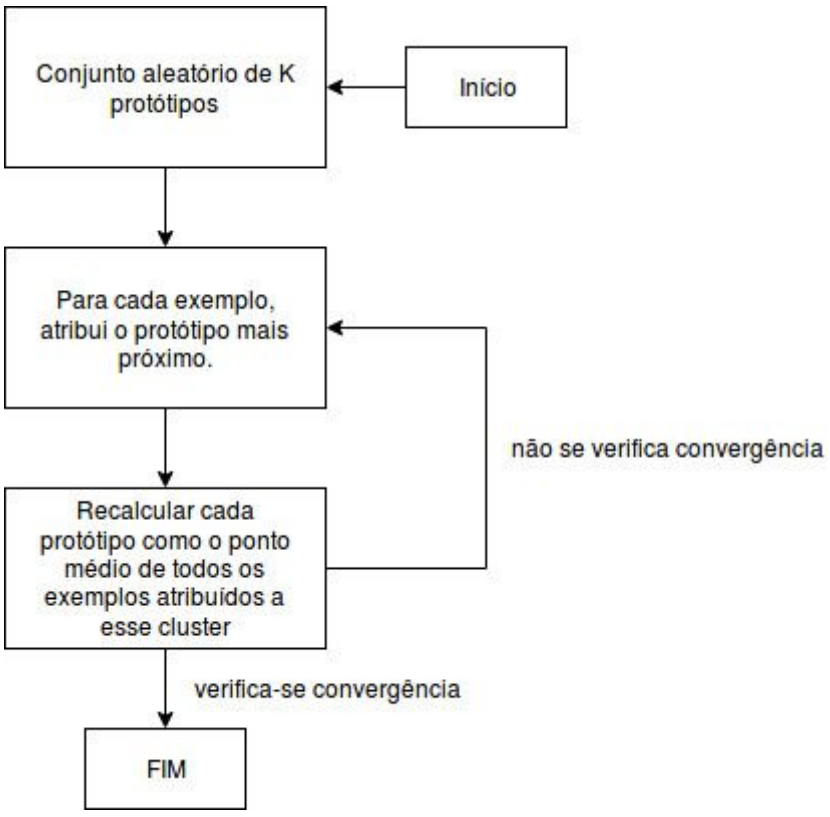
Figura 1 - Distribuição dos sismos

André Lopes nº 45617
Nelson Coquenim nº 45694

Conceitos teóricos

K-Means

O algoritmo K-Means consiste em particionar o conjunto de dados em K conjuntos (*clusters*), cada um definido pelo seu protótipo, neste caso, um centróide pois este ponto não faz, necessariamente, parte do conjunto de exemplos. Cada exemplo irá pertencer ao *cluster* cujo o protótipo é o mais próximo. Este algoritmo é **exclusivo** porque cada exemplo pertence apenas a um único *cluster*, **particional** uma vez que os exemplos foram divididos em *clusters* do mesmo nível, **baseado em protótipos** dado que cada *cluster* é caracterizado pelo seu protótipo, e **completo** pois todos os exemplos serão atribuídos a um *cluster*. O algoritmo é o seguinte:



André Lopes nº 45617
Nelson Coquenim nº 45694

DBSCAN (Density-based spatial clustering for applications with noise)

Este algoritmo foi criado aspirando resolver 3 principais problemas: clusters com uma forma arbitrária, a dificuldade em seleccionar os parâmetros, e a performance utilizando um volume de dados grande[2]. Sendo este **exclusivo**, **particional,** baseado na **densidade** dos pontos e **parcial** , pois nem todos os exemplos terão um cluster associado

Em relação à forma arbitrária dos *clusters*, analisou-se que quando se observa um conjunto de pontos, a razão pela qual se reconhece um *cluster* é que dentro de cada a densidade de pontos é maior que no exterior. Por outro lado, nas zonas de *noise* a densidade é menor que em qualquer outro *cluster*. Importa definir estas noções:

- considera-se vizinhança de um ponto (N_ε) todos os pontos dentro de uma distância ε;
- p é um ponto *core* se tem pelo menos **minPts** na sua vizinhança;
- q está ao alcance de p se é diretamente alcançável de p ou a partir de um *core* point p' que é alcançável a partir de p

```
DBSCAN algorithm
For each point p
   If |Nε(p)| <= MinPts
      p <- noise
   else
      c = create_cluster(p)
      For each q in Nε(p)
         c.add(q)
         If Nε(q) >= MinPts
            cluster_merge(c, getCluster(q))
```

De facto, este tipo de *clustering* permite formas arbitrárias pois a *membership* de um exemplo a um *cluster* depende de existir um ponto *core* desse *cluster* a uma distância mínima ε do exemplo. Caso não satisfaça a afirmação anterior para nenhum *cluster* este exemplo será considerado *noise*, o que é uma propriedade interessante pois traz alguma flexibilidade ao algoritmo não forçando todos os pontos a pertencer a um algoritmo, o que potencialmente prejudicaria a coesão dos *clusters*. Em relação à parametrização é feita através de heurísticas simples, ao contrário de outros algoritmos de *clustering* em que a priori não são conhecidos os valores apropriados.

Seleção do parâmetro ε

Os autores deste algoritmo aconselham atribuir o valor 4 ao MinPts para 2 dimensões, tendo-se utilizado a mesma sugestão para 3 dimensões. A ideia geral desta heurística é encontrar o ε do *cluster* mais "fino". Para este efeito, ordenando os pontos em ordem decrescente pela distância ao ponto mais distante dos 4 e representando graficamente, ter-se-á uma ideia quanto à distribuição da densidade dos exemplos. Assim, se se encontrar o ponto (representado pelo "cotovelo" da função) em que estas 4-dist deixam de estar ligadas a exemplos considerados *noise* ter-se-á o desejado ε correspondente ao 4-dist do *cluster* mais "fino". É importante, no entanto, experimentar com valores à volta do ε selecionado, pois esta seleção do "cotovelo" da função é feita "a olho".

