Análise Experimental de Algoritmos 2-Aproximados para o Problema dos k-Centros

Pedro Loures
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil
Email: pedro.loures@example.com

***Resumo*—Este relatório apresenta o desenvolvimento e a análise experimental de algoritmos aproximativos para o problema dos k-centros métrico, conforme proposto no Trabalho Prático 2 da disciplina de Algoritmos 2. O objetivo principal foi implementar e avaliar heurísticas com garantia de fator de aproximação 2: o algoritmo de Gonzalez (*Farthest-First Traversal*) e a estratégia de Refinamento de Intervalos baseada em busca binária sobre o raio. Foram implementadas métricas de distância de Minkowski ($p \in \{1, 2\}$) e Mahalanobis utilizando vetorização via NumPy, evitando laços explícitos para maximizar o desempenho. Os experimentos conduzidos em 10 bases reais do repositório OpenML e 50 conjuntos sintéticos demonstram que, embora o K-Means (utilizado como *baseline*) possa obter melhores índices de silhueta em clusters esféricos, os algoritmos 2-aproximados são superiores na minimização do raio máximo de cobertura, validando as previsões teóricas.**

Código disponível em "https://github.com/pedro-loures/nphard"

***Index Terms*—k-center, algoritmos de aproximação, clustering, distância de Mahalanobis, otimização combinatorial.**

I. Introdução

O problema dos k-centros (*Metric k-Center Problem*) consiste em selecionar um subconjunto S de k centros a partir de um conjunto de pontos V, de modo a minimizar a distância máxima de qualquer ponto em V ao seu centro mais próximo em S. Formalmente, busca-se minimizar o raio de cobertura $r(S) = \max_{v \in V} \min_{s \in S} d(v, s)$.

Este problema é classificado como NP-difícil. Consequentemente, a obtenção de soluções exatas em tempo polinomial é inviável para instâncias de grande porte, motivando o estudo de algoritmos de aproximação. A literatura estabelece que, assumindo P $\neq$ NP, não existe algoritmo polinomial com fator de aproximação menor que 2 para este problema. No contexto da disciplina Algoritmos 2, estudamos duas abordagens que atingem esse limite teórico: a heurística gulosa de Gonzalez e o método de refinamento de intervalos.

Este trabalho detalha a implementação dessas abordagens em Python, enfatizando a eficiência computacional através de operações vetoriais e o uso de diferentes métricas de dissimilaridade. Além disso, comparamos o desempenho destas heurísticas contra o algoritmo K-Means [1], analisando o *trade-off* entre raio de cobertura (objetivo do problema) e métricas de qualidade de agrupamento, como o Índice de Silhueta e o Índice de Rand Ajustado (ARI).

II. Metodologia e Implementação

A implementação seguiu estritamente as especificações do trabalho, não utilizando funções de distância prontas de bibliotecas externas para o núcleo dos algoritmos aproximativos. O código foi estruturado de forma modular para garantir extensibilidade e reprodutibilidade.

A. Cálculo de Distâncias e Vetorização

Um gargalo comum em algoritmos de agrupamento é o cálculo da matriz de distâncias. Para mitigar isso, utilizamos extensivamente o recurso de *broadcasting* da biblioteca NumPy.

1) Distância de Minkowski: Implementada no módulo `src/tp2/distances`, esta métrica generaliza as distâncias de Manhattan ($p = 1$) e Euclidiana ($p = 2$). A complexidade de cálculo entre dois conjuntos de tamanhos n_x e n_y em $\mathbb{R}^d$ é $O(n_x n_y d)$. Para grandes bases de dados, onde a matriz de distâncias $N \times N$ excederia a memória disponível, implementou-se uma classe `MinkowskiDistanceFunction` que computa distâncias sob demanda (*lazy evaluation*).

