Métodos de Seleção de Pivôs: Uma avaliação experimental

João V. S. Leite[1], Wagner R. Telles[1], Rodolfo A. Oliveira[1], Daniel de Oliveira[2], Marcos Bedo[3]

[1]Instituto do Noroeste Fluminense – Universidade Federal Fluminense – INFES/UFF

[2]Instituto de Computação – Universidade Federal Fluminense – IC/UFF

[3]Faculdade de Medicina de Ribeirão Preto – Universidade de São Paulo – FMRP/USP

{joaovitorleite, wtelles, rodolfooliveira, marcosbedo}@id.uff.br,

danielcmo@ic.uff.br

Resumo. *Consultas por similaridade em Espaços Métricos apoiam tarefas computacionais que envolvem comparações por distância, e.g., recuperação por conteúdo e classificação. Esse paradigma permite utilizar índices baseados em pivôs para reduzir o número de cálculos de distância e otimizar a execução das consultas. Os índices armazenam as distâncias dos objetos da base de dados para um conjunto de elementos selecionados (i.e., os pivôs) tal que, durante uma busca, esses valores pré-computados são combinados com a desigualdade triangular para descartar objetos da resposta. Portanto, a escolha de pivôs de "boa qualidade" é determinante para o desempenho dessas estruturas. Esse estudo investiga o impacto de oito estratégias distintas de escolha de pivôs (*`kMEDOIDS`*, *`C-HULL`*, *`PCA`*, *`M-VARIANCE`*, *`SELECTION`*, *`S-S-SELECTION`*, *`GNAT`* e *`M-SEPARATED`*) aplicadas aos índices Omni kd-Tree e VP-Tree. A avaliação experimental sugere que os métodos* `M-VARIANCE` *e* `kMEDOIDS` *encontram os melhores pivôs, enquanto as estratégias* `GNAT` *e* `C-HULL` *apresentam as piores escolhas. Os resultados também sugerem que índices diferentes são, potencialmente, melhor ajustados por diferentes métodos de escolha de pivôs.*

Abstract. *Similarity searching supports distance comparison-based tasks, e.g., content-based retrieval and classification. Under this paradigm, pivot-based indexes may speed up the search by reducing the number of distance calculations. Those indexes store the distances from data objects to a set of chosen elements (i.e., the pivots) so that the pre-computed values are combined with the triangle inequality to prune the search space during a query. Therefore, the "quality" of the pivots conditions the performance of such structures. In this study, we experimentally investigate eight distinct selection strategies (*`kMEDOIDS`*, *`C-HULL`*, *`PCA`*, *`M-VARIANCE`*, *`SELECTION`*, *`S-S-SELECTION`*, *`GNAT`*, and *`M-SEPARATED`*) coupled with indexes Omni kd-Tree and VP-Tree. The findings suggest methods* `M-VARIANCE` *e* `kMEDOIDS` *can find good pivots, whereas strategies* `GNAT` *and* `C-HULL` *may deliver low-quality pivots, on average. Results also suggest different indexes may be fine tuned by distinct pivot selection methods.*

1. Introdução

Consultas por similaridade em Espaços Métricos fornecem um modelo sólido de busca por objetos que são "parecidos" mas não idênticos. Um Espaço Métrico é um par $\langle \mathbb{O}, \delta \rangle$

onde $\mathbb{O}$ é o domínio dos dados (*e.g.*, $\mathbb{O} = \mathbb{R}^d$) e δ é uma função de distância (*e.g.*, $\delta = L_2$), sendo que fixos quaisquer objetos $o_i, o_j, o_k \in \mathbb{O}$, δ satisfaz as propriedades de *(i)* simetria: $\delta(o_i, o_j) = \delta(o_j, o_i)$, *(ii)* positividade: $\delta(o_i, o_j) > 0, o_i \neq o_j$, e *(iii)* desigualdade triangular: $\delta(o_i, o_j) \leq \delta(o_i, o_k) + \delta(o_k, o_j)$. Na prática, consultas por similaridade recuperam objetos de uma base de dados ao comparar sistematicamente suas distâncias para um elemento de referência, como no caso de buscas kNN [Hetland 2009, Chen et al. 2017].

Buscas kNN. Uma busca kNN recupera $k \in \mathbb{N}$ objetos de um conjunto de dados $\mathcal{O} \subseteq \mathbb{O}$ cujas distâncias para o elemento de referência $o_q \in \mathbb{O}$ são as menores. Formalmente, kNN $(o_q, k) = (o_1, \ldots, o_k)$; $o_1 = \{o_i \in \mathcal{O} \mid \forall\, o_j \in \mathcal{O},\ \delta(o_i, o_q) \leq \delta(o_j, o_q)\}$, $o_2 = \{o_i \in \mathcal{O} \setminus \{o_1\} \mid \forall\, o_j \in \mathcal{O} \setminus \{o_1\}, \delta(o_i, o_q) \leq \delta(o_j, o_q)\}, \ldots, o_k = \{o_i \in \mathcal{O} \setminus \{o_1, \ldots, o_{k-1}\} \mid \forall\, o_j \in \mathcal{O} \setminus \{o_1, \ldots, o_{k-1}\}, \delta(o_i, o_q) \leq \delta(o_j, o_q)\}$.

