社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＰ－69（46）
2007／12／21

任意個数の時系列に含まれる類似部分探索法

菅井 康祐†, 杉山 雅英†

†会津大学 コンピュータ理工学部
〒965-8580 会津若松市一箕町
E-Mail: †{s1120118, sugiyama}@u-aizu.ac.jp

あらまし 本報告では任意個数の時系列に含まれる類似部分を探索する方法として RDDS-n 法の新しい実現法を提案する．この新しい実現法を用いれば時系列の数を任意に与えられるので，時系列の数に合わせて個別に実装する必要がなくなる．決められた時系列数で実装した RDDS-3 法と同程度の処理速度を実現できた．この新しい実現法を用いて，時系列の数が 3, 4, 5 において実験し評価しこの実現法の有効性を示す．

キーワード: 類似部分探索法, RDDS-n

A New Implementation of Similar Segment Search in An Arbitrary Number of Time-Series

K. Sugai†, M. Sugiyama†

†Graduate School of Computer Science and Engineering, The Univ. of Aizu
Ikki-machi, Aizu-Wakamatsu, Fukushima, 965-8580 Japan
E-Mail: †{s1120118, sugiyama}@u-aizu.ac.jp

Abstract This paper describes a new implementation of RDDS-n to solve a similar segment search in an arbitrary number of time-series. This new implementation can process arbitrary time-series, so each implementation considering the number of time-series does not need. This new proposed implementation achieves the same level of processing time against RDDS-3 which depends on number of time-series. The new implementation are evaluated on 3, 4, 5 time-series.

Keyword: Similar Segment Search, RDDS-n

1 まえがき

近年のデータの大容量化により，より速く目的のデータを探索する技術が求められてきた．現在までにクエリ探索として Active 探索法 [1](AS 法と略す) がよく知られている．その後，2 つの時系列における類似部分探索法として RIFAS 法 [2] が提案された．さらにそれを発展させて 3 つ以上の時系列に対する類似部分探索法として RDDS 法 [3] が提案された．RDDS 法において，時系列数が 2 つ [3], 3 つ [4] の場合は既に実現された．RDDS 法を一般化し，任意の時系列数で実現されたのが RDDS-n[5] である．文献 [5] では 3 つの時系列に対して動作する RDDS-3[4] に対し，約 2 倍処理速度がかかっていたが，本報告では新しい実装法を提案し同程度の処理時間を得られたことを報告する．

本報告ではまず複数時系列中の類似セグメント探索問題と RDDS 法の原理を説明し，RDDS-n 法の実現法，処理時間の向上した実現法を述べ，実験評価を行う．

2 複数時系列中の類似セグメント探索

音声やビデオなどの特徴ベクトルの時系列 $\boldsymbol{v}_t$ $(t = 0,\ldots,T)$ を M 次元ユークリッド空間の非負の単位球 $\boldsymbol{U}_1 = \{\boldsymbol{x} = x_m |\ x_m \geq 0, \|\boldsymbol{x}\|_1 = 1\}$ のベクトル $\boldsymbol{r}_t$ に変換する．$\|\boldsymbol{x}\|_1$ は ℓ_1 ノルムであり，L 時刻分の $\boldsymbol{r}_t$ の相加平均を $\boldsymbol{p}_t$ とする．

$$\boldsymbol{p}_t = \frac{1}{L}\sum_{l=0}^{L-1} \boldsymbol{r}_{l+t}. \tag{1}$$

$\boldsymbol{U}_1$ は凸であるので $\boldsymbol{p}_t \in \boldsymbol{U}_1$ である．特徴ベクトルの時系列 $(\boldsymbol{v}_t)$ を $\boldsymbol{U}_1$ の時系列 $\boldsymbol{P} = (\boldsymbol{p}_t)$ に変換し，変換で得られた $\boldsymbol{U}_1$ の複数の時系列 $(\boldsymbol{p}_t^{(i)})$ を $\boldsymbol{P}^{(i)}$ $(i \in [1,n])$ と

