UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DCC207 – Algoritmos 2
Prof. Renato Vimieiro

Trabalho Prático 2 – Soluções para problemas difíceis

Objetivos

Nesse trabalho serão abordados os aspectos práticos relacionados à implementação de algoritmos aproximativos. Especificamente, abordaremos o problema dos k-centros, útil na tarefa de agrupamento em aprendizado de máquinas.

Abordaremos também questões de comparação empírica de algoritmos/programas. Nesse sentido, vamos comparar as implementações dos algoritmos 2-aproximados para o problema do k-centro com o algoritmo clássico para o problema de agrupamento (K-Means). As comparações deverão avaliar tanto aspectos de demanda computacional quanto de qualidade de solução.

Tarefas

Os alunos deverão implementar os algoritmos 2-aproximados vistos em aula usando a linguagem Python 3. Deverão ser implementadas também todas as funções auxiliares envolvidas no algoritmo. Isto é, deverão ser implementadas funções para o cálculo de distâncias.

Os alunos deverão implementar a função de distância de Minkowski, a qual deverá ser avaliada nos experimentos com diferentes valores de $p \geq 1$ (obrigatoriamente deverão ser testados $p=1$ e 2, as quais equivalem, respectivamente, a distância Manhattan e Euclidiana). A implementação deverá ser feita usando funções vetoriais com a biblioteca NumPy.

Quanto aos algoritmos aproximados, deverão ser implementados a primeira versão do algoritmo 2-aproximado, aquela em que o intervalo para o raio ótimo é refinado até uma largura definida, e o outro em que os centros são escolhidos para maximizar a distância entre os centros previamente escolhidos.

Tendo implementado os algoritmos para agrupamento e a métrica de distância, o próximo passo é a avaliação empírica dos métodos. Os métodos deverão ser avaliados com conjunto de dados obtidos na UCI Machine Learning Repository (referência abaixo) e com conjuntos sintéticos que serão descritos a seguir. Para os dados reais (UCI), devem ser escolhidos 10 conjuntos de dados com, no mínimo, 700 exemplos (instâncias). Os conjuntos de dados devem ser exclusivamente numéricos para a tarefa de classificação (o atributo classe/label) deve ser ignorado durante o agrupamento

cálculo de distância; o número de valores distintos para essa variável determina o número de grupos/clusters a buscar. Para os dados sintéticos, deverão ser usados conjuntos obtidos por duas abordagens. A primeira consistirá nos exemplos apresentados em https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py. A segunda abordagem consistirá em gerar dados em duas dimensões usando a distribuição normal multivariada. Nesse caso, deverão ser gerados pontos amostrados em torno do número de centros (médias diferentes para cada centro), controlando o desvio padrão para que a sobreposição entre os grupos varie entre inexistente até altamente sobrepostos (gere uma quantidade razoável de pontos em cada centro). Deverão ser gerados 10 conjuntos aleatórios com cada método, totalizando 30 conjuntos de dados para realizar a experimentação.

Embora os algoritmos sejam 2-aproximados, a escolha dos centros é arbitrária e pode influenciar na qualidade da solução. Dessa forma, para cada conjunto de dados, deverão ser realizados 30 testes/execuções do algoritmo (note que a matriz de distância deve ser computada uma única vez, pois essa nunca se altera). A cada execução deve-se armazenar o raio da solução e, também, computar duas medidas clássicas em avaliação de agrupamentos: silhueta e índice de Rand ajustado. Essas duas últimas métricas não precisam ser implementadas, devendo ser usadas as implementações da biblioteca Scikit Learn (https://scikit-learn.org/stable/index.html). Além das métricas de qualidade, deverão ser armazenados também o tempo de processamento de cada execução.

Para o caso específico do algoritmo baseado em refinamento de intervalos, avaliaremos como a largura do intervalo em que o raio ótimo se encontra afeta a qualidade da solução. Dessa forma, o número de refinamentos deverá ser variado entre 1-25%, com pelo menos cinco valores escolhidos nessa faixa. Como sugestão, as escolhas dos valores na faixa podem ser feitas tanto de forma linear quanto exponencial. A escolha final fica a cargo dos alunos.

A título de comparação da qualidade da solução, os experimentos acima deverão ser repetidos com a implementação do algoritmo K-Means também disponível no Scikit Learn (https://scikit-learn.org/stable/modules/clustering.html#k-means). Observe que todas as medidas de desempenho também deverão ser computadas para essa implementação.

Finalmente, os resultados dos experimentos devem ser agregados em uma tabela com as respectivas médias e desvios-padrão por experimento, e por algoritmo (parâmetro de largura). Em seguida, deve ser feita uma análise dos experimentos, descrevendo as observações da comparação (e.g. qualidade de resposta, tempo de execução). Para essa etapa, deve ser feito um relatório em formato de artigo científico com: uma introdução ao problema, uma descrição dos métodos e métricas usadas, a descrição da implementação, a descrição dos experimentos, incluindo as bases de dados, apresentação e análise dos resultados, e uma conclusão. Esse artigo deve ter entre 6 e 12 páginas no formato da SBC (link abaixo).

O trabalho poderá ser feito em **grupos de até dois alunos**.

O que entregar?

Devem ser entregues todos os arquivos fonte usados na implementação. O artigo contendo o relatório deve ser entregue em formato pdf. As bases de dados usadas não devem ser entregues, mas devem ser reportadas claramente no relatório, permitindo que sejam recuperadas posteriormente (as sintéticas podem ser armazenadas no repositório). Todos os arquivos (inclusive o relatório) devem estar disponíveis em um repositório a ser tornado aberto pelo grupo 2 dias após a data de entrega.

Política de Plágio

Os alunos podem, e devem, discutir soluções sempre que necessário. Dito isso, há uma diferença bem grande entre implementação de soluções similares e cópia integral de ideias. Trabalhos copiados na íntegra ou em partes de outros alunos e/ou da internet serão prontamente anulados. Caso haja dois trabalhos copiados por alunos/grupos diferentes, ambos serão anulados.

Datas

Entrega final Teams: 15/08/2024 às 23h59

Política de atraso

Haverá tolerância de 30min na entrega dos trabalhos. Submissões feitas depois do intervalo de tolerância serão penalizados, incluindo mudanças no repositório.

- Atraso de 1 dia: 30%
- Atraso de 2 dias: 50%
- Atraso de 3+ dias: não aceito

Serão considerados atrasos de 1 dia aqueles feitos após as 0h30 do dia seguinte à entrega (sexta-feira). A partir daí serão contados o número de dias passados da data de entrega.

Referências

- https://en.wikipedia.org/wiki/Minkowski_distance
- https://archive.ics.uci.edu/ml/index.php

- https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros
- https://scikit-learn.org/stable/modules/clustering.html#k-means
- https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py