Consultas kNN são apoiadas por índices métricos que evitam cálculos de distância desnecessários [Yianilos 1993, Chávez et al. 2001, Traina Jr et al. 2007, Chen et al. 2017]. Os índices baseados em pivôs pré-computam as distâncias dos objetos armazenados para um conjunto de elementos (*e.g.*, os *pivôs*) de forma que esses valores são combinados com a desigualdade triangular para definir a regra do Limite Inferior [Hetland 2009, Mao et al. 2016, Zhu et al. 2022].

Limite Inferior. Dado um conjunto de pivôs $\mathcal{P} \subseteq \mathcal{O}$, um elemento de referência $o_q \in \mathbb{O}$ e fixado um limite de distância $\xi, \xi \geq \delta(o_k, o_q)$, todo objeto $o_i \in \mathcal{O}$ pode ser descartado de uma busca kNN (o_q, k) se $\exists\, p \in \mathcal{P}$ tal que $|\delta(p, o_q) - \delta(o_i, o_q)| > \xi$.

Mao et. al (2016) e Chen et. al (2022) apresentam uma comparação sistemática entre métodos de pivôs, indicando que os índices da família Omni e VP-Tree estão entre os mais beneficiados pela escolha de "bons" pivôs. No entanto, as comparações são realizadas com *algoritmos de busca* de diferentes viéses, o que pode afetar a forma como o limite de distância ξ é ajustado durante uma consulta kNN, *e.g.*, algoritmos *nearest-first* (viés *branch-and-bound*) ou *depth-first* (viés em profundidade) [Yianilos 1993, Chávez et al. 2001, Hetland 2009, Chen et al. 2017].

Esse estudo aborda esse ponto em aberto ao discutir uma comparação experimental de oito métodos de escolha de pivôs acoplados aos índices Omni *k*d-Tree e VP-Tree. Ambos os índices foram implementados para executar buscas kNN com o algoritmo de viés *incremental* e ótimo em cálculos de distância *distance-browsing* [Hjaltason and Samet 2003]. As avaliações experimentais sugerem que há diferenças de desempenho derivadas da escolha dos pivôs e que, ainda assim, diferentes índices podem ser melhor ajustados por métodos distintos de escolha de pivôs.

O restante desse trabalho é organizado da seguinte forma. A Seção 2 introduz os conceitos preliminares, enquanto a Seção 3 descreve os materiais e métodos. A Seção 4 apresenta a avaliação realizada e a Seção 5 discute os resultados encontrados.

2. Preliminares

Omni *k*d-Tree. Seja um conjunto de dados $\mathcal{O}$ e outro de pivôs $\mathcal{P} = \{p_1, p_2, \ldots, p_t\} \subseteq \mathcal{O}$, uma Omni *k*d-Tree particiona $\mathcal{O}$ de acordo com a mediana das distâncias a um pivô $p_j \in \mathcal{P}$. As distâncias $\delta(o_i, p_j)$ inferiores à mediana são colocadas na partição esquerda e as demais na partição direita. Essa divisão se repete recursivamente considerando que *(i)* as partições geradas contém dois conjuntos de dados associados e *(ii)* a divisão das

partições geradas deve usar o próximo pivô $p_{(j+1)}$ *mod* $t \in \mathcal{P}$. Essa construção segue até que uma quantidade mínima de objetos C_T por partição seja alcançada.

VP-Tree. Dado um conjunto de dados $\mathcal{O}$, uma VP-Tree divide os objetos de $\mathcal{O}$ em dois nós de acordo com um pivô $p \in \mathcal{O}$, com a mediana das distâncias μ_p entre os objetos de $\mathcal{O}$ e p e a distância máxima M_p de p a todo $o_i \in \mathcal{O}$. Objetos o_i cuja distância seja $\delta(o_i, p) \leq \mu_p$ são alocados à esquerda e os demais à direita. Essa construção por bolas fechadas segue recursivamente até que um número de objetos C_T por nó seja alcançado.

