

時間帯や重要人物の都合を考慮した空き時間検索システムの開発

上河 竣哉^{†1} 大坪 竜誠^{†1} 成 凱^{†1}

概要: 近年, Google カレンダーのような便利なオンラインサービスを利用して PC やスマートフォン上でスケジュールを管理している人が多く見受けられる. 家族間や職場でスケジュールを共有することで, 共同作業を進めやすく仕事の効率が上がる利点がある. 打合せや勉強会などのような複数の人間が関わる事象を計画する際に, 全員の都合の良い時間帯を見つける必要がある本稿では, 時制データベースにおける時制集約演算の一つとして, **共通空き時間計数問題と重み付き空き時間計数問題に着目し**, その数学基礎とアルゴリズムを考案した. 時区間結合, 時区間短縮, により, 集約の処理時間を改善する方式を提案した. 提案手法を評価し, 実験データの生成方法と予備実験の結果を報告する.

キーワード: 時制データベース, 時制データの集約, 時区間集合, 集合演算, 空き時間検索

Temporal Aggregation of Null Time with VIP and Period Preferences

SHUNYA KAMIKAWA^{†1} RYUSEI OTSUBO^{†1} KAI CHENG^{†1}

Abstract: In this paper, we developed efficient algorithms for null time aggregation in large temporal databases, especially from event database where a series of events associated with time intervals. A null time interval is a time interval where no any events associated with it. Null times are assumed for scheduling new events from a large number of members. We propose α coalescible and interval shrink to handle this issue and preliminary experiments showed that our approach is feasible.

Keywords: Free time aggregation, temporal database, temporal aggregation, time interval set

1. はじめに

ビジネスにおいて社員同士や顧客との打ち合わせで, スケジュールを確認して日程を決めることが日常的に行われている. また, ビジネスだけでなくプライベートでの飲み会や旅行など計画を立てることもよくある. 近年, 手帳やメモ帳などの紙媒体に代わり, Google カレンダーのような便利なオンラインサービスを利用して PC やスマートフォン上でスケジュールを管理している人が多く見受けられる. 用件名, 開始時刻, 終了時刻, 場所等を含む予定を簡単に登録でき, スケジュールを立てやすくなっている. 目的に応じて複数のカレンダーを作成し使い分けことができ, 家族間や職場でスケジュールを共有することで, 共同作業を進めやすく仕事の効率が上がる利点がある. 打合せや勉強会などのような複数の人間が関わる事象を計画する際に, 全員の都合の良い時間帯を見つける必要がある. Google カレンダーではカレンダー共有を前提に画面上で全員が空いている時間を目視で確認でき会議の設定ができる. しかし, 非公開や共有されていない予定がある場合, 参加者の予定を個別に確認する必要がある, 担当者の負担が大きいと思われる. また, 実際に予定を立てるとき, 優先度の高い重要人物や時間帯を考慮しないとイケない場合がある.

我々は時制データベースにおける空き時間検索機能を

開発している[7]. 時制データとは, ある事象について時間的制約が存在し, その事象の開始と終了の時間や時間の継続を保持しているようなデータである. 電子カルテ, ライフログ, 施設予約, スケジュール管理などの応用分野では時制データが特に重要である. 空き時間検索は時制データベースにおける時区間を軸とする集約演算は時制集約(temporal aggregation)の一つであり, 多くの研究が存在している[2]. 時制集約のために, 集約対象のグループ化が重要である. 時制データのグループ化に固定長区間(fixed intervals)または不変区間(constant intervals)が用いられる. 固定長区間の場合は, 時間軸を一定の長さに固定してグループ化する. 長さは通常は時間単位とする. 例えば, 年間, 月間, 週間などである. 集約は各グループに対して行う. 一方, 不変区間では, 集約結果が不変である最大の時区間をグループ化の基準とする. 固定長区間は, 区間の長さを適切に決めるのが難しい問題, 不変区間は時区間の分割が細かすぎて集約結果が元データより大きくなる問題が知られている.

[7]では, 空き時間検索に特化した時制集約問題に着目し, 対象者の予定のある時区間を集約してその結果から全員共通の空き時間を求める方式を提案し, データベースの集約機能を活かした効率化を図った. しかし, これまでの研究ではまだいくつかの問題点が残された. (1) 集約アルゴリ

^{†1} 九州産業大学
Kyushu Sangyo University.

