Uniwersytet Jagielloński
Wydział Matematyki i Informatyki

Piotr Helm
Nr albumu: 1132708

Analiza algorytmów budowania coresetu dla problemu K-means

Praca licencjacka
na kierunku Informatyka Analityczna

Praca wykonana pod kierunkiem
dr Iwona Cieślik
Instytut Informatyki Analitycznej

Kraków, wrzesień 2020

Streszczenie

W pracy przedstawię stan wiedzy na temat budowania coresetów w kontekście algorytmu K-means. W szczególności poruszę techniki takie jak geometryczna dekompozycja oraz losowe próbkowanie bazując na badaniach [3].

Celem pracy jest przedstawienie wyników teoretycznych [3] jak i implementacja technik budowania coresetów, które mogą mieć zastosowanie praktyczne [1] [2]. Następnie porównam ich działanie testując je na zbiorach punktów z dwuwymiarowej przestrzeni euklidesowej.

Wstęp

More is more to jedna z podstawowych doktryn związanych z szeroko rozumiałym Big Data. Więcej danych to więcej informacji, które analizujemy licząc na poznanie ukrytych zależności. W erze Big Data skalowalność rozwiązań jest szczególnie ważna dlatego celem wielu naukowców jest dostarczenie kompromisu pomiędzy szczegółowością informacji a wymaganiami pamięciowymi. Tutaj warto zwrócić uwagę na dużą wartość takich rozwiązań w praktycznych zastosowaniach.

Sketch-and-solve to popularny paradygmat, który zakłada separacje algorytmu agregującego dane od właściwego algorytmu analizującego. Główną ideą jest redukcja danych tak aby ich rozmiar nie był zależny od wejściowych danych lub tylko *trochę* od nich zależał. Następnie aplikowany jest właściwy algorytm, który jest mniej zależy od początkowego rozmiaru danych. W rezultacie wykonuje swoją pracę szybciej, a niekiedy nawet lepiej. Dodatkową zaletą jest fakt, że w większości przypadków nie jest konieczna modyfikacja algorytmu analizującego.

Niestety w kontekście tego paradygmatu największym wyzwaniem jest znalezienie kompromisu pomiędzy stratą jakości danych a ich rozmiarem. To jak określamy charakterystykę ważnych informacji jest ściśle zależne od aplikacji danych. Coresety są strukturą algorytmiczną, która ma na celu indentyfikacje takich cech oraz określenie akceptowalnego kompromisu dla różnych funkcji celu.

Mówiąc ogólniej, mamy na wejściu zbiór danych A, zbiór potencjalnych rozwiązań C oraz funkcję celu f. Chcemy znaleźć istotnie mniejszy zbiór danych S, który dla każdego potenlego rozwiązania $c \in C$ daje $f(S, c)$ *dobrze* aproksymujące $f(A, c)$.

Algorytmy budujące coresety są aplikowalne do wielu problemów klasteryzacji. W tej pracy skupię się na konstrukcjach dla problemu K-means, który należy do klasy problemów *NP-trudnych*. Najpopularniejszym algorytmem heurystycznym dla tego problemu jest algorytm Lloyd'a. Z uwagi na jego niską skalowalność jest on idealnym kandydatem do optymalizacji poprzez odpowiednią konstrukcję coresetu.

Notacja i niezbędne definicje

K-means

Zacznijmy od zdefiniowania problemu dla którego będziemy analizować konstrukcje coresetów.

Definicja 2.1. *Problem K-means.* Niech X to zbiór punktów z $\mathbb{R}^d$. Dla danego X chcemy znalźć zbiór k punktów Q, który minimalizuje funkcję $\phi_X(Q)$ zdefiniowaną następująco:

$$\phi_X(Q) = \sum_{x \in X} d(x, Q)^2 = \sum_{x \in X} \min_{q \in Q} ||x - q||_2^2$$

Powyższa definicja zakłada, że działamy w przestrzeni euklidesowej. Uogólnioną wersję można zdefiniować analogicznie, zamieniając d na odpowiednią funkcję miary w danej przestrzeni.

Definicja 2.2. *Problem K-means - wersja ważona.* Niech C to zbiór punktów z $\mathbb{R}^d$ oraz niech w będzie funkcją $C \to \mathbb{R}_{\geq 0}$. Dla danego X oraz funkcji w chcemy znalźć zbiór k punktów Q, który minimalizuje funkcję $\phi_X(Q)$ zdefiniowaną następująco:

$$\phi_X(Q) = \sum_{x \in X} w(x) d(x, Q)^2$$

Ewaluację funkcji ϕ dla optymalnego rozwiązania oznaczamy $\phi^k_{OPT}(X)$.

Coreset

To jak definujemy coreset ściśle zależy od problemu, który optymalizujemy. Zacznijmy od podstawowej definicji coresetu dla problemu *K-means*.

Definicja 2.3. *Coreset.* Niech X to zbiór punktów z $\mathbb{R}^d$ oraz niech Q to dowolny podzbiór X rozmiaru co najwyżej k. Zbiór C nazywamy (ϵ, k) coresetem jeżeli zachodzi:

$$|\phi_X(Q) - \phi_C(Q)| \leq \epsilon \phi_X(Q)$$

Zauważmy, że taka definicja daje nam bardzo moce gwarancje teoretyczne. Ewaluacji coresetu $\phi_C(Q)$ musi aproksymować $\phi_X(Q)$ z dokładnością $1+\epsilon$ jednocześnie dla dowolnego zbioru k kandydatów na rozwiązanie Q w kontekście całego zbioru X. Jest to na tyle istotne, że w literaturze odróżnia się tą wersję nazywając ją *Strong Coreset*.

Lightweight Coreset

Lightweight coreset

TBA

Konstrukcja

TBA

Analiza

TBA

Geometryczna Dekompozycja

TBA

Analiza

TBA

Bibliografia

[1] Bachem, O., Lucic, M., and Krause, A. Scalable k-means clustering via lightweight coresets.

[2] Har-Peled, S., and Mazumdar, S. On coresets for k-means and k-median clustering. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2004), STOC '04, Association for Computing Machinery, p. 291–300.

[3] Munteanu, A., and Schwiegelshohn, C. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intell. 32*, 1 (2018), 37–53.

Spis treści