André Lopes nº 45617
Nelson Coquenim nº 45694

Gaussian Mixture Models (GMM)

O algoritmo GMM é um modelo **probabilístico**, **completo** e **particional** de *clustering* em que cada exemplo terá associada a probabilidade de pertencer a cada *cluster*. O GMM é constituído por uma mistura de distribuições, neste caso de distribuições gaussianas/normais. Com uma só variável, cada uma das distribuições é caracterizada pela média (μ) e o desvio padrão (σ), com mais do que uma variável cada distribuição irá ser caracterizada pela média (μ) e a matriz de covariância (Σ):

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Numa mistura de distribuições gaussianas, ir-se-á atribuir a cada das distribuições um peso (π_k) sendo que a soma de todos os pesos é igual a um para que as probabilidades estejam normalizadas. Sendo assim, pode-se definir a mistura de k distribuições gaussianas como:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k)$$

Seja Z a distribuição que retrata a probabilidade de um ponto (x) do conjunto de dados pertencer a cada distribuição gaussiana. A verdade é que esta distribuição Z não é conhecida inicialmente. Caso se tivesse esta informação, seria possível achar a máxima *likelihood* para os parâmetros das gaussianas. O problema é que este Z é desconhecido e depende tanto dos parâmetros da gaussiana (desconhecidos) como dos dados observados. É assim que surge a necessidade do algoritmo de Expectation-Maximization (EM):

1. Começa-se por atribuir valores aleatórios para os parâmetros das diversas gaussianas;
   EXPECTATION
2. Estima-se as diversas distribuições Z de cada ponto;
   MAXIMIZATION
3. Maximiza-se a *likelihood* dos parâmetros das gaussianas;
4. Atualiza-se os parâmetros das gaussianas;
5. Verificar se o algoritmo convergiu
   a. Se sim, termina
   b. Se não, repete-se a partir do 2

André Lopes nº 45617
Nelson Coquenim nº 45694

Abordagem

Pré-Processamento

O conjunto de dados original, constituído pela latitude e longitude, foi transformado para coordenadas ECEF [1] de modo a permitir o cálculo de distâncias entre os sismos. Posteriormente, não foi efectuada mais nenhuma transformação (e.g. normalização, standardização) aos dados, visto que todas as coordenadas apresentam uma escala comum. Adicionalmente, não se realizou *shuffle* do conjunto de dados visto que para este problema é desprezável a ordenação do mesmo. Contrariamente ao que era efectuado nos problemas de classificação nos quais é necessário separar os dados em conjunto de teste e treino. Ou seja, o resultado final do *clustering* não será influenciado pela ordenação dos dados.

K-Means vs DBSCAN vs GMM

O problema apresentado é baseado em dados espaciais, importa por isso analisar o significado dos parâmetros dos diferentes algoritmos. Começando pelo K-Means, o K como já foi referido anteriormente, representa o número de centróides, em que cada um representa um *cluster*. Neste problema, dado que se pretende agrupar os sismos dado a sua localização, a métrica utilizada para calcular as distâncias entre os vários sismos é a distância Euclidiana a três dimensões. Por outro lado, tendo a falha associada ao sismo para cada exemplo, isto permite verificar quantos *clusters* deverão ser criados, apenas com o cálculo de quantas falhas diferentes estão associadas a sismos. Neste *dataset* este valor é 27, portanto importa observar a performance à volta deste valor. Em relação ao DBSCAN, o parâmetro ε representa o raio no qual se considera que outro sismo faz parte da vizinhança de um dado sismo. Finalmente, o parâmetro do GMM que representa o número de componentes irá variar entre o número de falhas, de modo homólogo ao K-means.

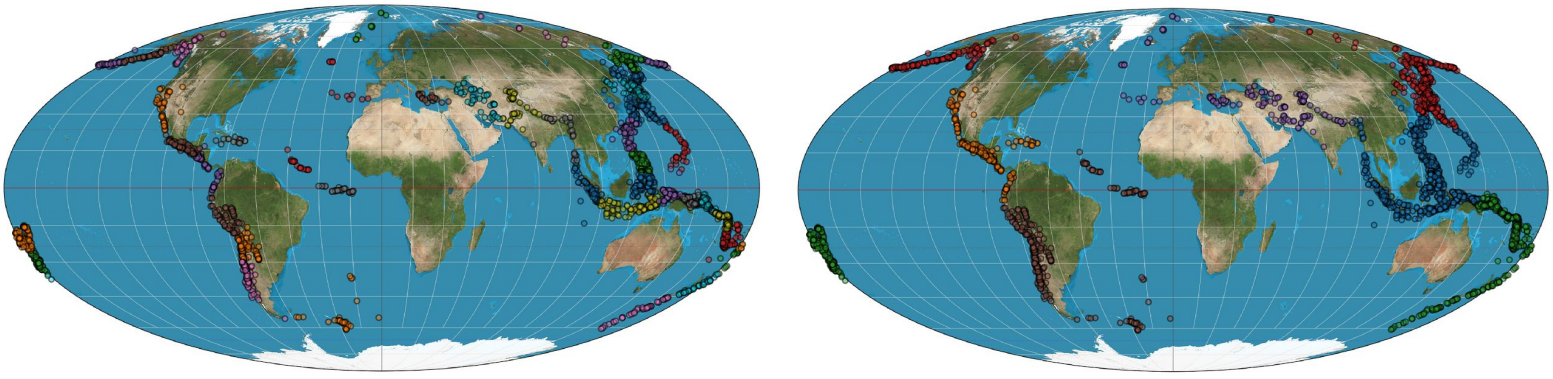
Em relação às vantagens e desvantagens de cada algoritmo para o problema apresentado:

- O K-Means, ao contrário dos outros algoritmos, tem algumas dificuldades a detectar *clusters* com formas não esféricas, e neste problema como as falhas possuem formas lineares curvas, este algoritmo não irá detectar correctamente estas. Adicionalmente, como o K-Means é completo, sismos cujas falhas foram consideradas irrelevantes (menos de 30 sismos registados) irão influenciar negativamente o *clustering*, potencialmente "arrastando" os protótipos na direcção do *noise*. Por último, o K-Means necessita do número de *clusters* a priori, o que neste contexto deveria ser à volta do número de falhas. Ou seja, caso o k seja inferior ao número de falhas poderá, como ilustra a Fig.2, levar duas falhas a pertencer ao mesmo *cluster*. Caso seja superior, pode acontecer uma falha ser representado por mais que um *cluster*;
- Analisando a adequação do DBSCAN para este problema, contrariamente ao K-means este algoritmo conseguirá adaptar os *clusters* às formas lineares curvas das falhas propagando-se assim a *membership* desse *cluster* pela linha curva da falha, podendo existir problemas ao definir, da melhor forma, a fronteira entre dois *clusters*. Outra das vantagens relativamente aos outros dois algoritmos considerados