ズムの効率性が低い。集約する対象は予定のある時区
間であり、集約後にさらに空き時間を求めないといけない
手間が残っている。(2) 時間帯や重要人物の都合を考慮す
る必要のある場合に、既存の手法は簡単に対応できない。
これらの問題を解決するために、本研究では、空き時間を
対象とする集約手法を開発し、時間帯や重要人物の都合を
考慮し、効率的な空き時間検索アルゴリズムを提案する。

2. 問題定義

本稿では、時制データベースとは時間的制約を伴う一連
の事象に関するデータを集めて保管しているものとする。
データベースに登録された時制データから、一定の条件を
満たすデータを抽出し集計することで空き時間検索を実現
する。

空き時間検索

開始日時:
2017-01-01 14:57

終了日時:
2017-02-05 14:57

空き時間:
100

● 分 ○ 時間

重要人物指定:
 阿部太郎 後藤太郎 伊藤四郎 松香太郎 佐藤健一 高橋花子

検索
取消

図 1 空き時間検索

図 1 では、空き時間を検索するユーザインターフェース
が示されている。複数のメンバーのある時間範囲において、
重要人物の指定とともに、共通の空き時間の長さ以上の空
き時間を求めるように条件を設定できる。

(1) 実時区間 vs 空時区間

各事象に時区間 (time interval) がついており、その事象
が有効である期間を示す。時区間は時間定義域上の連続な
区間である。 $E = \{e_1, e_2, \dots, e_n\}$ を事象識別子の集合とする
とき、 $(e_i, s_i, f_i) \in E \times N \times N$ を事象時区間 (または単に「事
象時間」) という。 N が時間定義域 (自然数集合) であり、
 s_i と f_i はそれぞれ事象の開始時刻と終了時刻である。時区
間には事象の存在を表す「実時区間 (real interval)」, 事象の
休止を表す「空時区間 (null interval)」に区別する[6]。

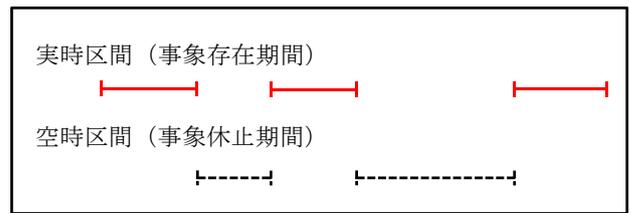


図 2 実時区間と空時区間

実時区間と空時区間の関係は図 2 に示されている。実時
区間から空時区間を求めることが簡単である。本論文では、
特に説明しない限り、事象識別子を意識する必要がないの
で、事象時区間と時区間は区別せず使う。また、同一事象
の時区間が互いに交わりを持たないとする。つまり、同一
事象は同じ時間帯に複数の予定を入れない。同一人物はある
時間帯に予定有無だけ重要なので、この仮定は本研究の
結論に影響しない。

(2) 共通空き時間

共通空き時間とは複数の事象の間に共通する空時区間で
ある。時制データベースより一定の条件を満たす共通空き
時間を検索する問題は、**空き時間検索問題**である。例えば、
図 11 時間帯の打合せを行おうとしてメンバー10 人全員
参加できる1 時間半以上の空時区間を見つけるのは、共通
空き時間検索の例である。全員参加できる時間が見つから
ない場合を考慮して、共通空き時間の定義を以下のように
修正する。

共通空き時間計数問題: 長さ α 以上の空き時間を共通す
る事象の最大数を求める。例えば、メンバー10 人のうち、
9:30~12:00 の間の1 時間半以上の空き時間を共通するメ
ンバーが最大8 名であるとき、この時間帯において9 名以
上のメンバーで1 時間半以上の共通空き時間が存在しない。

重み付き空き時間計数問題: 各空き時間の重要度を区別
するために重みを付けた場合は、空き時間を共通する事象
の数の代わりに、重みの合計で評価する空き時間検索問題
は重み付き空き時間計数問題である。時間帯や重要人物の
都合を考慮する場合は、適切な重みを付けることで検索が
できる。

3. 時区間集合上の演算

3.1 時区間結合

接続している時区間が**結合可能**(coalescible)な時区間と
いい、連続発生する事象を意味する。図 3(a) (左側) に示
すように、時区間 A と B が接続している (A meets B) か
ら、A, B 結合後一つの時区間になる。時区間が厳密に接続
していないが、かなり近い (距離 α 以内) ならば、結合可
能とみなすとき、 α **結合可能**(α coalescible)という。図 3(b)
(右側) は α 結合可能の例である。