Métodos de Escolha de Pivôs. Fixos os conjuntos de dados $\mathcal{O}$ e pivôs $\mathcal{P}$, Chen et. al (2022) introduziram a seguinte taxonomia para métodos de seleção de pivôs: *(i)* abordagens que usam apenas a distribuição de distâncias dentre os objetos de $\mathcal{P}$ (*e.g.*, `C-HULL`, `S-S-SELECTION`, `GNAT`, e `M-SEPARATED`), *(ii)* estratégias que consideram a distribuição de distâncias de $\mathcal{O}$ para $\mathcal{P}$ (*e.g.* `PCA` e `M-VARIANCE`), e *(ii)* técnicas que usam apenas a distribuição de distâncias dentre os objetos de $\mathcal{O}$ (*e.g.*, `kMEDOIDS` e `SELECTION`). Nesse estudo são comparados oito métodos representativos de todas essas três categorias, que são exemplificados na Figura 1 e se encontram detalhados na sequência.

`kMEDOIDS`. Escolhe pivôs medóides $p \in \mathcal{P}$ que são tais que $\mathcal{P} = \{o_i \mid o_i \in R, \min\left(\sum_{o_j \in R} \delta(o_i, o_j)/|R|\right)\}$ para relações $R = \{o_i \in \mathcal{O}, \delta(o_i, p) \leq \delta(o_i, p_k)\ \forall p_k \in \mathcal{P} \setminus \{p\}\}$ associadas às entradas $p \in \mathcal{P}$.

`C-HULL`. Seleciona os objetos que, idealmente, compõem o fecho-convexo de $\mathcal{O}$ tal que $\sum_{p_i, p_j \in \mathcal{P}} \delta(p_j, p_j) \geq \sum_{o_i, o_j \in R} \delta(o_j, o_j), R \subseteq \mathcal{O} \setminus \mathcal{P}, |R| = |\mathcal{P}|$.

`PCA`. Escolhe os objetos que possuem as maiores co-variâncias de distâncias em $\mathcal{O}$ (medidas pelos autovalores da distribuição de distâncias de $\mathcal{O}$) como pivôs.

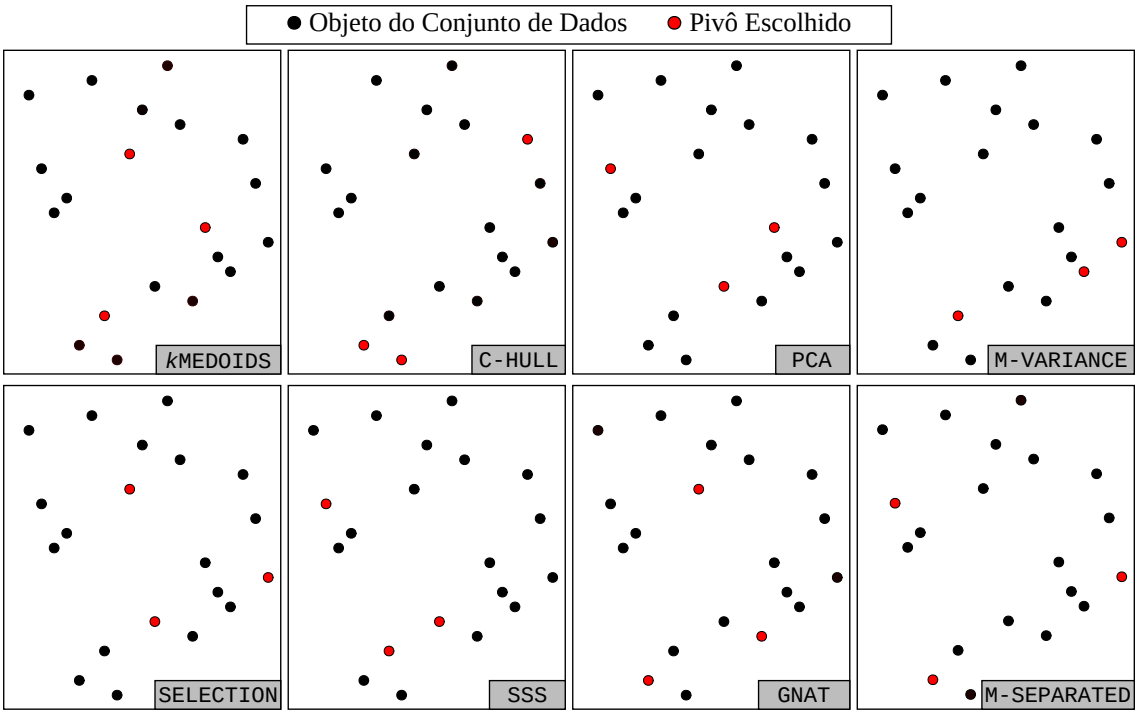

Figura 1. Escolhas para um mesmo conjunto de dados e $|\mathcal{P}| = 03$ pivôs.