é que existe a noção de *noise,* sendo que assim a formação dos *clusters* não será tão influenciada por exemplos de falhas irrelevantes contrariamente ao que acontece no K-means e GMM. Outro ponto muito importante é o facto de não ser necessário conhecer o número de *clusters* a priori. Sendo o número de clusters determinado pelo ε, ou seja, um valor superior ao ε seleccionado pelo algoritmo referido anteriormente, leva a *clusters* que podem englobar sismos de falhas diferentes. Enquanto que um valor menor, pode levar a sismos da mesma falha serem agrupados em diferentes *clusters*;

- O GMM, tal como o DBSCAN, identifica *clusters* com diferentes geometrias, sendo uma vantagem para este problema em comparação com o K-Means. Adicionalmente, como se conhece previamente o número de falhas, o número ideal de componentes variará à volta desse valor, sendo as consequências de ser inferior ou superior análogas às do K-means. GMM sendo um algoritmo de *cluster* completo, partilha o mesmo problema de não identificar *outliers*, tal como o K-means. Este algoritmo é o único dos três que é probabilístico, ou seja, para cada sismo é atribuído a probabilidade de pertencer a umas das componentes, contrariamente aos outros algoritmos que atribui um sismo a uma única falha, neste problema não é muito útil porque pela natureza dos sismos cada um estará associada apenas a uma falha.

Como foi apontado apenas o DBSCAN tem a noção de *noise*. Contudo, neste conjunto de dados, para além da localização dos sismos (exemplos), também se possui a informação das falhas (*labels*) a que cada sismo está associado. Assim, de forma a minimizar a influência dos sismos associados a falhas consideradas irrelevantes (< 30 sismos registados) no K-Means e GMM experimentou-se ignorar estes sismos.



k = 40 k = 6

Figura 2 - Comparação dos resultados do K-Means com k = 40 e k = 6

André Lopes nº 45617
Nelson Coquenim nº 45694

Validação

De forma a validar a solução é necessário a utilização de métodos de avaliação. Nesta secção descreve-se estes diferentes *scores* e a sua relevância em possíveis aplicações deste *clustering* de eventos sísmicos. Entre estes serão utilizados índices internos e externos, os primeiros representam medidas que avaliam a estrutura dos *clusters* obtidos. Em relação aos segundos, estes permitem comparar a estrutura conhecida, através de dados externos (falha associada ao sismo), aos *clusters* obtidos.

Índices internos:

Silhouette Score:

Este método avalia a coesão e separação dos *clusters*, para cada ponto i é calculado a distância média entre o ponto i e todos os outros pontos no mesmo *cluster* (uma medida de coesão, a(i)) é também calculado a distância média do ponto i a todos os pontos pertencentes ao *cluster* mais próximo (esta é uma medida de separação do *clusters* mais próximo, b(i)).

O silhouette score para o ponto i é dado pela seguinte fracção:

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

Pode-se obter o silhouette score para um *cluster* fazendo a média com base em todos os pontos do *cluster*. Do mesmo modo, pode-se obter o silhouette score para o *clustering* de um problema através da média efectuada com base em cada *cluster*.

Este *score* varia entre entre -1 e 1, sendo que 1 é o melhor valor e -1 o pior. Valores positivos indicam que um exemplo está mais longe dos *clusters* vizinhos do que o *cluster* que lhe foi atribuído, enquanto que valores próximos de 0 indicam que o exemplo aproxima-se da fronteira entre *clusters* vizinhos, valores negativos indicam que o exemplo pode ter sido atribuído ao *cluster* errado.

Índices externos:

Rand Index:

Após o *clustering* dos sismos pode-se contabilizar quatro grupos através de todos os pares de sismos, os verdadeiros positivos (**VP**) - pares de sismos da mesma falha que ficaram colocados no mesmo *cluster*, os verdadeiros negativos (**VN**) - pares de sismos provenientes de falhas diferentes e que foram colocados em *clusters* diferentes, os falsos positivos (**FP**) - pares de sismos provenientes de falhas diferentes e que foram colocados no mesmo *cluster*, os falsos negativos (**FN**) - pares de simos provenientes na mesma falha e que foram colocadas em falhas diferentes. Com estes quatro grupos podemos calcular um análogo à exactidão nos classificadores:

$$Rand\ Index = \frac{VP + VN}{VP + VN + FP + FN}$$

Varia entre 0 e 1, sendo 1 o melhor valor, ou seja, todos os pontos pertencem ao *cluster* que lhes corresponde segundo os dados externos.

André Lopes nº 45617
Nelson Coquenim nº 45694

De salientar que este *score* não tem em conta a hipótese do *clustering* ter sido feito ao acaso, pelo que se pode inferir, de forma errónea, que o método de *clustering* em questão demonstra elevado grau de exactidão.

Precision:
Dos pares de sismos que foram atribuídos o mesmo *cluster* quantos pertenciam à mesma falha. Varia entre 0 e +1, sendo +1 o valor maior.

$$\frac{VP}{VP + FP}$$

Recall:
Dos pares de sismos que pertencem à mesma falha quantos é que foram atribuídos o mesmo *cluster*. Varia entre 0 e +1, sendo +1 o valor maior.

$$\frac{VP}{VP + FN}$$

Medida F1: Média harmónica entre a Precision e o Recall. Varia entre 0 e +1, sendo +1 o valor maior.

