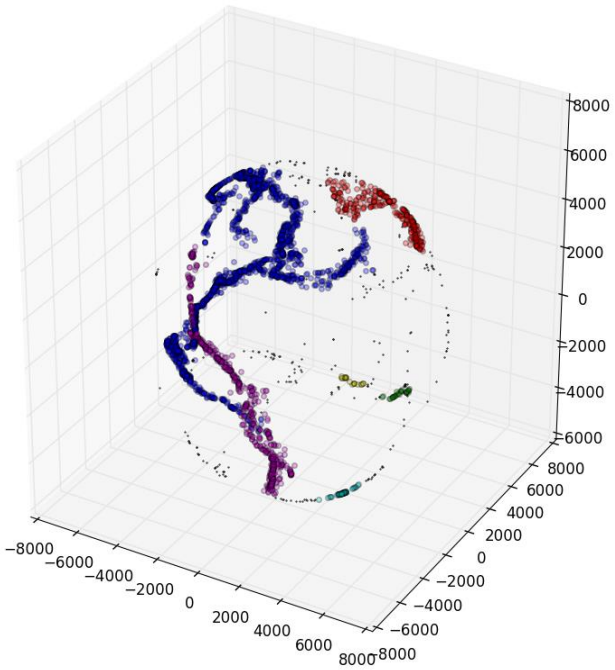
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Aprendizagem Automática

Trabalho Prático 2
2016



Clustering de sismos
Usando os algoritmos
K-Means
DBSCAN

Professores:
Ludwig Krippahl
Susana Nascimento

Alunos:
Bruno Ferreira - 44763
David Gago - 41710
Emídio Correia - 46815

Índice

Introdução

O objectivo deste trabalho é agrupar pontos semelhantes "clusters" de sismos com magnitude de pelo menos 6,5 nos últimos 100 anos.
Os dados de input estão num ficheiro, "century6_5.csv". Vindo de USGS catalog.
Para este trabalho tivemos que parametrizar e comparar dois algoritmos de clustering, o K-Means e o DBSCAN.
Usou-se o silhouette score para estimar a qualidade dos clusters que se obteve com os diferentes parâmetros usados.

O **silhouette score** pode ter um valor de -1 (pior) a 1 (melhor), e o valor 0 indica uma sobreposição de clusters. É um teste sem supervisão, isto é, a máquina trata de encontrar o valor automaticamente sem ser preciso um humano.
O silhouette score de um ponto, é obtido pela diferença entre a distância média desse ponto para todos os pontos no cluster mais próximo, e esse ponto e todos os pontos do próprio cluster, dividindo pela maior das duas médias.
Fazendo a média do silhouette score de todos os pontos, obtemos o silhouette score do conjunto.

K-Means, é um algoritmo iterativo simples, rápido e escalável para um número elevado de pontos. O K-Means recebe como parâmetros de entrada o número de clusters (k) e uma seed para a sequência aleatória, a partir da qual vão ser calculados os centróides iniciais.
Os centróides são pontos virtuais que definem a média dos clusters respetivos. Para expandir um cluster encontra-se o próximo ponto mais perto do centróide e calcula-se a nova média desse cluster. Repete-se este processo até todos os pontos pertencerem a algum cluster.

DBSCAN, é um algoritmo de clustering que pode resolver problemas difíceis onde existe uma grande densidade de pontos.
O algoritmo explora o espaço em todas as direções e marca as fronteiras quando a densidade diminui, representado por um valor epsilon (ε).
As áreas onde a densidade é insuficiente, são simplesmente consideradas vazias, e todos estes pontos são considerados ruído (noise ou outliers).

K-Means vs DBSCAN

O DBSCAN é mais complexo e requer mais tempo de computação que o K-Means.
Este algoritmo exclui os pontos de ruído (noise), isto é, não os classifica a todos, ao contrário do K-Means.
Devido aos dados estarem tão dispersos, espera-se que o clustering com o algoritmo DBSCAN obtenha melhores resultados que o do algoritmo K-Means.

Implementação

K-Means

Tem 2 parâmetros de entrada, o K é o número de clusters que vamos dividir os dados, e o segundo parâmetro, *random_state* é a *seed* do valor aleatório para os centróides iniciais.