M-VARIANCE. Seleciona os objetos de maior variância na distribuição de distâncias de $\mathcal{O}$, *i.e.* $\sum_{p\in\mathcal{P}} \sigma(\{\delta(o_i,p)\}) \geq \sum_{o_j\in R} \sigma(\{\delta(o_i,o_j)\})\ \forall o_i \in \mathcal{O}\setminus\mathcal{P}, R\subseteq\mathcal{O}\setminus\mathcal{P}, |R|=|\mathcal{P}|$.

SELECTION. Escolhe os objetos que maximizem a poda pelo Limite Inferior de $\mathcal{P}$ para os demais elementos em $\mathcal{O}$, *i.e.*, $\sum_{p\in\mathcal{P}}\sum_{o_j\in\mathcal{O}\setminus\mathcal{P}} |\delta(p,o_i) - \delta(o_i,o_j)| \leq \sum_{r\in R}\sum_{o_j\in\mathcal{O}\setminus\mathcal{P}} |\delta(r,o_i)-\delta(o_i,o_j)|,\ \forall o_i\in\mathcal{O}\setminus\mathcal{P}, R\subseteq\mathcal{O}\setminus\mathcal{P}, |R|=|\mathcal{P}|$.

S-S-SELECTION(SSS). Escolhe pivôs afastados entre si por pelo menos $\alpha.M_{\mathcal{O}}$, onde $M_{\mathcal{O}}$ é a maior distância entre os objetos em $\mathcal{O}$ e $\alpha\in\mathbb{R}_+$ é uma escala informada pelo usuário.

GNAT. Seleciona um pivô com o critério `C-HULL` e, então, escolhe iterativamente objetos com a maior dentre as menores distâncias para os pivôs selecionados anteriormente.

M-SEPARATED. Seleciona um pivô com o critério `C-HULL` e, então, escolhe iterativamente objetos com o maior soma de distâncias para os pivôs anteriores.

3. Materiais e Métodos

Algoritmo de busca. Foi implementado o mesmo algoritmo de busca para os índices Omni *k*d-Tree e VP-Tree para uma comparação mais justa: o *distance-browsing*, ótimo em cálculos de distância [Hjaltason and Samet 2003]. A rotina roda um mecanismo de decisão iterativo sobre duas filas de prioridade: a primeira contém partições não visitadas ordenadas pelo limite de distância mínimo (δ_{min}) ao objeto de consulta (desempates pelo menor limite máximo δ_{max}), e a segunda com objetos das partições visitadas ordenados pela distância à referência da busca. Na prática, δ_{min} e δ_{max} dependem de cada índice. A Figura 2 mostra a proposta desse estudo para a construção desses valores na Omni *k*d-Tree e VP-Tree. O algoritmo começa inserindo a partição raiz na primeira fila. A cada iteração, se a partição no topo da primeira fila estiver mais próxima do elemento de consulta do que o objeto na segunda fila então ela é removida e analisada, caso contrário o objeto no topo da segunda fila é recuperado como o próximo elemento do conjunto resposta. Caso a partição analisada seja uma folha seus elementos são carregados na segunda fila, do contrário suas filhas são carregadas do disco e inseridas na primeira fila. A rotina encerra com k recuperações.

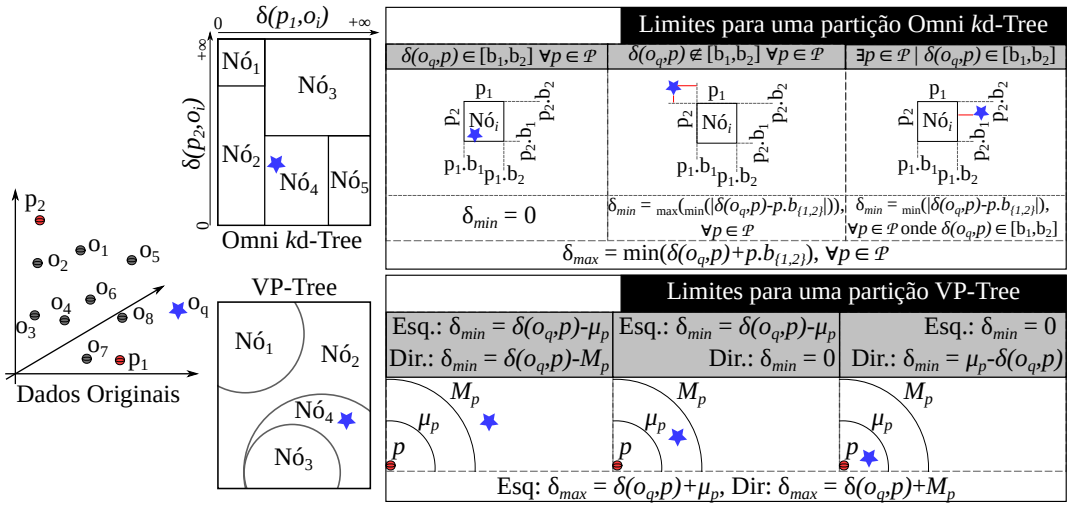

Figura 2. Construção dos limites da menor (δ_{min}) e maior (δ_{max}) distância dentro de uma partição dos índices baseados em pivôs para um elemento de busca o_q.

