2F1-2in

階層的時間刻みを用いた k-means 高速化

Hierarchical Blocked Time-step Algorithm for Accelerating k-means

唐恒進 Hengjin Tang　松林達史 Tatsushi Matsubayashi　澤田宏 Hiroshi Sawada

日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, NTT Corporation

The k-means algorithm is widely used for discovering clusters in data. In this paper, we propose a new acceleration method using hierarchical blocked time-step algorithm. Our algorithm can skip distance calculations efficiently by changing the frequency of searching nearest cluster centers while keeping the clustering accuracy. Our experiments show our algorithm reduces distance calculations about $20-60\%$.

1. はじめに

データマイニング分野において k-means は主要なアルゴリズムの 1 つである．k-means は，非階層型クラスタリングのアルゴリズムであり，データを K 個のクラスタに分割する．代表的な手法としては Lloyd 法 [Lloyd 82] が広く利用されているが，各オブジェクトとクラスタ中心間の距離計算回数が，オブジェクト数 N，クラスタ数 K，収束までにかかる繰り返し回数 I_{end} の場合に NKI_{end} となり，計算処理に時間がかかるという問題がある．

本研究では，階層的時間刻みを用いた k-means クラスタリング法の高速化手法を提案する．提案手法では，クラスタリングの代表的な手法である k-means 法に対し，最近傍クラスタ中心探索の実施判定頻度をオブジェクト毎に独立に与える．オブジェクトの所属するクラスタ中心の移動距離に応じて判定頻度を変えることによって，オブジェクトとクラスタ中心間の距離計算を効率的に省略し，最適化計算の高速化を実現した．

2. アルゴリズム

Lloyd 法と提案手法では，N 個の d 次元オブジェクトを K 個のクラスタに分割する処理を行っており，各オブジェクトを $\boldsymbol{x}_i = (x_i^1, x_i^2, \ldots, x_i^d)^T, i = 1, \ldots, N$，各クラスタ中心を $\boldsymbol{c}_j = (c_j^1, c_j^2, \ldots, c_j^d)^T, j = 1, \ldots, K$ とする．また，オブジェクト $\boldsymbol{x}_i$ が所属するクラスタ中心を $\boldsymbol{c}_{a_i}$ とする．目的関数は

$$J = \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{c}_{a_i}\|^2 \tag{1}$$

で定義され，最近傍クラスタ中心探索と，クラスタ中心更新の２つの処理による交互最適化を行う．

2.1 提案手法

Lloyd 法では，最近傍クラスタ中心探索処理において，すべてのオブジェクトとクラスタ中心間の距離計算を行っていた．これに対して提案手法では，階層的時間刻み法 [Matsubayashi and Yamada 07] を利用して，各オブジェクト $\boldsymbol{x}_i$ の現所属クラスタ中心の 1 周期前からの移動距離 $p_{a_i,t} (= d(\boldsymbol{c}_{a_i,t}, \boldsymbol{c}_{a_i,t-1}))$ に応じた更新頻度（階層的時間刻み）$\triangle t_i = 2^{k_i}$ で，各オブジェクトの所属判定を行う．指数 $k_i \in \{0, 1, \ldots, k_{\max}\}$ は，以下の式で更新される．

$$k_i = \begin{cases} k_i - 1 & \text{if } 2^{k_i-1} \leq \eta / p_{a_i,t} \leq 2^{k_i-1+1} \\ k_i + 1 & \text{if } \eta / p_{a_i,t} \geq 2^{k_i+1} \text{ and } t \bmod 2^{k_i+1} \\ k_i & \text{otherwise} \end{cases}$$

ここで η は定数で，数値計算の精度を決めるとともに，各階層での最大時間更新幅を与える．ここで図 1 に更新概念図を示す．

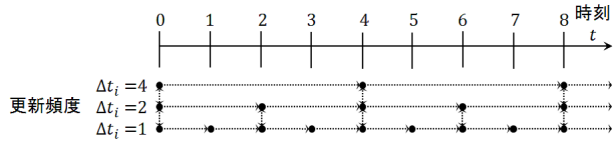

図 1: 階層的時間刻み法の時間更新概念図

各階層に分けられたオブジェクトは “•” 上に階層的時間を持ち，時刻 t が左から右に向かって進行し，t と同期したオブジェクトのみが階層的時間を更新する．この時 $\triangle t_i$ の変更は図の点線上の方向のみ許されることになる．

またさらに，計算の更新条件として時間刻みに上限 $\triangle t_{\max} (= 2^{k_{\max}})$ を設定する．上限を与えることにより，一定の間隔で定期的に全データの時間同期を行う．提案法では，階層的時間が時刻 t と同期したデータ集合に関してのみ，所属クラスタ探索処理が実行される．このとき，時刻 t において同期しているオブジェクト集合 S_t は

$$S_t = \{i \mid t_i + \triangle t_i = t\} \tag{2}$$

で定義され，そのデータ数を $N_{s,t}$ とする．ここで，t_i はオブジェクト $\boldsymbol{x}_i$ が最後に同期した時刻を表している．したがって，オブジェクトとクラスタ中心間の距離計算回数は，時刻 t においては 1 周期当たり NK から $N_{s,t}K$ に軽減できる．

3. 実験

提案法を人工データと実データに適用し，Lloyd 法と比較して省略できる距離計算の割合を示す．実験において初期クラスタ中心の生成には k-means++ [Arthur and Vassilvitskii 07] を用いた．

連絡先: 唐恒進，NTT サービスエボリューション研究所，tang.hengjin@lab.ntt.co.jp

収束条件は時刻 t において，Lloyd 法は 1 周期前と比較して目的関数値が変化なし，提案手法は全オブジェクトが同期し，かつ 1 周期前と比較して目的関数値が変化なしという設定とした．

