D-014

ストリーム環境での位置情報を持つテキスト集合に対する類似検索

久保 幸平 †　　古賀 久志 †

1 はじめに

近年，IoT などの隆盛に伴いストリームデータを扱うアプリケーションが増え，ストリームデータに対する検索も盛んに研究されるようになった．ストリームデータに対する検索は，クエリやデータベースが時間経過に伴って変化することが特色で，その変化方法の違いに依って様々な問題定義が存在する．代表例としては，データベース内のオブジェトに有効期限があり，オブジェクトが追加されたり消滅したりする状況下での top-K クエリなどが挙げられる [1].

一方，Xu ら [2] は 1 個人のウェブページの閲覧履歴のようなヒストリを集合で表現し，情報推薦を目的として，集合に対する類似検索を取り扱った．新規にウェブページを閲覧するたびにヒストリが更新されクエリ集合が変化することから，このようなクエリは evolving query と呼ばれる．[2] では要素がアルファベットである集合を取り扱っているが，これは閲覧したオブジェクトを量子化できるという前提条件が必要である．

本研究では，この条件を緩和し，テキストを集合要素とする evolving query に対する類似検索を取り扱う．正確には，tweet のような位置情報を持つテキストを要素とする集合を考える．この類似検索問題は，クエリユーザ q を SNS 上で q が生成した空間テキストオブジェクト集合で特徴付け，q の類似ユーザを検索する状況をモデル化している．なお，位置情報を持つテキスト集合に対する類似検索に関しては，ストリームではない静的な環境下で Efstatthiade[3] が join クエリを実現するアルゴリズムを提案している．本研究では，これをストリーム環境におけるレンジ探索用に改変したアルゴリズムを提案する．

本論文の構成は以下のようになる．第 2 章で先行研究を，第 3 章で本研究における問題設定を述べる．第 4 章でベースラインアルゴリズム，第 5 章で提案アルゴリズムについて述べる．第 6 章で提案したアルゴリズムを実験で評価し，最後に第 7 章で結論を述べる．

2 先行研究

Efstathiades ら [3] は SNS 上で類似ユーザを検索する手法について研究した．この研究ではユーザ u を u が生成したオブジェクトで特徴付ける．具体的にはオブジェクトは twitter における tweet のような位置情報を持つテキストであり，空間テキストオブジェクトと呼ばれる．空間テキストオブジェクト o は位置情報 $o.loc = \langle x, y \rangle$ とテキスト情報 $o.doc$ を持つ．$o.doc$ はテキストに含まれる単語集合である．

[3] では，ユーザ間類似度が閾値 ϵ_u 以上となるユーザペアをすべて見つける join クエリを取り扱った．この join クエリを STPSjoin(Spatio-Textual Point-Set Similarity Join) と呼ぶ．ここで，ユーザペア u と u' 間の類似度は，u が生成した空間テキストオブジェクトの集合 D_u と u' が生成した空間テキストオブジェクトの集合 $D_{u'}$ 間の類似度 $\sigma(D_u, D_{u'})$ により表現する．$\sigma(D_u, D_{u'})$ は式 (1) により定義される．

$$\sigma(D_u, D_{u'}) = \frac{|M(D_u, D_{u'})| + |M(D_{u'}, D_u)|}{|D_u| + |D_{u'}|}. \quad (1)$$

式 (1) において，$M(D, D')$ は D' 中の少なくとも 1 つのオブジェクトとマッチした D 内のオブジェクトの集合を表す．したがって，式 (1) は，u と u' が生成したオブジェクトのうち，相手のオブジェクトとマッチしたオブジェクトの割合を表している．

式 (1) を計算するには，2 つのオブジェクトがマッチしているかどうかの判定が必要になる．そこで 2 つのオブジェクト o, o' がマッチすることの定義を与える．

o と o' の空間距離 $\delta(o, o')$ を $o.loc$ と $o'.loc$ 間のユークリッド距離とし，o と o' のテキスト類似度 $\tau(o, o')$ をテキストに含まれる単語集合の Jaccard 類似度

$$\tau(o, o') = \frac{|o.doc \cap o'.doc|}{|o.doc \cup o'.doc|}$$

とする．o と o' は，$\delta(o, o') \leq$ 閾値ϵ_{loc} かつ $\tau(o, o') \geq$ 閾値ϵ_{doc} という条件を満たすならば，マッチする．この定義の直感的な意味は，空間的に近く，テキスト情報も似たオブジェクト同士がマッチするということである．なお，閾値 ϵ_{loc}, ϵ_{doc} は join クエリの中で指定されるパラメータである．