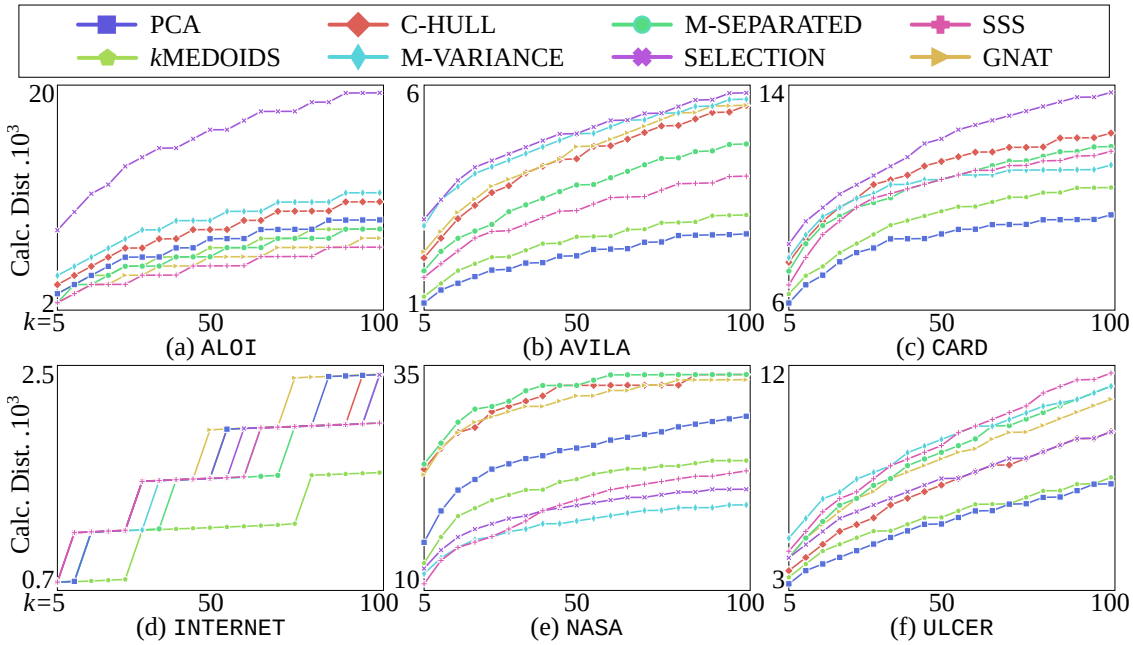

Figura 3. Impacto dos pivôs no índice Omni *k*d-Tree.

Infraestrutura de Teste. As implementações foram realizadas em C++v9 no *framework* Qt 5.15.1 com validação por testes unitários. Os experimentos foram executados em um *cluster* com 48 *cores* AMD Opteron 2.2GhZ, 96GB RAM, disco SATA de 1TB, rodando a distribuição Linux QLustar. Todas consultas foram efetuadas em paralelo com a coleta das métricas determinísticas de cálculos de distância e acessos a disco.

4. Avaliação Experimental

Conjuntos de Dados. O número de pivôs da Omni *k*d-Tree foi definido em função da dimensionalidade intrínseca $\mathcal{D}$ dos conjuntos d-dimensionais avaliados, *i.e.*, $|\mathcal{P}| = \lceil \mathcal{D} \rceil$, sendo $\mathcal{D}$ estimado como em [Chávez et al. 2001]. Foram utilizados seis conjuntos de dados[1,2,3] na avaliação, a saber *(i)* ULCER ($d = 12, \mathcal{D} = 4, |\mathcal{O}| \approx 4 \cdot 10^4$), *(ii)* NASA ($d = 20, \mathcal{D} = 6, |\mathcal{O}| \approx 4 \cdot 10^4$), *(iii)* INTERNET ($d = 11, \mathcal{D} = 2, |\mathcal{O}| \approx 6 \cdot 10^4$), *(iv)* CARD ($d = 23, \mathcal{D} = 3, |\mathcal{O}| \approx 3 \cdot 10^4$), *(v)* AVILA ($d = 10, \mathcal{D} = 2, |\mathcal{O}| \approx 2 \cdot 10^4$) e *(vi)* ALOI ($d = 13, \mathcal{D} = 2, |\mathcal{O}| \approx 1 \cdot 10^5$). Todas consultas foram realizadas com a função $\delta = L_2$.