実験は次の環境下で行った．Linux 64-bit Intel，CPU : 2.60GHz，8 コア，RAM : 2G，実装 : C 言語．また，表 1 に使用したデータを示す．

表 1: データセット

データ名	データ数	次元数
birch	100,000	2
kddcup98	95,412	56
mnist-50	60,000	50
covertype small	150,000	54
U-N10K-D02/08/32	10,000	2/8/32
U-N50K-D02/08/32	50,000	2/8/32

使用したデータのうち，birch，kddcup98，mnist-50，covertype small はそれぞれ，10×10 ガウス型クラスターによる人工データ，ダイレクトメールによる効果推定に関するデータ，手書き文字認識のラスターイメージ，森林の覆土の推定に関するデータに関する．また，U-NX-DY の各データは一様乱数によって生成された人工データで，X はデータ数を，Y は次元数を表している．

3.1 距離計算の省略率

アルゴリズムの性質上，Lloyd 法と提案手法では収束までにかかる繰り返し回数が異なる．時刻 t における Lloyd 法の繰り返し回数を $I_{t,L}$，提案手法の実質的な繰り返し回数を $I_{t,H}$ とし下記で定義する．

$$I_{t,L} = t, \tag{3}$$

$$I_{t,H} = \frac{\sum_{i=1}^{t} N_{s,t}}{N}. \tag{4}$$

繰り返し回数と目的関数値の変化の一例を図 2 に示す．図の横軸は繰り返し回数，縦軸は目的関数値を表している．図 2 より，提案手法は Lloyd 法と比較して，収束時の目的関数値は同等の精度を維持しつつ，より少ない繰り返し回数（距離計算回数）で収束していることが確認できる．

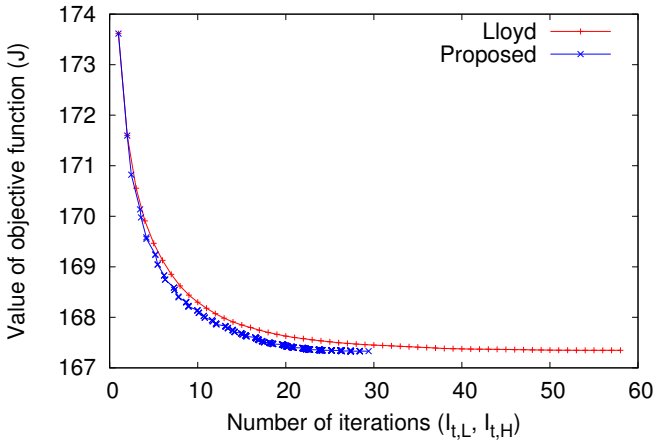

図 2: 繰り返し回数と目的関数値（birch, $K = 1500$）．

この時，距離計算の省略率 R_{dc} は，Lloyd 法を用いた場合の累計距離計算回数を 1 とした時の比として算出した．

$$R_{dc} = \frac{\sum_{t=1}^{T_{end,H}} N_{s,t} K}{\sum_{t=1}^{T_{end,L}} NK} = \frac{I_{T_{end,H},H}}{I_{T_{end,L},L}} \tag{5}$$

ただし，$T_{end,L}$，$T_{end,H}$ はそれぞれ Lloyd 法と提案手法の収束時の時刻を表している．

各データセットについて，Lloyd 法と比較して省略された距離計算の割合を図 3 に示す．図の横軸はクラスタ数，縦軸は距離計算の省略率を表している．図 3 より，birch，kddcup98，mnist-50，covertype small について，クラスタ数が少ない（$K = 3, 5, 10$）ケースを除き，提案手法は距離計算を効率的に削減できていることがわかる．これに対し，一様乱数で生成されたデータ（uniform）に対して提案手法を適用した場合は，距離計算の省略率は Lloyd 法と比べてあまり改善されない，または悪化するケースが多い．これは一様乱数で生成された人工データは，データ自体がクラスタ構造を持たず，クラスタ中心の移動にばらつきが生じにくいためだと考えられる．

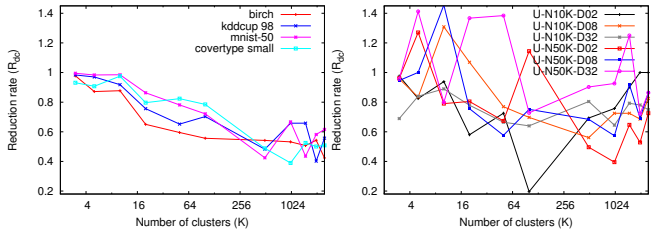

図 3: 距離計算の省略率（左 : birch，kddcup98，mnist-50，covertype small，右 : uniform）．

4. 結論と考察

Lloyd 法と比較した時に，提案手法により各オブジェクトとクラスタ中心間の距離計算が効率的に削減されることを確認した．また，提案手法は Lloyd 法との間で目的関数値の同一性は保証しないものの，分類精度で Lloyd 法に勝るケースもあり，精度面で見た場合はほぼ同等であると言える．

参考文献

[Arthur and Vassilvitskii 07] Arthur, D. and Vassilvitskii, S. : k-means++: The advantages of careful seeding, *Proc. of SDM 2007*, pp.1027-1035, 2007.

[Matsubayashi and Yamada 07] Matsubayashi, T. and Yamada, T. : A Force-directed Graph Drawing based on the Hierarchical Individual Timestep Method, *International Journal of Electronics, Circuits and Systems*, 1(3), pp.427-432, 2007.

[Lloyd 82] Lloyd, S. P. : Least Squares Quantization in PCM, *IEEE Trans. Inf. Theory*, 28(2), pp.129-137, 1982.