3 ストリームデータに対する類似検索

本章では，本論文で取り扱う新しい問題であるストリーム STPS レンジ探索の設定について述べる．Efstathiades ら [3] らは類似度 $\sigma(D_u, D_{u'}) \geq \epsilon_u$ となるユーザペア u, u' をすべて見つける join クエリを取り扱ったが，本研究では特定の 1 ユーザ q をクエリとして，類似度 $\sigma(D_q, D_u) \geq \epsilon_u$ となるユーザ u をすべて見つけるレンジ探索を取り扱う．

さらに時間とともに新しい空間オブジェクトが生成さ

† 電気通信大学大学大学院情報理工学研究科
〒 182−8585 東京都調布市調布ヶ丘 151

れるストリーム環境を想定し，クエリユーザ q を，q が生成した空間テキストオブジェクトのうち時間的に新しい定数 w 個の空間テキストオブジェクトで特徴付ける．つまり，クエリユーザ q が保有するオブジェクト集合の D_q は

- ユーザ q の最新の w 個の空間テキストオブジェクトであり，$|D_q| = w$.

ということになる．

本研究では，ストリームデータをスライディングウィンドウモデルで管理する．更新時にスライディングウィンドウに新たに入った最新のクエリオブジェクトを IN，逆にスライディングウィンドウから追い出された最古のクエリオブジェクトを OUT とする．図 1 にスライディングウィンドウモデルを示す．

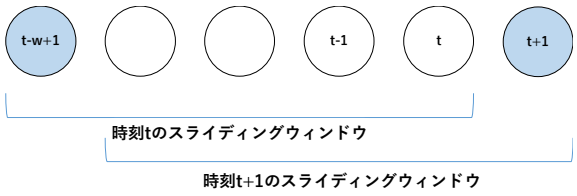

図 1　スライディングウィンドウモデル

一方，データベースには多数のユーザ集合 U が生成したオブジェクトが登録されている．U の各メンバ u に対して，u が生成した空間テキストオブジェクト集合を D_u とする．本研究ではデータベース側の D_u は時間に対して不変であると仮定する．

つまり，ストリーム STPS レンジ探索はクエリのみが時間と共に変化し，データベースは固定である．クエリ D_q が変化すると $\sigma(D_q, D_u)$ も変化するため，レンジ探索の結果も変化するが，D_q が変化する度に検索結果も更新することが要求される continunous なレンジ探索を本研究では取り扱う．

本論文では空間距離の 閾値ϵ_{loc} はクエリ毎に変わらない定数であると仮定する．

4　ベースラインアルゴリズム

本章では，ストリーム STPS レンジ探索を解く単純なアルゴリズムを説明する．本研究ではこれをベースラインアルゴリズムとする．

ベースラインアルゴリズムは，クエリ更新の度に D_u の全オブジェクトに対し，IN がマッチするかを決定する．また，消滅する OUT についてマッチング情報を更新する．つまり，このアルゴリズムでは，IN と OUT のみ処理する．IN と OUT 以外の D_q のオブジェクトについては，クエリ更新前に D_u の全オブジェクトに対してマッチ判定が完了しており，その判定結果はクエリ更新により変化することはないので処理しない．また，クエリオブジェクトとの距離が ϵ_{loc} 未満であるユーザのオブジェクトを高速に調べるために，空間を格子状に幅 ϵ_{loc} で分割して管理する．空間分割の例を図 2 に示す．オブジェクト o が所属するセルを 11 とする．この時，11 と隣接しないセルに属する u のオブジェクトは o からの距離が ϵ_{loc} 以上なので調べる必要がない．従って，オブジェクト o は 6,7,8,10,11,12,14,15,16 のセルに所属するオブジェクトのみとマッチ判定をすればよい．