No nosso trabalho usamos os valores de K a variar entre 2 a 40, para obter o K com melhor *silhouette score*.

DBSCAN

A implementação do DBSCAN requer um valor mínimo de vizinhos de um ponto, chamado de K.

Para obtermos o valor ideal de K, variamos o K entre 1 e 30, e para cada um encontramos o melhor valor de ε, comparando depois o resultado do *silhouette score* com o melhor obtido até ao momento e, caso este seja superior, guardando os outros parâmetros associados.

O DBSCAN recebe um outro parâmetro, um valor *epsilon* ε, que é a ordenada da função das distâncias ordenadas com base no vizinho mais afastado, entre esses K vizinhos. Ou seja, é a distância a partir da qual um ponto deixa de pertencer ao *cluster* e passa a ser considerado *noise*.

Para determinar o ε, deve-se encontrar o *knee* ou *elbow* da função das distâncias ordenadas. Para isso, deriva-se as distâncias ordenadas, ordenando posteriormente o vetor de derivadas crescentemente e escolhendo o primeiro valor que ultrapassa um determinado *threshold*. Encontra-se o valor correspondente no vetor de distâncias e será esse o valor de ε escolhido. Implementou-se também um método alternativo de obtenção do ε em que se procurava o primeiro valor acima do *threshold* no vetor de derivadas desordenado, mas decidiu-se apenas usar o primeiro método pois este oferecia mais garantias de sucesso.

Este ponto é importante porque é aqui que as distâncias entre os pontos têm maior expressão, representando então o meio termo entre a muita densidade e a dispersão dos pontos, para se formar clusters.

Experimentação

K-Means

Testámos o K-means com K (número de clusters) de 2 a 40. E os resultados obtidos podem ser observados no gráfico 1.
Verificou-se que quando o número de clusters é 24 obtém-se o melhor silhouette score, e que no nosso caso foi 0.5312.

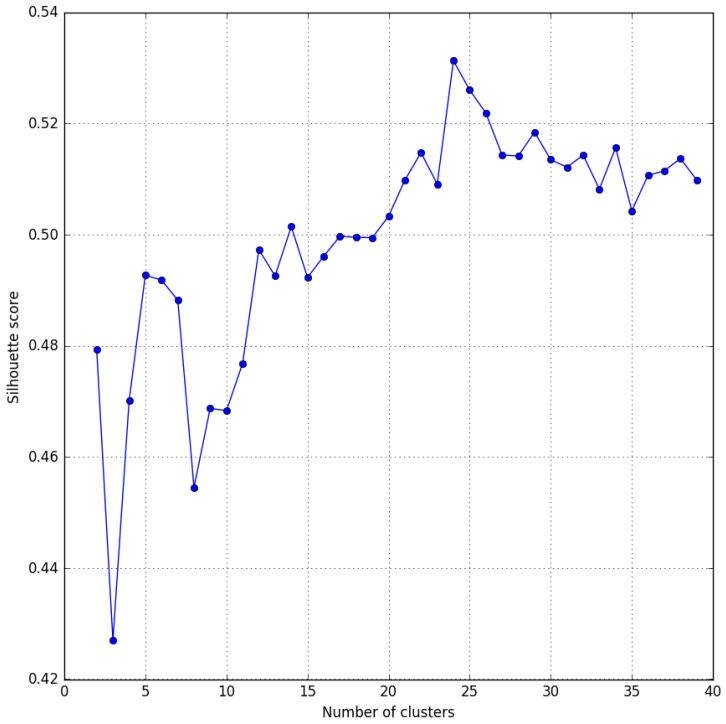


Gráfico 1 - Silhouette score para diferentes números de clusters com K-Means

O resultado da separação dos dados em 24 clusters pode ser observado na figura 1.