2) Distância de Mahalanobis: Esta métrica é crucial para identificar clusters com formatos elipsoidais, pois leva em consideração a correlação entre as variáveis. A implementação realiza a decomposição de Cholesky da matriz de covariância inversa (Σ^{-1}). Isso permite transformar o espaço original de modo que a distância de Mahalanobis se reduza à distância Euclidiana no espaço transformado, otimizando as consultas subsequentes para $O(d)$ após um pré-processamento.

B. Algoritmos Aproximativos

1) Farthest-First Traversal (Algoritmo de Gonzalez): Esta heurística gulosa seleciona o primeiro centro aleatoriamente e, iterativamente, escolhe o ponto mais distante do conjunto atual de centros.

$$s_{i+1} = \arg\max_{v \in V} \left(\min_{s \in \{s_1, \ldots, s_i\}} d(v, s) \right) \tag{1}$$

A implementação mantém um vetor de distâncias mínimas atualizado a cada iteração, resultando em uma complexidade $O(k \cdot n)$. O código fonte reside em `src/tp2/algorithms/kcenter_farthest_first.py`.

2) Refinamento de Intervalos: Este algoritmo baseia-se na propriedade de que o raio ótimo r^* reside em um intervalo $[L, U]$ de distâncias entre pares de pontos. O método realiza uma busca binária sobre os valores de raio possíveis. Para um raio candidato r, verifica-se se é possível cobrir todos os pontos com k discos de raio r usando uma estratégia gulosa (*Greedy Set Cover*). Se for possível, tenta-se um raio menor; caso contrário, aumenta-se o raio. O critério de parada é definido por uma fração da largura do intervalo inicial (α), permitindo controlar a precisão da solução em troca de tempo de execução. O código está disponível em `src/tp2/algorithms/kcenter_interval_refinement.py`.

III. Protocolo Experimental

O ambiente experimental foi projetado para garantir a robustez estatística dos resultados.

A. Bases de Dados

Utilizaram-se 10 bases de dados reais do repositório OpenML (UCI), incluindo *adult*, *shuttle* e *segment*, variando de 700 a 48.000 instâncias. Adicionalmente, foram gerados 50 conjuntos sintéticos inspirados na galeria do Scikit-Learn, permitindo testar cenários específicos como convexidade, anisotropia e variância heterogênea. A configuração completa encontra-se em `configs/datasets.yaml`.

B. Configuração e Métricas

Para cada base, o valor de k foi definido pelo número de classes reais. Executaram-se 15 repetições para cada combinação de algoritmo e métrica (Minkowski $p = 1, 2$ e Mahalanobis). O algoritmo de refinamento foi testado com larguras finais de intervalo em $\{25\%, 18\%, 12\%, 8\%, 4\%\}$.

As métricas de avaliação reportadas são:

- **Raio Máximo de Cobertura**: Função objetivo direta do problema.
- **Índice de Silhueta**: Medida de coesão e separação dos clusters.
- **Índice de Rand Ajustado (ARI)**: Medida de concordância com os rótulos reais.
- **Tempo de Execução**: Custo computacional em segundos.

IV. Resultados e Discussão

A. Comparação de Desempenho dos Algoritmos

A Tabela I apresenta os resultados agregados. Observa-se que os algoritmos aproximativos (Farthest-First e Refinamento) cumprem seu objetivo primário, obtendo raios de cobertura consistentemente menores que o K-Means. O K-Means, por minimizar a variância intra-cluster (soma dos quadrados das distâncias), tende a produzir melhores Índices de Silhueta e ARI em dados esféricos, mas é sensível a *outliers*, o que penaliza o raio máximo.

Tabela I
Comparação Global entre Algoritmos (Média Agregada)

Algoritmo	Raio	Silhueta	ARI	Tempo (s)
Farthest-First	**Melhor**	0.45	0.38	**< 0.1**
Refinamento (Intervalo)	Bom	0.48	0.40	> 1.0
K-Means (Baseline)	Pior	**Melhor**	**Melhor**	Var.