表す．$\boldsymbol{P}^{(i)}$ の時刻を t_i で，i,j 番目の時系列の時刻 t_i, t_j のベクトル $\boldsymbol{p}_{t_i}^{(i)}, \boldsymbol{p}_{t_j}^{(j)}$ 間の距離を $d_1(t_i,t_j)$ と表わすことにする．時刻の組 $\boldsymbol{t}=(t_1,t_2,\ldots,t_n)$ における時系列の距離を式 (2) で定義する．

$$d(\boldsymbol{t})=\sum_{i<j} d_1(t_i,t_j). \tag{2}$$

d の対称性から $i<j$ の組に限定している．組の数は $n(n-1)/2$ で $d_1(t_i,t_j)\leq 2$ なので $d(\boldsymbol{t})\leq n(n-1)$ である．ここで $d(\boldsymbol{t})$ の値が小さいことは n 個の時系列が時刻の組 $\boldsymbol{t}=(t_1,t_2,\ldots,t_n)$ において類似のセグメントを含むことを意味している．従って $\theta(\geq 0)$ を探索閾値とすると式 (3) を満たす時刻の組 $\boldsymbol{t}$ の探索は時系列に含まれる類似セグメントの探索に意味している．

$$d(\boldsymbol{t})\leq\theta. \tag{3}$$

ここで探索閾値 θ の値域は $d(\boldsymbol{t})$ の値域と同一であるので $\theta\leq n(n-1)$ となる．時刻ベクトルに対して $\boldsymbol{t}+\boldsymbol{s}=(t_i+s_i)$ とすると，$|d(\boldsymbol{t}+\boldsymbol{s})-d(\boldsymbol{t})|$ は不等式 (4) を満たす．

$$|d(\boldsymbol{t}+\boldsymbol{s})-d(\boldsymbol{t})|\leq\frac{2(n-1)}{L}\|\boldsymbol{s}\|_1. \tag{4}$$

ここで $\|\boldsymbol{s}\|_1=\sum_i|s_i|$ で定義される ℓ_1 ノルムである．式 (4) より式 (5) を満たす $\boldsymbol{s}$ に対して

$$\|\boldsymbol{s}\|_1<\frac{L}{2(n-1)}|\theta-d(\boldsymbol{t})|, \tag{5}$$

$d(\boldsymbol{t})>\theta$ であれば $d(\boldsymbol{t}+\boldsymbol{s})>\theta$ であり，$d(\boldsymbol{t})<\theta$ であれば $d(\boldsymbol{t}+\boldsymbol{s})<\theta$ となることを示せる．

式 (5) の右辺をスキップ半径 R と呼ぶことにする．多次元時刻 $\boldsymbol{t}$ を考えることでクエリ探索において導き出された AS 法におけるスキップ幅は時刻空間におけるノルムで半径に拡張される．式 (5) を満たす時刻ベクトル $\boldsymbol{s}$ は ℓ_1 ノルムの球で n 次元空間の超菱形領域（1 次元では区間，2 次元では菱形，3 次元では正 8 面体）となる．$\theta, d(\boldsymbol{t})$ の値域からスキップ半径は式 (6) を満たす．

$$R\leq\frac{nL}{2}. \tag{6}$$

探索時は時刻 $\boldsymbol{t}$ での距離 $d(\boldsymbol{t})$ を計算し式 (3) を判定する．$d(\boldsymbol{t})\leq\theta$ であれば $W(\leq R)$ を半径とする ℓ_1 球 $B_1(\boldsymbol{t},W)$ 内の全ての時刻 $\boldsymbol{t}+\boldsymbol{s}$ が式 (3) を満たす．一方，$d(\boldsymbol{t})>\theta$ であれば球 $B_1(\boldsymbol{t},W)$ の内部の全ての時刻は条件を満たさないとして枝刈できる．

2.1 再帰的菱形分割探索法 (RDDS)