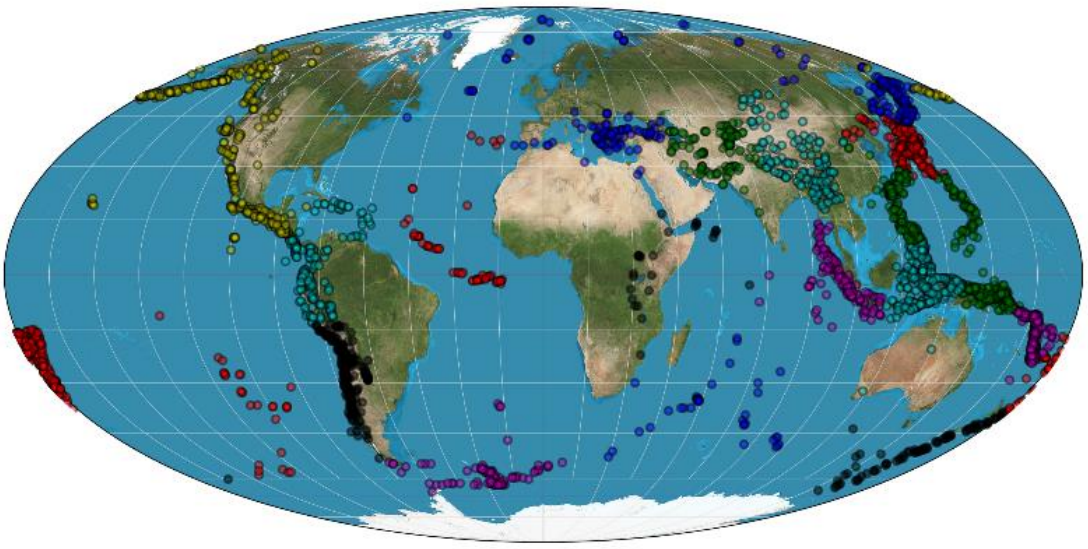


Figura 1 - Resultado de 24 clusters de eventos sísmicos usando o algoritmo K-Means

DBSCAN

Para cada k (números de vizinhos) calculou-se o ε como descrito na secção de implementação.

Para verificar se a nossa implementação estaria a comportar-se como o desejado, fez-se o plot do vetor de distâncias ordenado, e do ponto escolhido, como podemos observar no gráfico 2. Verificou-se que o método escolhido para obtenção do ε foi adequado e estava de encontro com a informação contida no artigo fornecido pelos docentes, pois em grande parte das experiências feitas com diferentes valores de k o valor de ε estava a meio do elbow.