Parâmetros da avaliação. Cada conjunto analisado foi dividido em uma partição *holdout* onde 90% dos objetos foram indexados e os demais 10% foram usados como elementos de consulta. Para cada um desses elementos foram realizadas 20 buscas kNN com $k \in \{5, 10, \ldots, 100\}$, totalizando mais de $1 \cdot 10^6$ consultas. Nós-folhas foram configurados para armazenar um número de objetos de até $C_T = 1\%$ dos conjuntos indexados.

Desempenhos individuais[4]. A Figura 3 apresenta a mediana de cálculos de distâncias necessárias para se resolver uma busca kNN para um intervalo crescente de vizinhos sobre o índice Omni *k*d-Tree. Os resultados indicam que há uma separação entre o desempenho

[1] https://archive.ics.uci.edu/ml/datasets.php
[2] https://aloi.science.uva.nl/
[3] https://github.com/gu-blanco/qtdu
[4] Cálculos de distância e acessos ao disco apresentaram comportamentos análogos.

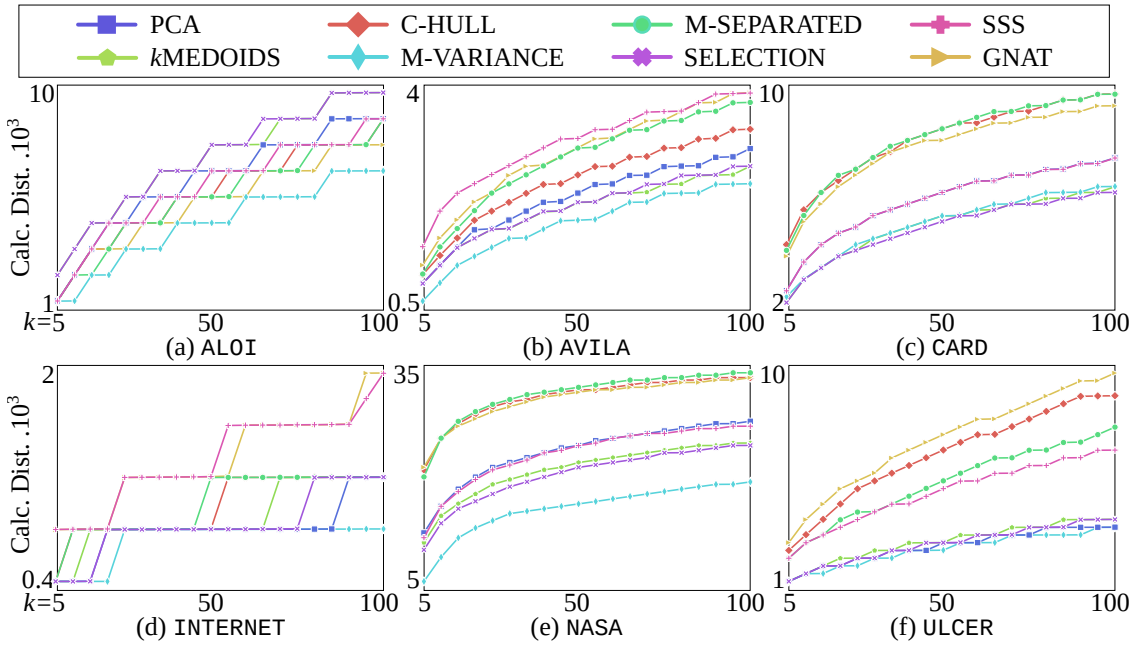

Figura 4. Impacto dos pivôs no índice VP-Tree.

dos métodos avaliados para a maioria dos conjuntos e valores de k analisados. Esse mesmo comportamento é observado na Figura 4, que apresenta o impacto dos métodos de seleção de pivôs sobre o índice VP-Tree. Em todas as avaliações, os métodos de pivôs reduziram substancialmente o número de cálculos de distância na comparação com uma busca sequencial, *i.e.*, a pior abordagem para a escolha de pivôs sempre foi mais eficiente do que utilizar uma busca kNN sequencial comparando e ordenando por distância todos elementos do conjunto de dados ao objeto consultado.

Não obstante, é possível observar que uma mesma estratégia de escolha de pivôs em um conjunto de dados pode se comportar de forma diferente nos dois índices analisados. Por exemplo, para o conjunto de dados `CARD`, o método `SELECTION` foi a abordagem com melhor resultado no índice VP-Tree, porém a de pior desempenho no índice Omni *k*d-Tree. Para consolidar essas avaliações individuais e generalizar o comportamento das estratégias de seleção de pivôs comparadas, foi realizado um *ranking* dos métodos em função do seu desempenho por conjunto de dados e vizinhança.