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

Adjusted Rand Index (ARI):
É uma versão do Rand Index que tem em conta a possibilidade do *cluster* ter sido feito ao acaso. Varia entre -1 e +1, sendo +1 quando o *clustering* é "perfeito" tendo em conta os dados externos. Por outro lado, 0 corresponde a um *clustering* aleatório, e menor que 0, a um *clustering* pior que a aleatoriedade.

$$\frac{Rand\ Index - Expected\ Value}{Max\ Index - Expected\ Value}$$

No processo de *clustering* dos sismos é importante ter em conta a aplicação que será dada a este. Por este motivo é importante ir de encontro às necessidades da aplicação, que neste caso é agrupar sismos de modo a determinar as falhas (tendo uma base teórica inicial). De forma a seleccionar os *scores* que melhor se ajustam ao problema, executou-se e verificou-se como variam estes com a variação dos parâmetros de cada um para cada algoritmo (c/ e sem *noise*). Neste tipo de problemas de *clustering*, estes métodos de validação parecem ajustar-se muito à natureza do problema, portanto, observar-se a tendência dos *scores*, permite verificar que alguns destes não contribuem com informação adicional (e.g. mantêm-se relativamente constantes).

Não irá ser utilizado o Silhouette score devido à distribuição dos sismos, a verdade é que os eventos sísmicos distribuem-se ao longo da falha. Isto faz com que os pontos possam estar mais próximos de outro *cluster* que do próprio (fronteira entre falhas) não sendo isto grave, mas o Silhouette score irá penalizar este facto. Ao observar-se o comportamento deste, nas diferentes execuções, verifica-se que este mantém-se relativamente constante para o K-means e GMM (Fig.4,5,8 e 9), por outro lado, para o DBSCAN este é inferior a 0, pelo simples facto do *noise* estar

a ser considerado como um cluster distribuído por todo o globo, e por isso, com uma coesão muito baixa que prejudicará o valor do Silhouette Score.

Nesta aplicação será tão importante a Precision como a Recall, visto que se quer simultaneamente aumentar o número de pares de sismos que pertencem à mesma falha e foram atribuídos ao mesmo *cluster* bem como o número de pares de sismos correctamente atribuídos a cada falha dentro de cada *cluster*. Para tal ir-se-á ter em conta o F1 score e não os anteriores, visto que este efectua a sua média harmónica. Como este demonstra ser maximizado nos mesmos pontos que o ARI, apenas se irá considerar o último.

Sobra o Rand index e o ARI, como o primeiro não contabiliza a hipótese dos *cluster* terem sido efectuados de forma aleatória, será utilizado o ARI em alternativa. Assim sendo, o objectivo será maximizar o número de pares de sismos que foram corretamente atribuídos à falha. Para tal utilizar-se-á o ARI como base para escolha dos parâmetros de cada algoritmo de *clustering*.

Ir-se-á executar os três algoritmos com e sem os sismos das falhas irrelevantes, é esperado que a inclusão destes sismos piore o desempenho, uma vez que o resultado obtido com estes sismos pode ser menos representativo do *cluster*.

De salientar que os sismos identificados pelo o DBSCAN como noise serão contabilizados na análise dos resultados, para que o resultados dos métodos de validação de todos os algoritmos contemplem os mesmos sismos.

Análise de Resultados

Nesta etapa do relatório ir-se-á proceder à análise e discussão dos resultados dos *scores* discutidos na secção anterior pelos três algoritmos de clustering.

K-Means

Através da Fig.4 pode-se observar os resultados dos scores pelo K-Means, sem os sismos de falhas irrelevantes (noise), variando o número de cluster (k).

O intervalo entre, aproximadamente, 15 e 20 é onde se verifica melhores resultados para o ARI (i.e 0.38) sendo este maximizado com k = 17 na execução sem noise.

Quando se adiciona mais do que 17 *clusters* menor é número de pares de sismos que pertencem à mesma falha que são atribuídos ao mesmo *cluster*, sendo que a recall reduz-se mas a precision aumenta porque com mais cluster, mais pares de sismos são agrupados e dentro destes, maior é número de pares de sismos que foram atribuídos correctamente a mesma falha. Mas como estes não variam do mesmo modo, o resultado da média harmónica, F1, é decrescente. O Silhouette Score não varia substancialmente, mantendo-se por volta dos 0.5, isto deve-se a coesão dentros dos clusters e à distância entre clusters não variar muito.

De salientar que com a remoção dos sismos de falhas irrelevantes (Fig.5), os scores aumentam, o ARI melhorou para 0.5. Dado que estes sismos fazem com que a distância euclidiana média entre cada ponto do cluster ao centróide seja maior do que quando estes sismos não são considerados (Fig. 3).

Figura 3 - Representação de um outlier, no topo da imagem, e o centróide A.
https://i.stack.imgur.com/vg7G1.png

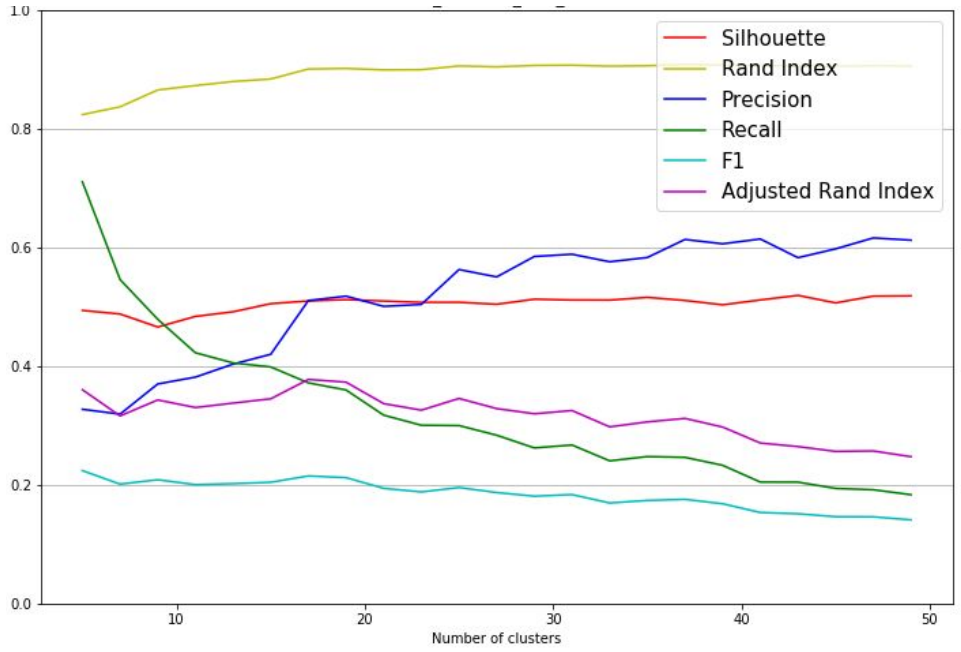


Figura 4 - Resultados dos métodos de validação do K-Means no eixo das ordenadas, **com** os sismos de falhas irrelevantes, variando o número de clusters no eixo das abscissas.