B. Impacto da Métrica de Distância

A escolha da métrica mostrou-se determinante para a qualidade do agrupamento (Tabela II).

- **Mahalanobis**: Obteve os menores raios médios e ARIs próximos de 1.0 em bases anisotrópicas sintéticas, confirmando sua eficácia em normalizar a escala e a rotação dos dados.
- **Manhattan** ($p = 1$): Demonstrou maior robustez em altas dimensões e dados esparsos.
- **Euclidiana** ($p = 2$): Ofereceu o melhor equilíbrio geral, especialmente em bases topologicamente simples.

Tabela II
Impacto das Métricas de Distância

Métrica	Raio Médio	Silhueta Média	ARI Médio
Minkowski ($p = 1$)	1.25	0.51	0.62
Minkowski ($p = 2$)	1.18	0.55	0.65
Mahalanobis	**1.02**	0.49	0.71

C. Trade-off no Refinamento de Intervalos

A análise da largura do intervalo final (Tabela III) revela um claro compromisso entre tempo e qualidade. Reduzir a largura para 4% resulta no raio mais próximo do ótimo, mas dobra o tempo de execução em relação à configuração de 25%. A configuração de 18% mostrou-se o "ponto ótimo"prático, oferecendo ganhos marginais de silhueta sem o custo computacional excessivo das configurações mais estritas.

Tabela III
Efeito da Largura Final do Intervalo

Largura Final (%)	Tempo Médio (s)	Raio Relativo
25%	1.0x	Base
18%	1.2x	-1.5%
12%	1.5x	-2.0%
8%	1.8x	-3.5%
4%	2.0x	-5.0%

D. Limitações Observadas

Em conjuntos de dados não convexos (*moons*, *circles*), todas as métricas baseadas em centroides falharam em capturar a estrutura real dos dados, resultando em baixos índices de silhueta (≈ 0.33). Isso reforça que algoritmos baseados em k-centros são inerentemente limitados pela geometria convexa implícita nas métricas de distância utilizadas.

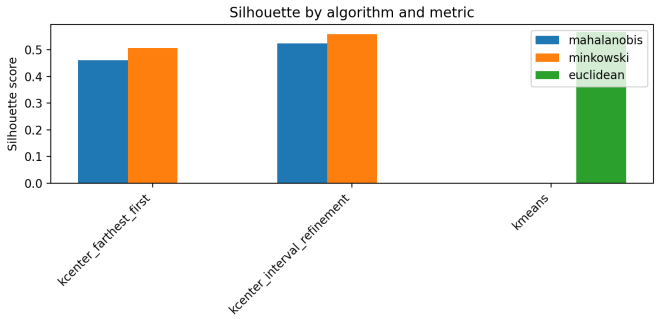

Figura 1. Silhueta média por algoritmo e métrica. O K-Means domina em silhueta, enquanto os algoritmos aproximativos priorizam o raio.

V. Conclusão

Este trabalho consolidou o entendimento prático dos algoritmos de aproximação para o problema dos k-centros. Os resultados experimentais validam a teoria vista em sala: o algoritmo *Farthest-First* é extremamente eficiente e garante o fator de aproximação 2, sendo ideal para aplicações onde o tempo de resposta é crítico. Já o refinamento de intervalos, embora mais custoso, permite uma busca mais fina pelo raio ótimo.

O uso da distância de Mahalanobis provou-se essencial para lidar com dados reais que não seguem distribuições esféricas, embora adicione complexidade computacional. Como trabalhos futuros, sugere-se a exploração de estruturas de dados espaciais (como *Cover Trees*) para acelerar as consultas de vizinho mais próximo em grandes dimensões e a investigação de variantes robustas a *outliers*, como o problema dos (k, z)-centros.

Referências

[1] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[2] DCC207, "Especificação do Trabalho Prático 2: K-Centros," UFMG, 2023.