時区間結合(temporal coalescing)[5]は、時制データベースに断片化された時区間を処理し、時制データベースの最適化技術として用いられている。本稿では、これまでと違い、 α 結合可能を導入し、不必要な処理コストを削減し、時制集約の効率化を目的としている。

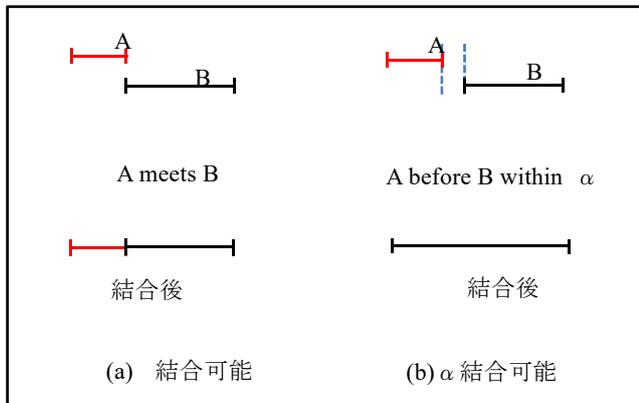


図 3 時区間結合

3.2 時区間短縮

時区間結合と合わせて存在するのが、時区間短縮である。時区間短縮とは時区間の両端より一定の長さを縮めることである。時区間の α 結合可能に対応して、時区間の長さ α 以上であるとき、 α 短縮可能と定義する。時区間 $[s, f]$ の α 短縮は以下の式で行う。

短縮前： $[s, f]$

短縮後： $[s + (1 - \lambda)\alpha, f - \lambda\alpha]$, ($0 \leq \lambda \leq 1$)

図 4 は α 短縮のイメージを示している。元の時区間が矢印の方向に左端より $(1 - \lambda)\alpha$ 、右端より $\lambda\alpha$ だけを短縮している。短縮された部分は $(1 - \lambda)\alpha + \lambda\alpha = \alpha$ である。

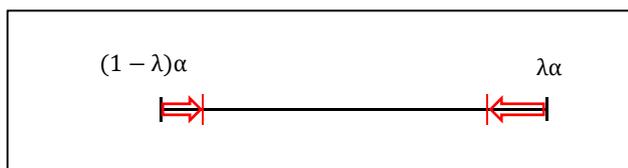


図 4 時区間短縮（両端長さ合計 α 分を縮める）

時区間短縮により、長さ未満の空時区間をあらかじめ集約対象から排除し、余計な計算コストを省くことができる。それだけでなく、後で述べるように、空き時間集約の簡素化にもつながる。

3.3 時区間の集合演算

時区間を要素とする集合に対して、二つの時区間の和集合と差集合を求める演算（和演算 $A+B$ 、差演算 $A-B$ ）は表 1 に示している。時区間の時間関係により、結果が違う。(a), (f)には A と B の重なりがないが、(b)~(e)では A と B

の一部が重なっている。(c)では、時区間 A が B により分断され、二つの区間になっている。和演算は事象データベースから空きではない時区間を求めるために利用される。差演算は、実時区間より空時区間を求めるに使われる。

表 1 $A = [a_s, a_f]$ と $B = [b_s, b_f]$ の集合演算

| 時制関係 | $A + B$ | $A - B$ |
|--------------|--------------|--------------------------|
| A before B | A, B | A |
| A overlaps B | $[a_s, b_f]$ | $[a_s, b_s]$ |
| A contains B | A | $[a_s, b_s], [b_f, a_f]$ |
| A starts B | B | \emptyset |
| A finishes B | B | \emptyset |
| A meets B | $[a_s, b_f]$ | $[a_s, a_f]$ |
| A equals B | A | \emptyset |

4. 空き時間の評価

問合せ区間 $Q = [p, q]$ に長さ α 以上の空時区間を求めるために、問合せ時区間と交差するすべての事象の空時区間をデータベースより取り出し結果を集約する。データベースに実時区間と空時区間両方を検索できると仮定する。実時区間しか保存されていない場合、図 2 に示すように空時区間を求められる。

4.1 問合せ区間における空時区間抽出

空時区間 $X = [s, f]$ が問合せ区間 $Q = [p, q]$ との共通部分で長さ α 以上の時区間であるために、以下のいずれかの条件を満たす必要がある。