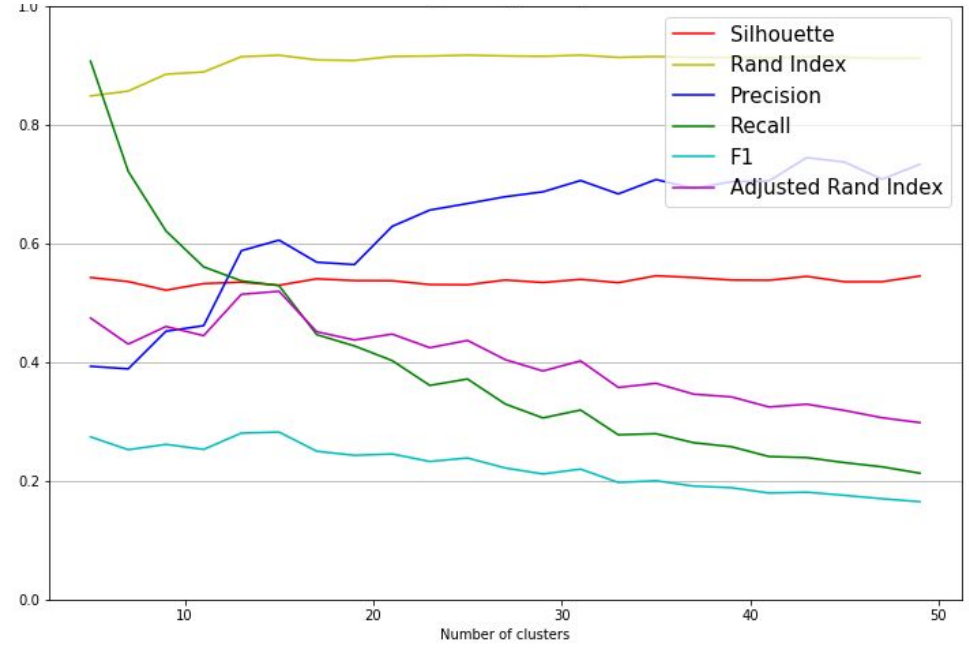


Figura 5 - Resultados dos métodos de validação do K-Means no eixo das ordenadas, **sem** os sismos de falhas irrelevantes, variando o número de clusters no eixo das abscissas.

DBSCAN

Em relação ao DBSCAN, na Fig. 7 e 8 está representado a variação dos scores com a variação do ε com e sem *noise*, respectivamente. Primeiro, seleccionou-se o intervalo de valores de ε à volta do qual será feita a análise (Fig.6). Em relação à validação, começando pelo Silhouette Score, este não parece transmitir muita informação devido à elevada oscilação, tanto com como sem *noise*. No caso da execução com *noise* deve-se ao facto de se estar a considerar o noise como um *cluster*. No que diz respeito ao Precision, Recall e F1 a análise é análoga ao K-means.

Finalmente, o ARI, o *score* que se definiu como sendo o mais informativo neste contexto, sem os sismos de falhas irrelevantes é maximizado para valores de ε = 140, ARI = 0.49 e 65 clusters enquanto que com *noise* é maximizado em ε = 160, ARI = 0.39 e 70 clusters. O que caracteriza os sismos das falhas irrelevantes é o facto de para cada falha existirem menos do que 30 sismos, ou seja, a densidade é muito baixa, o que pode levar o DBSCAN a agrupar estes sismos que não têm mais nenhum na vizinhança e acabe por separar sismos das mesmas falhas, causando uma redução no ARI.

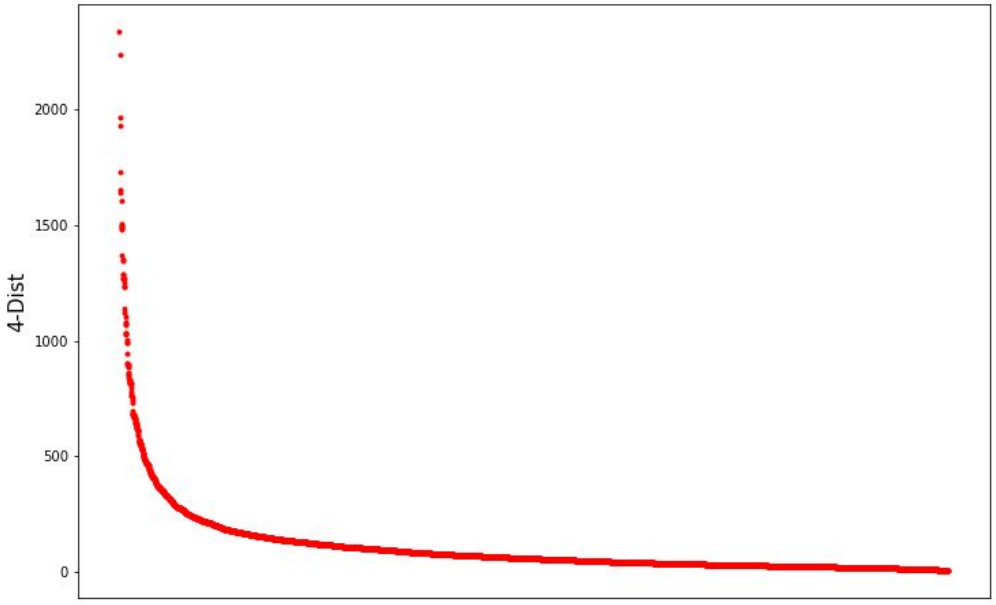


Figura 6 - Gráfico 4-Dist *sorted* com "cotovelo" entre 100 e 400.