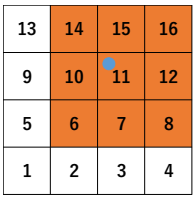

図 2　空間分割の例

4.1　IN の処理

IN の処理を以下に述べる．

1. IN の位置情報から所属するセル c を求める．
2. セル c と隣接する 9 個のセルにユーザ u のオブジェクトがあるか調べる．ない場合は非マッチと判定．
3. ある場合，隣接するセル内の u のオブジェクトとマッチング判定を行う．
4. IN が D_u のオブジェクトとマッチしていた場合，$|M(D_q, D_u)|$ を 1 増やす．
5. IN とマッチした u のオブジェクトで IN のみとマッチするオブジェクトがある場合，そのオブジェクトの数だけ $|M(D_u, D_q)|$ を増やす．

4.2　OUT の処理

OUT の処理を以下に述べる．

1. OUT が D_u のオブジェクトとマッチしていた場合，$|M(D_q, D_u)|$ を 1 減らす．
2. OUT のみとマッチするオブジェクトがある場合，そのオブジェクトの数だけ $|M(D_u, D_q)|$ を減らす．

5　提案手法

4 章で述べたベースライン手法は到着した IN に対して即座にマッチ判定を行うので全クエリオブジェクトが必ずマッチ判定される．これに対して提案手法では，マッチ判定を行うクエリオブジェクトを限定することによって計算時間の短縮を狙う．ユーザ u との類似度が ϵ_u を超えるかどうかの必要な分だけクエリオブジェクトのマッチ判定し，全オブジェクトに対してマッチ判定をしない．従って，各クエリオブジェクトは，

- マッチ
- 非マッチ
- 未処理

の 3 つの状態になる．マッチを青，非マッチを赤，未処理を緑で表現した場合の例を図 3 に示す．クエリ側は，図 3 の緑の集合である未処理クエリオブジェクトリストを持つ．到着するクエリオブジェクトのマッチ判定をできるだけ遅らせるという特性からこの提案手法を遅延評

価法と呼ぶ．

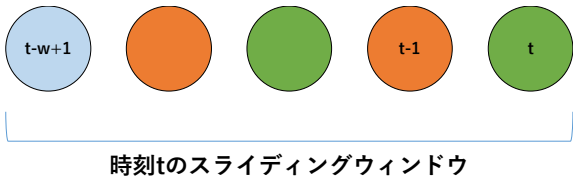

図 3　クエリオブジェクトの状態

遅延評価法の動作の流れを図 4 に示す．

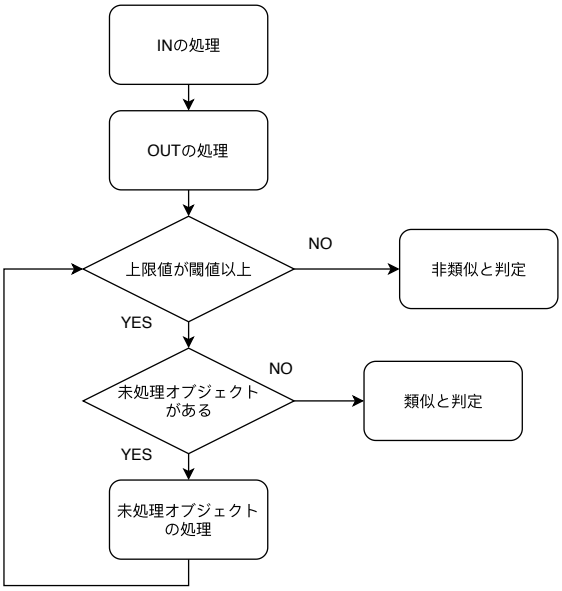

図 4　遅延評価法の動作の流れ

遅延評価法では，IN と OUT の処理をした後にマッチオブジェクト数と未処理オブジェクト数の合計から，ユーザ u との類似度の上限値 $\bar{\sigma}(D_q, D_u)$ を求める．そして，$\bar{\sigma}(D_q, D_u)$ が閾値 ϵ_u 未満である場合，q と u が非類似と判定しマッチ判定を打ち切る．$\bar{\sigma}(D_q, D_u) \geq \epsilon_u$ である場合，より正確な上限値を求めるために未処理クエリオブジェクトのマッチ判定を行う．この時，遅延評価法では，新しいものから順番にマッチ判定を行う工夫を行っている．この工夫は未処理のクエリオブジェクトが未処理のままスライディングウィンドウから離脱する機会を増やすことを狙っている．

