Relatório
Cientista de Dados

Nome: Guilherme Fernandes Marchezini

1.0 Introdução

A resolução apresentada nesse relatório foi feita na linguagem Python. A técnica utilizada foi o k-means, fazendo a localização e o clustering a partir da distância do ponto até o centroide.

2.0 Implementação

2.1 Tratamento dos dados

No problema os dados foram entregues em um arquivo csv, foi necessário extrair esses dados deste formato. No código foi feito uma lógica para pegar os valores de cada dimensão e transformá-lo em um ponto com 19 dimensões. Também foram armazenados o ID e a posição real da área, para que fosse possível fazer a análise dos dados.

Um problema que existia era a falta de dados para alguns sensores em determinadas ocasiões, para isso foi implementado uma solução em que para cada sensor se o valor não estiver lá, será feita uma média dos valores dos sensores pertos. Essa lógica é funcional até certo ponto, mas se existirem vários sensores sem valor, provavelmente 7 ou mais sensores, poderia ocorrer de entrar em um Loop infinito, pois os sensores tentariam fazer média, mas eles precisariam de um que também não possui valor e precisa fazer média.

2.2 K-means

O K-means utilizado foi baseado em um código que está disponibilizado na internet, o link está nas referências. Na implementação a centroide é calculada a partir dos pontos que são apresentados e a cada ponto é localizado seu centroide mais próxima e colocado seu valor no cluster daquele centroide.

Nesse código é utilizado um valor de amostra para cada dimensão, esse valor de amostra gera um centroide primária.

Foi escolhido essa implementação, pois por se tratar de um problema de área, ter como referência um centroide dentro da área e os pontos definidos a partir da distância desse centroide parece plausível. Outro fator importante a ser avaliado é que no problema após ser previsto qual área ele pertence, ele não poderia ser mudado por um cluster correto e avaliado como correto, pois o problema trata de encontrar a área naquele momento e após previsto não teria utilidade, a princípio, de mudar ele de cluster.

Nesse código foi escolhido para fazer 34 clusters, um para cada área, 19 dimensões, uma para cada sensor e a distância de optimização.

2.3 Resultados

Os resultados foram impressos em um arquivo ‘csv’ com a área original, a área que o algoritmo previu e o ID daquele conjunto de coordenadas. Também foi impresso em um arquivo chamado “dados.txt” o valor da distância de optimização e o número de acertos.

3.0 Resultados

O gráfico abaixo demonstra o número de acertos no eixo Y por distância de optimização no eixo X:

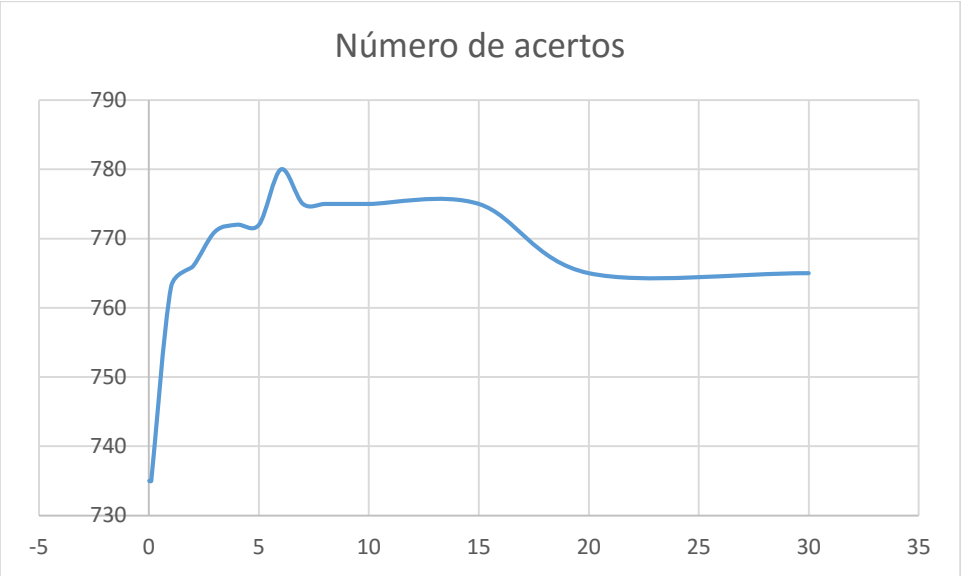

Gráfico Número de acertos X Distância

Como é possível verificar no gráfico o valor mais alto encontrado foi com a distância entre 6 e 7, com aproximadamente 780 acertos.

A taca de acerto considerando que eram 3543 valores foi de 22%, um valor muito abaixo do esperado para um sistema de aprendizado de Máquina.

Esses resultados foram abaixo do esperado por diversos motivos, um deles foi o tratamento dos dados que não possuíam valores, se for tratado de uma forma melhor esses resultados podem melhorar. Outra melhoria a ser feita é utilizar uma média de valores para criar o primeiro centroide, pois com apenas um valor o valor inicial pode ser muito fora do esperado.

Outras abordagens poderiam ter sido utilizadas para comparação de resultados e avaliar qual método se encaixa melhor e traz um melhor resultado

para esse problema. O K-means que utiliza os pontos mais próximos para determinar a qual cluster ele pertence poderia ter um resultado melhor. Métodos mais complexos existem e deveriam ser testados também, mas a falta de tempo foi um empecilho para fazer esses testes.

4.0 Considerações

Na época da confecção do algoritmo e do relatório o autor estava em semana de entrega de trabalhos e prova e possui apenas uma semana para solucionar o problema. A falta de tempo foi um empecilho muito grande para a confecção do algoritmo, pois não foi possível fazer testes sobre qual o melhor algoritmo para utilizar, tratar corretamente os dados e otimizar o código corrigindo erros que aumentam o tempo de execução e diminuem a precisão das previsões.

5.0 Referências

https://gist.github.com/iandanforth/5862470 Acessado em: 29/03/2016 às 02:13