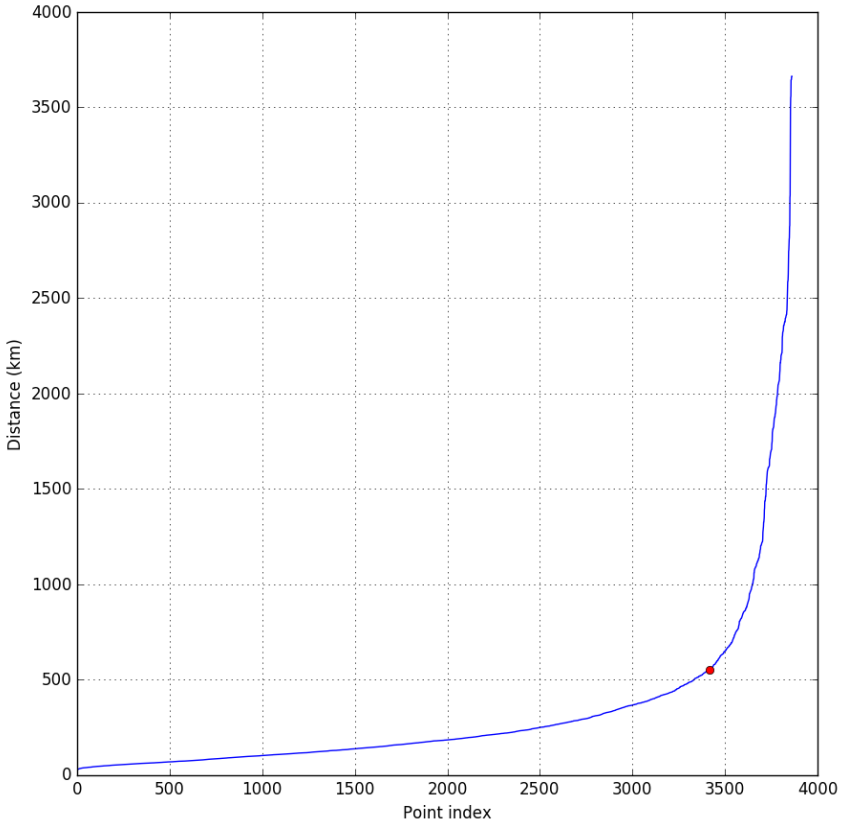


Gráfico 2 - Elbow

Para cada k (números de vizinhos) e subsequente ε,calculou-se o silhouette score para os clusters gerados pelo algoritmo DBSCAN. O melhor valor foi obtido quando se parametrizou o algoritmo com um número de vizinhos k = 15 e com um epsilon ε= 560.203 km, resultando num valor do silhouette score de 0.3069.

Os valores do silhouette score para cada k podem ser observados no gráfico 3.

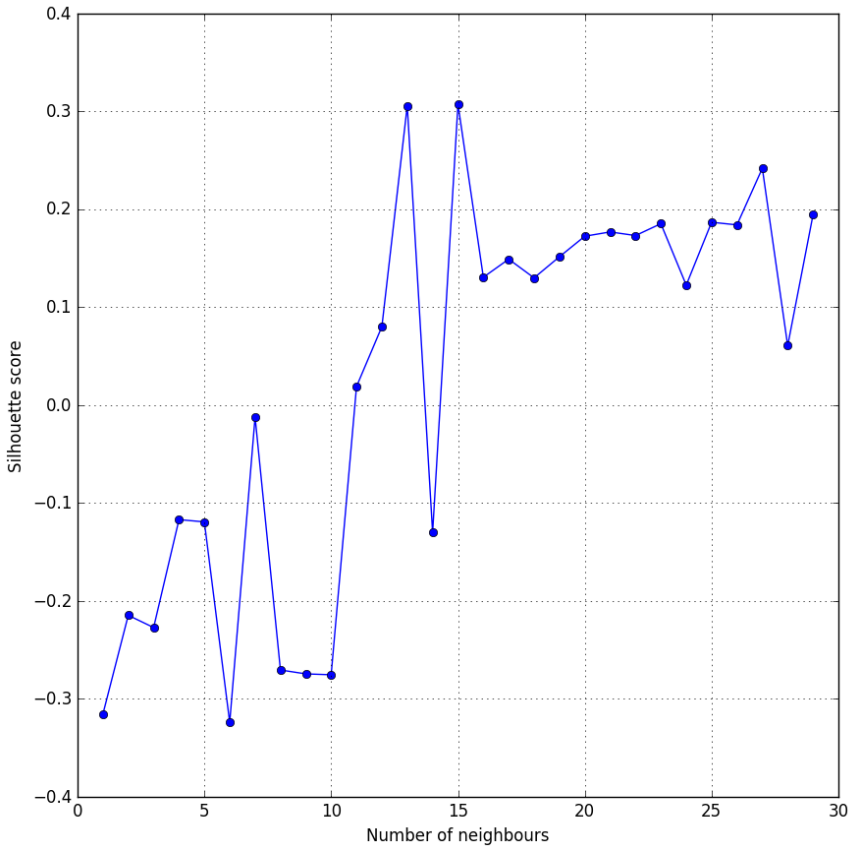


Gráfico 3 - Silhouette score para diferentes número de vizinhos com DBSCAN

A separação dos dados com o algoritmo DBSCAN parametrizado com k = 15 e ε= 560.203 km pode ser observada na figura 2.