- (1) X 前半の長さ α 以上の部分が Q の範囲内にある。

$$p + \alpha \leq s \leq q + \alpha$$

- (2) X 後半の長さ α 以上の部分が Q の範囲内にある。

$$p + \alpha \leq f \leq q + \alpha$$

上記の条件を一般化すると以下ようになる。

$$[s + (1 - \lambda)\alpha, f - \lambda\alpha] \cap [p, q] \neq \emptyset, \quad (1)$$

λ は $[0, 1]$ 以内の任意の実数である。例えば、

$$\lambda = 1 \text{ のとき, } [s, f - \alpha] \cap [p, q] \neq \emptyset, \quad (2)$$

$$\lambda = 0 \text{ のとき, } [s + \alpha, f] \cap [p, q] \neq \emptyset, \quad (3)$$

つまり、対象の空時区間を全て同じ方法で長さ α だけ短縮すると、 α 以上である状況を考慮せず集約でき、空き時間集約の計算を簡素化できる。これは、事象の実時区間から見ると、これは実時区間を長さ α だけ伸ばす結果であり、延長により実時区間が接続できるようになる。このとき、元の実時区間が α 結合可能でなければならない。

4.2 時区間のグループと集約

前述の抽出方法で、データベースから長さ α 以上の空き時区間を抽出し開始時刻の早い順で並べ替えているとする。図 5 は検索結果の例を示している。

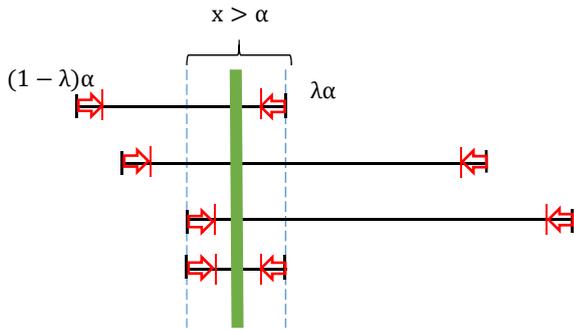


図 5 時区間短縮による検索の簡略化

定理 1 : 短縮前の時区間集合に長さ α 以上の共通部分を持つ必要十分な条件は、短縮後の時区間集合は空ではないことである

証明 : 短縮前の時区間集合に長さ α 以上の共通部分を持つとき、短縮によりその共通部分から減らされた部分が α 以下なので、結果として、短縮後の時区間集合は空ではない。

逆に、短縮後の時区間集合は空ではないとき、短縮前の時区間は共通部分を中心に最低長さ α がプラスするので、元の時区間集合が長さ α 以上の共通部分を持つ。□

α 短縮の空き時間を入力として、以下のように共通空き時間を集約する。 ω_j に j 番目の共通空き時間を格納し、 Ω_j には対象時区間を格納する。集約アルゴリズムは図 6 に示している。

(共通空き時間計数問題)

- 1) . (初期化) $i=0; j=0; \Omega_j = \emptyset, \omega_j = [\bar{s}_j, \bar{f}_j] = [0, +\infty]$
- 2) . (走査) $i = i + 1; I_i = [s_i, f_i]$
- 3) . (ω_j と I_i の共通部分を求める)
 $s = \text{MAX}(\bar{s}_j, s_i), s = \text{MIN}(\bar{f}_j, f_i)$
- 4) . もし $s \leq f$ ならば (共通部分が空ではない場合)
 $\omega_j = [s, f], \Omega_j = \Omega_j \cup I_i$
 ステップ 2) へ続く
- 5) . もし $s > f$ であれば (共通部分が空である場合)
 $j = j + 1, \omega_j = [\bar{s}_j, \bar{f}_j] = [0, +\infty], \Omega_j = \Omega_j \cup I_i$
 ステップ 2) へ続く
- 6) . ω_j と $|\Omega_j|$ を出力する

図 6 時区間集約アルゴリズム

まず、ステップ 3) で処理中の時区間 I_i と現在の途中結果である ω_j との共通部分 $[s, f]$ を求める。この共通部分が空でなければ、 I_i を現在のグループに残して処理を続ける。しか

し、もし共通部分 $[s, f]$ が空であれば、現在グループを閉じて、新しいグループに I_i を入れる。繰り返し終了後、各共通空時区間 ω_j とそれぞれの数 $|\Omega_j|$ を出力する。図 7 は空き時間検索結果の例である。