マッチ判定の打ち切りにより，未処理のクエリオブジェクトが発生する．そのため，遅延評価法では u との類似度判定において次のデータ構造を持つ．

- M_u:ユーザ u に対する未処理クエリオブジェクトリスト．
- M_u^c:ユーザ u のセル c に対する未処理クエリオブジェクトリスト．

5.1　IN の処理

IN の処理を以下に示す．

1. IN の位置情報から所属するセル c を求める．
2. IN に含まれる単語集合 T を作る．
3. セル c と隣接するセル c' 内で $\forall t \in T$ を持つユーザ u のオブジェクトがあるか調べる．ある場合，マッチの可能性有りと判定し IN を未処理クエリオブジェクトリストに加える．ない場合，非マッチと判定する．

また，単語の共有判定を高速に行うために，各セル c' は c' 内のオブジェクトに含まれる各単語 t に対する転置インデックスを持つ．但し，t に対応する転置インデックスには，単語 t を含むオブジェクトではなく，そのオブジェクトを生成したユーザが登録される．

5.2　OUT の処理

OUT の状態で処理が異なる．

マッチ状態:　$|M(D_q, D_u)|$ を 1 減らす．OUT のみとマッチするオブジェクトがある場合，そのオブジェクトの数だけ $|M(D_u, D_q)|$ を減らす．

非マッチ状態:　何もしない

未処理状態:　OUT を未処理クエリオブジェクトリストから取り除く

5.3　未処理クエリオブジェクトの処理

未処理クエリオブジェクトの処理以下に示す．

1. 未処理クエリオブジェクトリスト M_u の中で最新のものを取り出す.（以下 o と示す）
2. $M_u^{c'}$ に o を含んでいるセル c' に対してユーザ u のセル c' のオブジェクト集合である $D_u^{c'}$ と o のマッチ判定を行う．
3. 未処理クエリオブジェクトリストから o を取り除く

この処理を $\bar{\sigma}(D_q, D_u) < \epsilon_u$ となるか未処理オブジェクトがなくなるまで繰り返す．

6　実験

本研究では，遅延評価法の有効性を示すために，ベースラインアルゴリズムと実行時間を比較した．また，遅延評価法の工夫の一つである新しいものから順番にマッチ判定を行うことの評価を行うために次のアルゴリズムを実装した．

- 未処理クエリオブジェクトを最新オブジェクトからではなく，古いものからマッチ判定することに変更した．

本研究は人工データを用いて性能評価実験を行った．データベースは以下のように作成した．1000 人のユーザが 200 個ずつ合計で 20 万個のオブジェクトを生成する．1 つの空間オブジェクトに含まれる単語の個数は 10 個とした．そして，ユーザごとに偏りを持たせるために，ユーザ u のオブジェクトは次のように生成する．ユーザ u のオブジェクトの位置情報は次のように決定する．

- ランダムに重心を決定した 10 個の 2 次元正規分布

を $\{N_1, N_2, ..., N_{10}\}$ 定める．
- 各ユーザを $N_1 \sim N_{10}$ までのどれかに割り当てる．
- ユーザ u のオブジェクトの位置は割り当てられた正規分布に従って発生させる．

ユーザ u のオブジェクトのテキスト情報は次のように決定する．

- 互いに共通単語を持たない M 個の単語集号を 10 個生成する．
- 生成された単語集号を $\{B_1, B_2, ..., B_{10}\}$ とする．
- 各ユーザ u_i を 10 個の単語集号のどれか 1 つに割り当てる．割り当てられた単語集号を B_i とする．
- ユーザ u_i のオブジェクトは 10 単語のうち 7 単語は 10000 語からランダムに選択し，残り 3 個は B_i の M 単語からランダムに選択する．

また，クエリもデータベースと同様に作成した．

本研究では，1000 個のユーザに対してクエリユーザに対する類似度が ϵ_u 以上のユーザの集合を検索する実験を行った．時刻 t を 1 から 10000 まで進め，10000 回の類似検索の実行時間を評価する．また，本研究で扱う基本的なパラメータは $\epsilon_{loc} = 0.05, \epsilon_{doc} = 0.05, \epsilon_u = 0.2$,, ユーザ特有の単語数 $M = 50$ である．