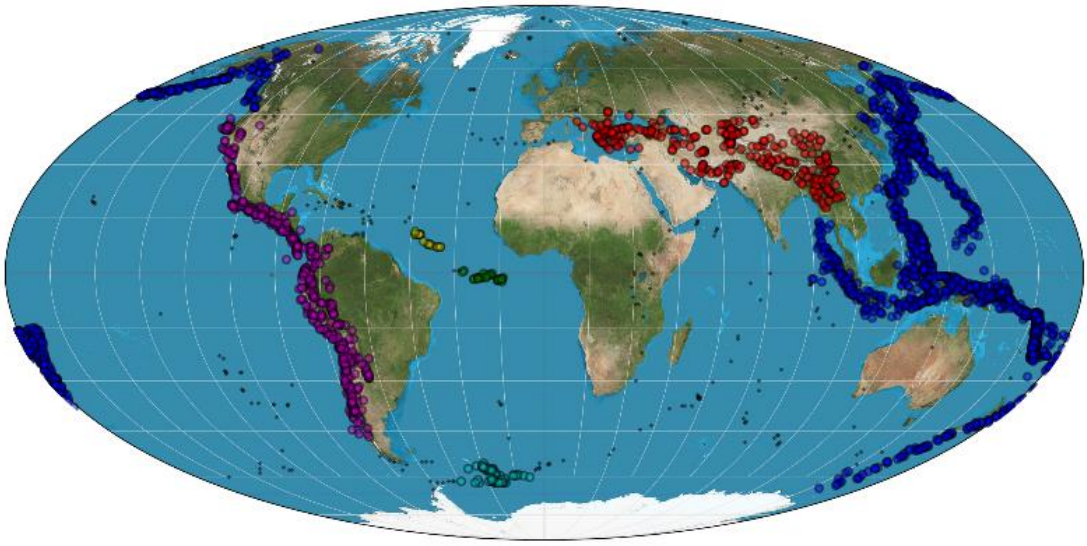


Figura 2 - Resultado de clusters dos eventos sísmicos usando o algoritmo DBSCAN para k=15

Análise de Resultados

Se contarmos apenas com os valores obtidos fazendo o silhouette score para cada algoritmo podemos concluir que o K-Means é melhor a separar os dados de sismos em clusters. No entanto o silhouette score não é muito bom a avaliar clusters que tenham formas arbitrárias e como podemos facilmente observar na figura 1 ou na figura 2, estes dados não estão concentrados em alguns pontos.

Ao analisarmos visualmente o cluster criado pelo K-Means na figura 1 observamos que há vários clusters que deveriam fazer parte do mesmo cluster, assim como clusters que incorporam pontos que não deveriam.

Observando os clusters criados pelo algoritmo DBSCAN na figura 2 ficamos com ideia que esta separação dos dados faz mais sentido que a anterior embora muitos dados sejam considerados *noise* devido ao número de vizinhos ser tão elevado. Isto deve-se a estarmos a maximizar o silhouette score que, como já vimos anteriormente, não é o melhor método de validação para este problema.

Assim decidiu-se seguir o conselho do artigo fornecido e parametrizar o número de vizinhos a 4, calculando o εpara esse número que, como podemos observar no gráfico 4, foi de 415.663 km

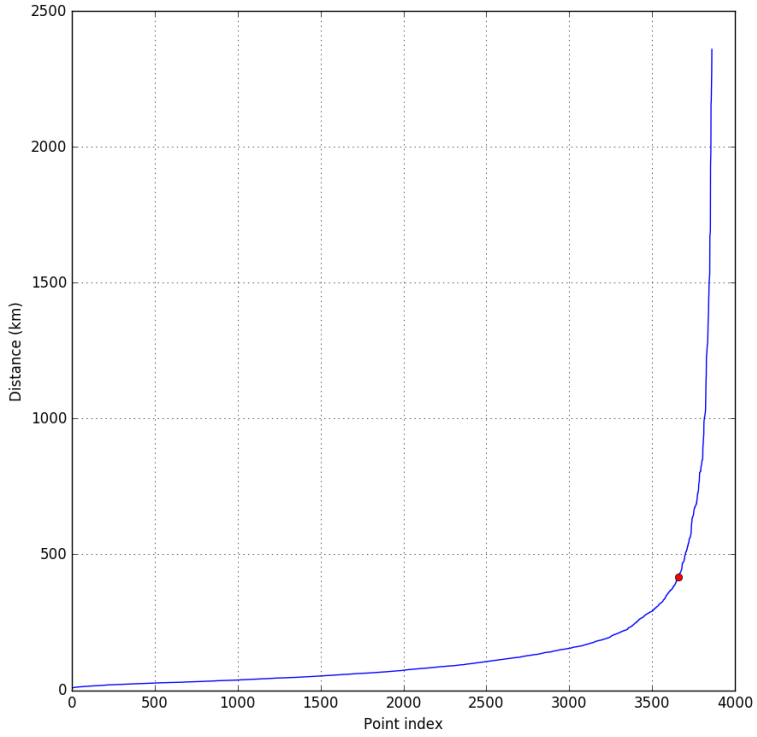


Gráfico 4 - Elbow

Posteriormente, fez-se o plot dos dados separados com estes parâmetros (figura 3).