重み付き共通空き時間計数問題は同様に対応できる。出力は $|\Omega_j|$ の代わりに、 Ω_j に含まれる時区間の重みの合計になる。

空き時間検索結果

| 開始日時 | 終了日時 | 対象者 | 評価結果 |
|----------------|----------------|-------------------------------|------|
| 1月02日 (月)02:00 | 1月03日 (火)10:00 | 後藤太郎 伊藤四郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 8 |
| 1月05日 (木)14:30 | 1月09日 (月)09:00 | 後藤太郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 7 |
| 1月12日 (木)08:00 | 1月12日 (木)12:00 | 伊藤四郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 7 |
| 1月13日 (金)20:00 | 1月16日 (月)13:00 | 阿部太郎 松香太郎 佐藤健一 高橋花子 | 6 |
| 1月17日 (火)15:00 | 1月19日 (木)12:30 | 伊藤四郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 7 |
| 1月21日 (土)21:30 | 1月23日 (月)09:00 | 伊藤四郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 7 |
| 1月23日 (月)10:40 | 1月28日 (土)06:00 | 阿部太郎 松香太郎 佐藤健一 高橋花子 | 6 |
| 1月29日 (日)21:30 | 1月31日 (火)14:00 | 伊藤四郎 阿部太郎 松香太郎 佐藤健一 高橋花子 | 7 |

図 7 空き時間検索結果

5. 実験評価

実験用データとして、事象 ID、開始時刻、終了時刻、重みをもってシミュレーションデータを生成し、時制データベースに登録した。時刻は $0 \sim 10^5$ の整数として、長さ 500 以内の乱数で時区間を 13,068 件生成した。

実験は α 短縮を使わない一般の集約アルゴリズムと α 短縮に基づくアルゴリズムを比較した。結果は図 8 に示すように、集約処理時間が改善できたことが確認できた。

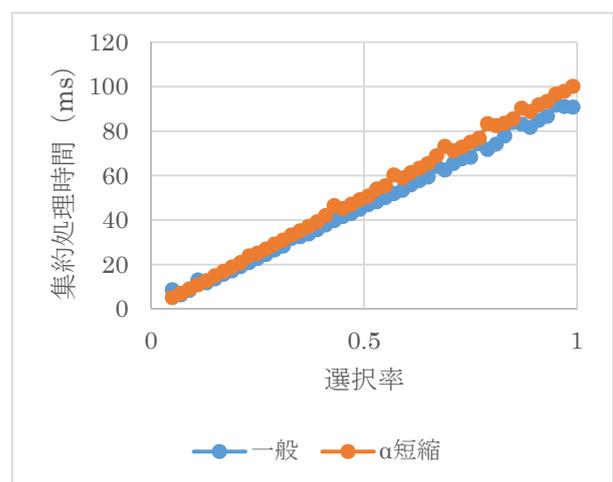


図 8 α 短縮による集計時間の改善

6. おわりに

本稿では時間帯や重要人物の都合を考慮した空き時間検索システムの開発について報告した。時区間結合、時区間短縮により集約処理効率の改善を図った。その結果、空き時間の検索を成功し、一定の効率性があることを確認した。

参考文献

- [1] J. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of ACM*, vol.26, issue 11, pp.832-843, 1983.
- [2] Lopez, Ines Fernando Vega, Richard T. Snodgrass, and Bongki Moon. Spatiotemporal aggregate computation: A survey. *IEEE Transactions on Knowledge and Data Engineering*, Vol.17 No.2 (2005): 271-286.
- [3] Chen, Yi-Cheng, et al. An efficient algorithm for mining time interval-based patterns in large database. *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM 2010)*. pp. 49-58, 2010.
- [4] Yang, J., and Widom, J. Incremental computation and maintenance of temporal aggregates. In *Proceedings of the 17th International Conference on Data Engineering (ICDE 2001)*. (pp. 51-60). IEEE.2001.
- [5] Zhou, Xin, Fusheng Wang, and Carlo Zaniolo. Efficient temporal coalescing query support in relational database systems. *International Conference on Database and Expert Systems Applications*. Springer Berlin Heidelberg, 2006.
- [6] 天笠 俊之, 田頭 利規, 金森 吉成, 増永 良文. 制約を導入した時区間代数. 電子情報通信学会技術研究報告. DE, データ工学 95(148), 41-48, 1995-07-19
- [7] 田中涼幸, 浦博喜, 成 凱. 時制データベースにおける空き時間検索機能の開発, 火の国情報シンポジウム 2016, 平成 28 年 3 月, 宮崎大学