Desempenhos consolidados. A Figura 5 apresenta na forma de um mapa de calor o resultado da classificação dos métodos de escolha de pivôs, do melhor (1^o) ao pior (8^o) desempenho para todos os cenários analisados considerando a métrica de cálculos de distância. O mapa de calor consolidado mostra a distribuição do desempenho de cada método (linha) em cada um dos dois índices analisados e esses resultados permitem identificar métodos que apresentaram desempenhos consistentemente pobres em ambos os índices, como as abordagens `GNAT` e `C-HULL`. A Figura 6 resume as distribuições para as abordagens que obtiveram ou o melhor (Top-1) ou um dos três melhores (Top-3) desempenhos na execução das consultas kNN (calculado sobre as Figuras 3 e 4).

Os resultados mostram que a estratégia `M-VARIANCE` foi dominante na escolha de bons pivôs para a VP-Tree, sempre obtendo um dos três melhores desempenhos. As estratégias `SELECTION` e *k*`MEDOIDS` também obtiveram bons desempenhos para esse

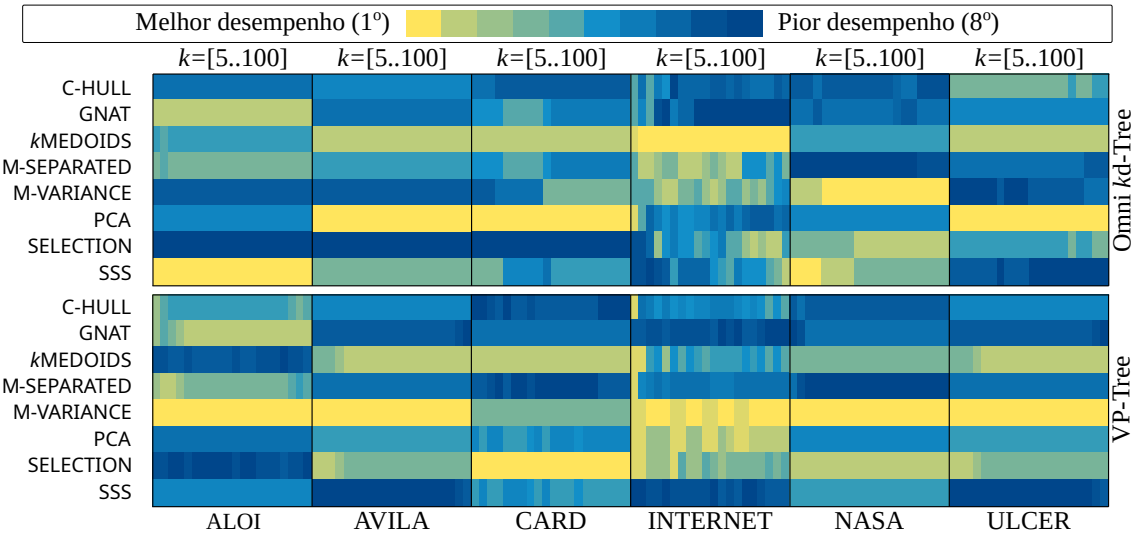

Figura 5. *Ranking* consolidado para os métodos de escolha de pivôs.

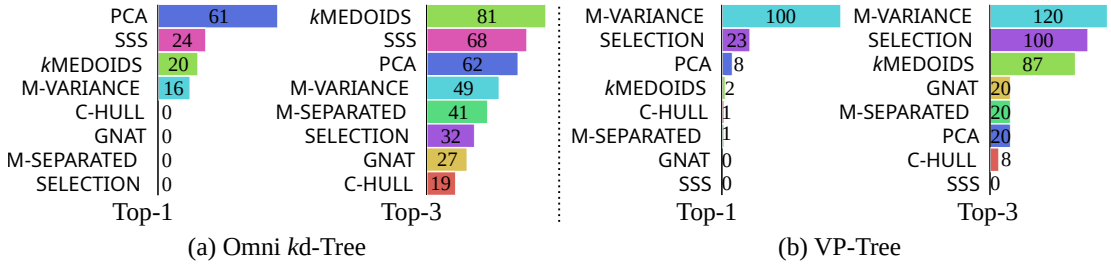

Figura 6. *Ranking* de desempenhos Top-1 e Top-3 por índice (máx. 120).