Nesta figura podemos observar que os dados parecem estar bem separados, sendo que há menos pontos ignorados e as formas dos clusters parecem fazer sentido embora alguns dos clusters pareçam pertencer a clusters vizinhos.

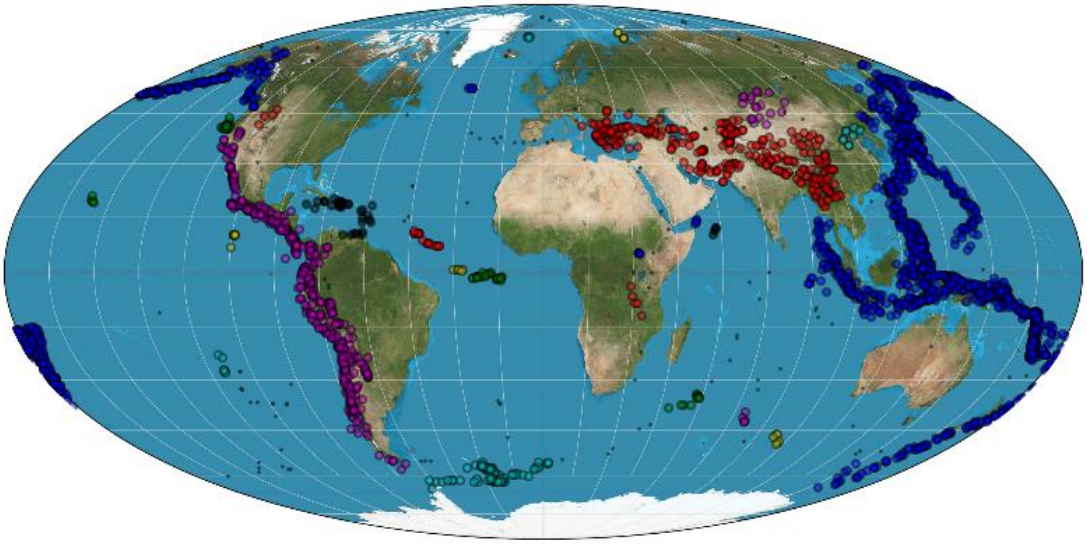


Figura 3 - Resultado de clusters dos eventos sísmicos usando o algoritmo DBSCAN para k=4

Conclusão

O artigo sobre DBSCAN "A density-based algorithm for discovering clusters in large spatial databases with noise (1996). Martin Ester , Hans-Peter Kriegel , Jörg Sander , Xiaowei Xu" fornecido pelo professor, foi útil para a recolha de mais informações, sobre o algoritmo, assim como o que deveria ser tomado em consideração dos resultados obtidos e como deveríamos proceder para efetuar o cálculo do valor de epsilon.

Verificámos que o DBSCAN é mais complexo e requer mais tempo de computação que o K-Means, mas tem a vantagem de não precisar do número de clusters como parâmetro de entrada. No entanto requer o cálculo do epsilon o que pode dar origem a erros.

Ao utilizarmos apenas a posição vinda da longitude e latitude, convertida para x, y, z dos sismos, o algoritmo que nos deu melhor resultado foi o K-Means. Mas isto deve-se ao K-Means ser um algoritmo para cálculo de clusters esféricos e o método de validação utilizado (silhouette score) não ser um bom método para avaliar clusters de forma arbitrária.

Assim sendo, através da observação dos dados agrupados com os dois algoritmos consideramos que o melhor algoritmo de clustering para estes dados é o DBSCAN.

Concluímos também que não há forma perfeita de validar clusters, e temos que escolhê-lo bem para cada problema onde quisermos efetuar clustering.