6.1 距離閾値

この実験では距離閾値 ϵ_{loc} の値を変更した場合の各アルゴリズムの実行時間を評価した．

実験結果を図 5 に示す．図 5 から遅延評価法がどのパラメータでもベースラインよりも実行時間を大幅に短くなっていることが示された．また，遅延評価法がどのパラメータでも古い未処理オブジェクトから評価するアルゴリズムよりも実行時間が短くなった．したがって，遅延評価法が有効であるといえる．

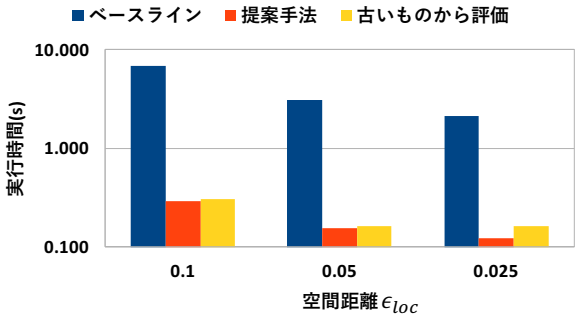

図 5　様々な距離閾値に対する実行時間

6.2 ユーザ間類似度閾値

この実験ではユーザ間類似度閾値 ϵ_u の値を変更した場合の各アルゴリズムの実行時間を評価した．

実験結果を図 6 に示す．図 6 から図 5 と同様の結果が得られた．しかし，遅延評価法ではユーザ間類似度閾値を利用した枝刈りを行ってるため ϵ_u が大きくなると実行時間が短くなることを期待したが，ユーザ間類似度閾値が変わっても遅延評価法の実行時間に影響がなかった．したがって，遅延評価法のユーザ間類似度による枝刈りがあまり効いていないと言える．

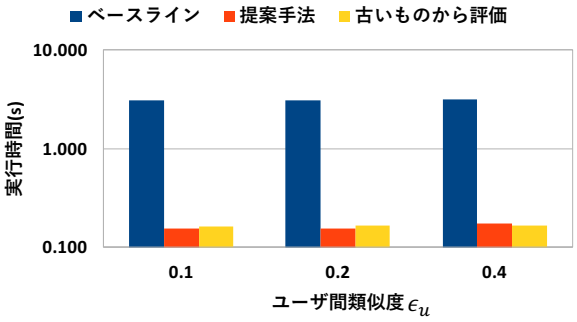

図 6　様々なユーザ間類似度閾値に対する実行時間

7 まとめ

本研究は [3] で扱った空間テキストオブジェクト集合の類似検索問題に時間情報を加えたストリーム STPS レンジ探索問題を扱った．本研究ではストリーム STPS レンジ探索を効率的に扱うアルゴリズムを提案した．提案したアルゴリズムにおける工夫は 2 つある．1 つ目はユーザ間類似度の上限値を利用し，マッチングの判定回数を削減した．2 つ目はクエリの未処理オブジェクトを新しいものからマッチ判定することにより，マッチング判定なしでスライディングウィンドウから離脱するクエリオブジェクト数を増やした．

そして，人工データを用いて提案したアルゴリズムとベースラインアルゴリズムを実行時間で比較した．遅延評価法はベースラインと比較して実行時間を大幅に減らすことができた．また，本研究では人工データのみで実験を行ったため，実データを用いた実験を行うことが今後の課題である．

謝辞

本研究は科研費基盤研究 (C)18K11311 の助成を受けたものである．

参考文献

[1] R. Zhu, B. Wang, X. Yang, B. Zheng, and G. Wang, “SAP: Improving Continuous Top-K Queries Over Streaming Data,” IEEE Trans. Knowl. DataEng., vol.29(6), pp.1310-1328, June 2017.

[2] X. Xu, C. Gao, J. Pei, K. Wang, and A. Al-Barakati, “Continuous similarity search for evolving queries,” Knowledge and Information Systems, vol.48(3), pp.649-678, September 2016.

[3] C. Efstathiades, A. Belesiotis, D. Skoutas, and D. Pfoser. ”Similarity Search on Spatio-Textual Point Sets,” EDBT, pages 329-340, 2016.