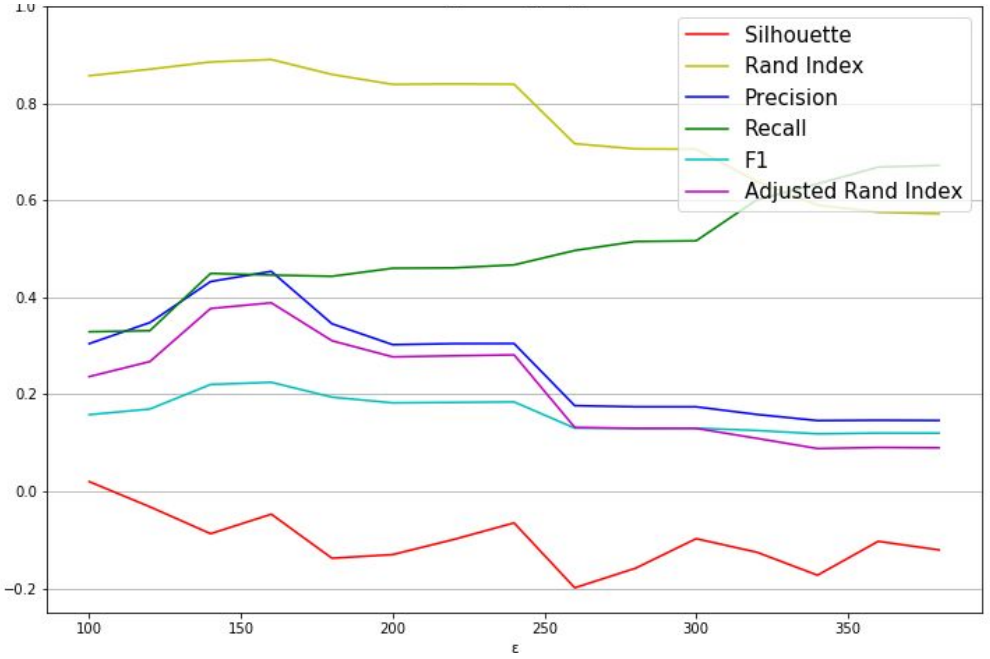


Figura 7 - Resultados dos métodos de validação do DBSCAN no eixo das ordenadas, **com** os sismos de falhas irrelevantes, variariando o valor do epsilon no eixo das abscissas.

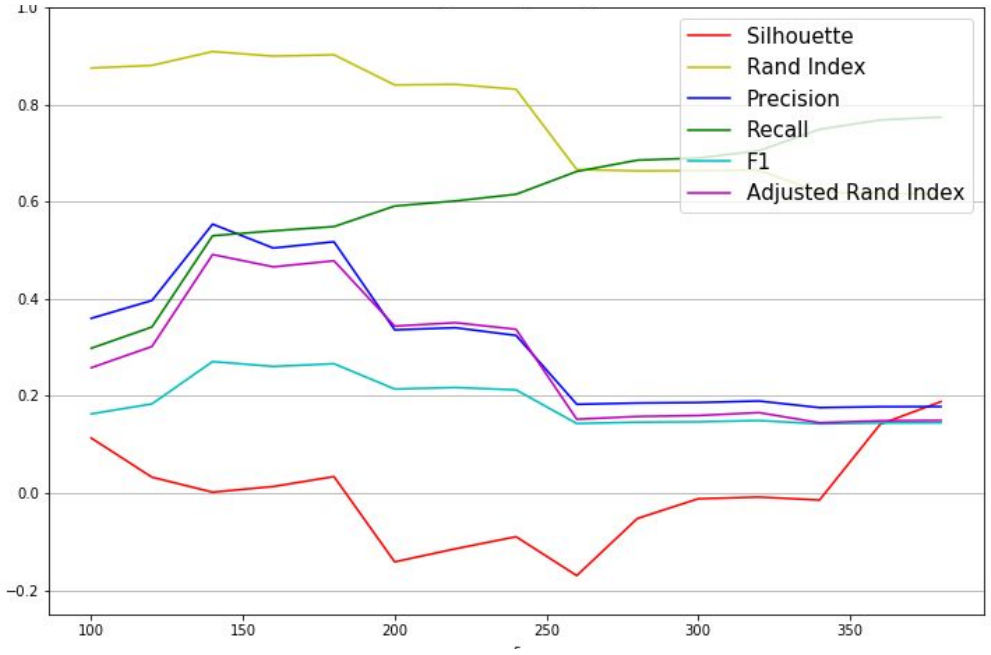


Figura 8 - Resultados dos métodos de validação do DBSCAN no eixo das ordenadas, **sem** os sismos de falhas irrelevantes, variariando o valor do epsilon no eixo das abscissas.

GMM

Finalmente, analisando o Gaussian Mixture Model, com (Fig.9) e sem (Fig.10) os sismos de falhas irrelevantes, o máximo do ARI, para em ambas as situações, atinge-se quando o número de componentes é igual a 7. De forma semelhante ao K-Means, as Fig.9 e 10 seguem a mesmas tendências. Um facto interessante que se observa quando se efectua a maximização dos algoritmos segundo o ARI, é que no K-Means para um ARI = 0.52 obteve-se 15 clusters mas com o GMM para o mesmo valor de ARI = 0.52 obteve-se apenas 7 clusters. Tendo em conta que na realidade existem 27 falhas pode-se inferir que o K-Means acabou por oferecer um resultado mais próximo do resultado verdadeiro caso o objectivo da nossa aplicação fosse identificar as falhas consoante os sismos. Esta diferença pode ser devido ao GMM ser baseado em distribuições normais e a distribuição dos sismos não o ser.