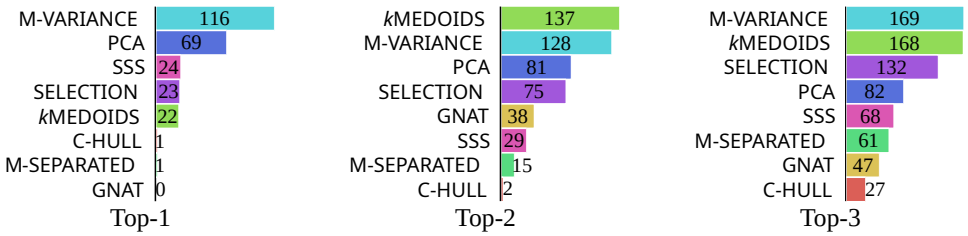

Figura 7. *Ranking* geral de desempenhos (máx. 240).

índice. Por outro lado, uma maior variabilidade no desempenho das estratégias comparadas foi observada para o índice Omni *k*d-Tree. Por exemplo, a estratégia `PCA` obteve o melhor desempenho em metade das consultas, mas sequer ficou entre as três melhores opções para a outra metade. Nesse cenário de maior variância, as estratégias `kMEDOIDS` e `S-S-SELECTION` obtiveram desempenhos melhores e mais estáveis.

Melhores e piores desempenhos. A classificação geral de desempenhos para os dois índices avaliados é mostrada na Figura 7. Os resultados consolidados reforçam ainda mais a variância do método de escolha `PCA` e o desempenho pobre das estratégias `M-SEPARATED`, `GNAT` e `C-HULL`. Por outro lado, as abordagens `M-VARIANCE` e `kMEDOIDS` estiveram entre as três melhores em 70% das consultas realizadas, revezando-se entre primeiro e segundo lugar em mais da metade das consultas. Os ganhos de ambos os métodos sobre o último concorrente no consolidado geral (`C-HULL`) são de até 24,0% (média 20,2% $\pm$ 31,1 –

M-VARIANCE; média 24,0% $\pm$ 28,8 – kMEDOIDS), o que ilustra a probabilidade de melhoria de desempenho de uma busca kNN executada por qualquer um dos dois índices desde que os pivôs sejam escolhidos por uma das duas estratégias vencedoras.

5. Conclusão e Trabalhos Futuros

Esse trabalho examinou experimentalmente o impacto de oito estratégias de seleção de pivôs aplicadas sobre dois índices métricos para a otimização de buscas kNN. Os resultados mostraram uma separação entre o comportamento mediano dos métodos em todas as consultas realizadas, o que permitiu classificar as abordagens competidoras em um *ranking* de desempenho para cada um dos conjuntos de dados e vizinhança k analisados. Considerando o desempenho consolidado, os métodos M-VARIANCE e kMEDOIDS apresentaram os melhores comportamentos, enquanto as abordagens M-SEPARATED, GNAT e C-HULL obtiveram os piores resultados. Não obstante, as avaliações também mostraram que diferentes índices podem ser melhor ajustados por diferentes métodos (*e.g.*, SELECTION para VP-Tree e S-S-SELECTION para Omni *k*d-Tree), muito embora ajustes específicos estejam sujeitos a alta variância com relação ao caso geral (*e.g.*, método PCA). Como trabalhos futuros, pretende-se investigar o impacto dos métodos de seleção sobre outros índices métricos implementados para executar o algoritmo de busca *distance-browsing*.

Agradecimentos. Marcos Bedo está em afastamento do INFES/UFF na FMRP/USP (G. #21/06564-0 – Fundação de Amparo à Pesquisa do Estado de São Paulo). Esse estudo foi apoiado pela Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (G. E47/2021-SEI260003/016517/2021-R210.107/2022 – Wagner R. Telles e Rodolfo A. Oliveira).

Referências

Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J. L. (2001). Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321.

Chen, L., Gao, Y., Zheng, B., Jensen, C., Yang, H., and Yang, K. (2017). Pivot-based metric indexing. *Proceedings of the VLDB Endowment*, 10(10):1058–1069.

Hetland, M. (2009). The basic principles of metric indexing. In *Swarm Intelligence for Multi-objective Problems in Mining*, pages 199–232. Springer.

Hjaltason, G. and Samet, H. (2003). Index-driven similarity search in metric spaces. *ACM Transactions on Database Systems*, 28(4):517–580.

Mao, R., Zhang, P., Li, X., Liu, X., and Lu, M. (2016). Pivot selection for metric-space indexing. *International Journal of Machine Learning and Cybernetics*, 7(2):311–323.

Traina Jr, C., Filho, R., Traina, A., Vieira, M., and Faloutsos, C. (2007). The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. *The VLDB Journal*, 16(4):483–505.

Yianilos, P. (1993). Data structures and algorithms for nearest neighbor. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, volume 66, page 311. SIAM.

Zhu, Y., Chen, L., Gao, Y., and Jensen, C. (2022). Pivot selection algorithms in metric spaces: A survey and experimental study. *The VLDB Journal*, 31(1):23–47.