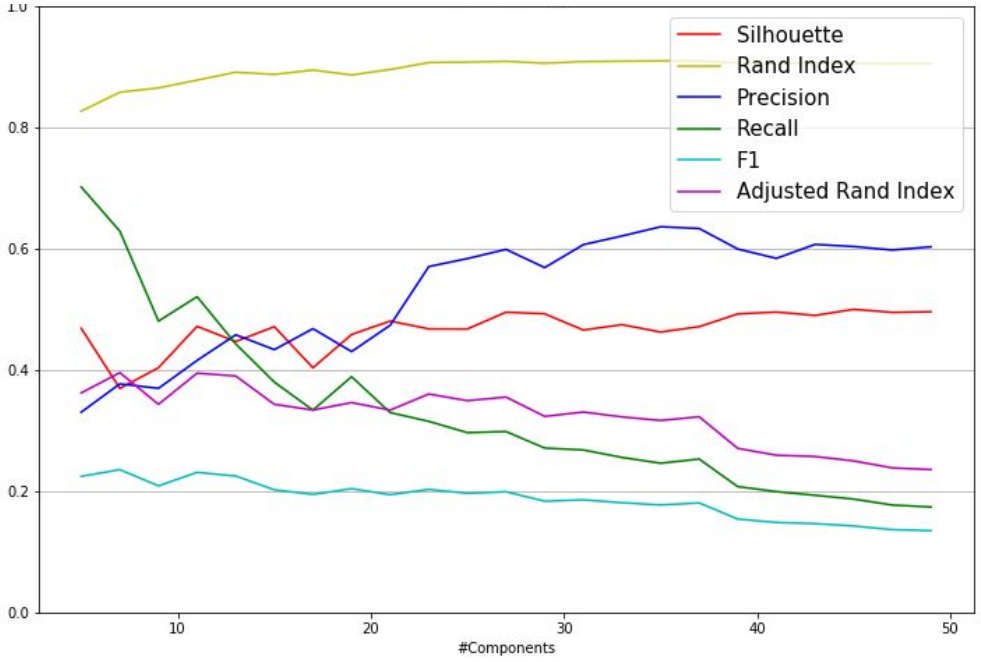


Figura 9 - Resultados dos métodos de validação do GMM no eixo das ordenadas, **com** os sismos de falhas irrelevantes, variar o número de componentes no eixo das abscissas.

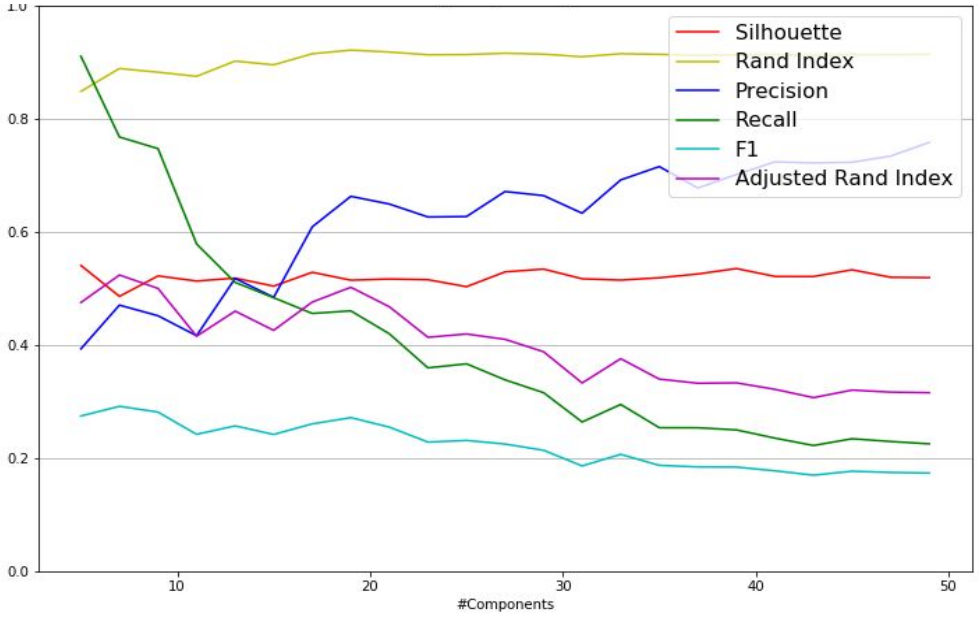


Figura 10 - Resultados dos métodos de validação do GMM no eixo das ordenadas, **sem** os sismos de falhas irrelevantes, variar o número de componentes no eixo das abscissas.

Em anexo, apresenta-se a ilustração do melhor *clustering* de sismos para cada um dos 3 algoritmos. Pode-se observar, comparando com a distribuição e usando os dados externos, que no DBSCAN o erro é feito por excesso, enquanto que nos restantes acontece o oposto. Ou seja, no problema existem 27 falhas relevantes sendo que ao maximizar o ARI obteve-se 7, 15 e 65 clusters, respectivamente para o GMM, K-Means e DBSCAN.

Conclusão

No final do presente relatório importa retirar conclusões. Em problemas de classificação, a validação é menos aberta a interpretações e mais independente da aplicação. Enquanto que em aprendizagem não supervisionada, mais especificamente em *clustering,* a escolha dos algoritmos, dos seus parâmetros e do *score* a utilizar é mais dependente do objectivo da aplicação. Sendo que a interpretação dos resultados é uma tarefa difícil, uma vez que é necessário compreender a importância e a contribuição de cada um dos diversos métodos de validação face ao problema em questão.

Neste trabalho, o propósito do *clustering*, escolhido pelo grupo, era identificar as falhas a partir da actividade sísmica. Decidiu-se parametrizar os algoritmos de *clustering* segundo o Adjusted Rand Index, o que não se revelou eficaz pois o número de clusters obtidos ficou aquém do que os dados externos determinavam, sendo que os diferentes algoritmos tiveram performance equivalente (ARI $\approx$ 0.5). Assim, uma de três situações ocorreu: O *score* usado para a validação não foi o adequado; os algoritmos não se adaptam bem ao problema ou a conjunção das duas anteriores. Não se tendo obtido uma conclusão definitiva sobre qual dos cenários se verificou.

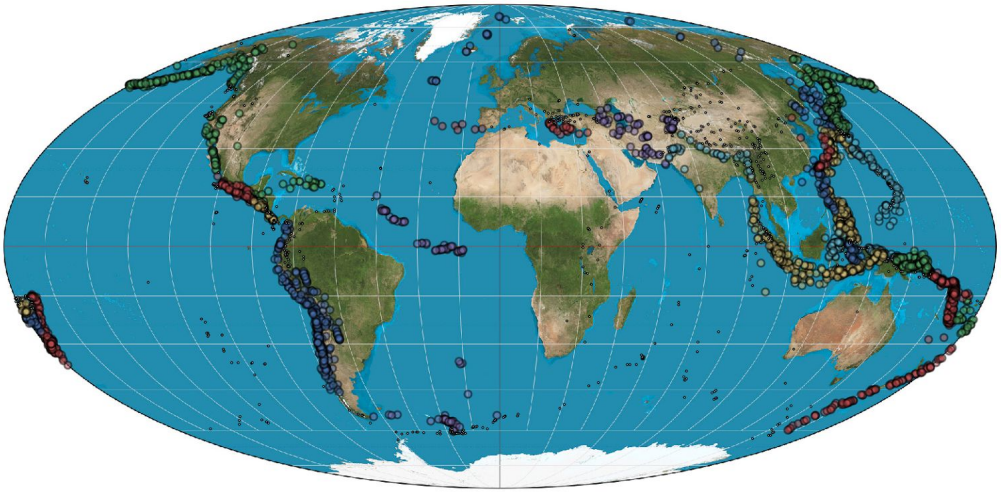
André Lopes nº 45617
Nelson Coquenim nº 45694

Bibliografia

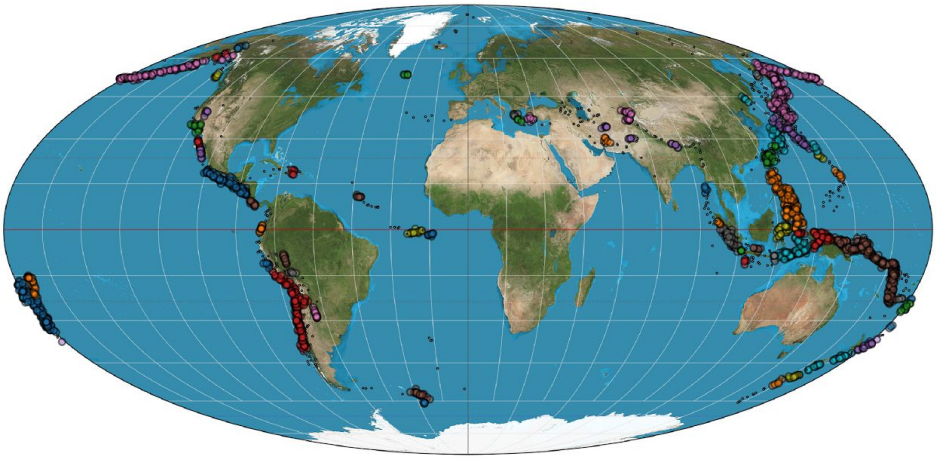
O grupo agradece as notas das teóricas disponibilizadas pelo prof. Ludwig Krippahl bem como a informação disponibilizada online nos seguintes sites:
[1] - https://en.wikipedia.org/wiki/ECEF
[2] - Martin Ester , Hans-Peter Kriegel , Jörg Sander , Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise (1996).

ANEXOS

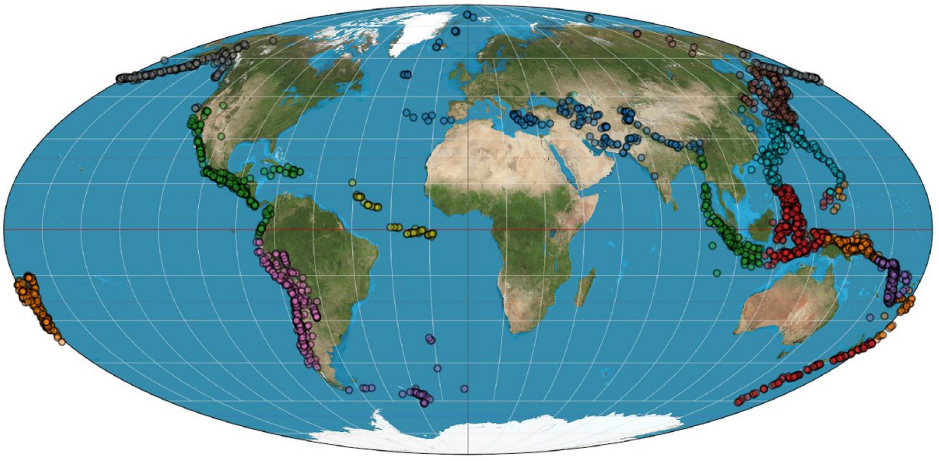
Sismos atribuídos a cada falha consoante as *labels*



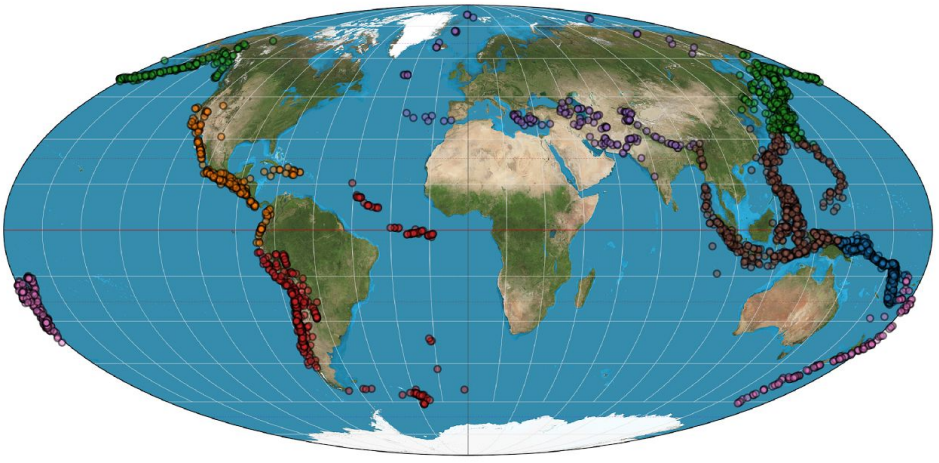
DBSCAN (eps = 140, sem noise, ARI = 0.49)



K-means (k = 15, sem noise, ARI = 0.52)



GMM (n = 7, sem noise, ARI = 0.52)



André Lopes nº 45617
Nelson Coquenim nº 45694