複数時系列類似セグメント探索問題の解法である再帰的菱形分割探索法 (RDDS: Recursive Diamond Division Search) を述べる．時刻領域 $\boldsymbol{T}=[0,T_1)\times[0,T_2)\times\ldots[0,T_n)$ を同一の半径 W の ℓ_1 球[1]で被覆する．

$$\boldsymbol{T}\subset\cup_i B_1(\boldsymbol{t}_i,W). \tag{7}$$

球の中心 $\boldsymbol{t}_i$ での距離 $d(\boldsymbol{t}_i)$ を求めれば式 (5) を用いて球 $B_1(\boldsymbol{t}_i,W)$ が検出領域，枝刈領域，判定不可能のいずれかになる．判定不可能の場合には半径 1/2 の球に分割し球の中心で距離を再度求め判定を行う．分割の限界を $W_{\min}$ とし，これを変化させることで探索処理時間及び精度を制御可能であり，$W_{\min}=0$ にすると，全数探索と同精度で探索可能である．高次元で式 (7) を満たす ℓ_1 球を ℓ_∞ 球 (超立方体) との関係 (8) を利用して作成する．

$$B_\infty(\boldsymbol{t}_i,r)\subset B_1(\boldsymbol{t}_i,nr). \tag{8}$$

3 RDDS-n

3.1 RDDS-2, RDDS-3 と RDDS-n の違い

RDDS-2, RDDS-3 は与えられる時系列数が既知であり，2 つ及び 3 つの時系列に対して動作するが，RDDS-n は実行時に時系列数を得ることで，任意の個数の時系列で動作することができる．時系列の数に合わせた実装法では汎用性がない．RDDS-3 と RDDS-n の実現法の違いを以下に述べる．

RDDS-3 は以下の様に 3 つの時系列の時刻を指定するための 3 重ループで実現している．

RDDS-3 の実現法

```
loop z in file3 {
 loop y in file2 {
  loop x in file1 {
   RDDS-3(x, y, z, r);
  } } }
RDDS-3( x, y, z, r ){
  if( d(x,y,z) > theta1 ) reject;
  else if( d(x,y,z) < theta2 ) detect;
  else RDDS-3(x',y',z',r/2) /* 8個 */
}
```

一方 RDDS-n は無限ループで実現する．

[1] n 次元空間の ℓ_1 球は正 2^n 面体で頂点数 $2n$，正 n 面体 2^n 個であり，ℓ_∞ 球は正 $2n$ 面体で頂点数 2^n，正 $2n-2$ 面体 $2n$ 個で構成され，互いに双対である [6]．

RDDS-n の実現法

```
loop (全ての中心座標 t を生成) {
  t = ( t1, t2, ... tn )
  RDDS-n( t, r );
}
RDDS-n( t, r ){
  if( d(t) > theta1 ) reject;
  else if( d(t) < theta2 ) detect;
  else RDDS-n( t', r/2 ) /* 2^n 個 */
}
d(t){
  時系列の t = ( t1, t2, ... tn ) に
  対応する距離
}
```

RDDS-3 と RDDS-n の違いは探索処理部のループの実装方法にある. RDDS-3 は既に時系列の個数が 3 と決められているので, 時系列の時刻 $t = (t_1, t_2, t_3)$ を指定するための 3 重ループを用いる. 一方, RDDS-n は RDDS-n を起動させる時に指定される時系列のファイル名と同時に時系列の個数 n を得る. RDDS-n は n 重ループとなる. n 重のループを前もって実装できないので, 無限ループを用いて n 重ループの動作を得ることにする. 第 3.2 項で RDDS-n の中心座標の生成法を説明し, 第 3.3 項で超立方体の分割法について述べる.

3.2 初期中心座標の生成法

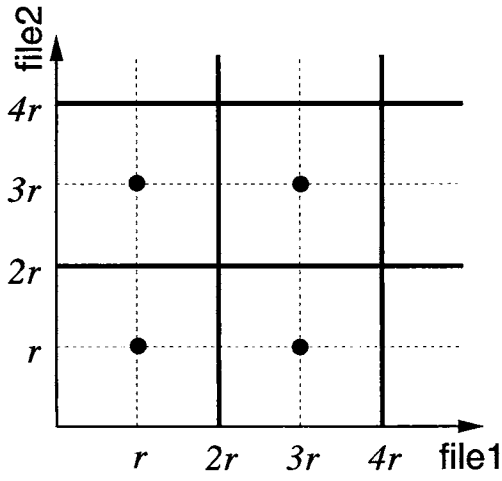


図 1: 超立方体の中心座標.

超菱形 (超立方体) の中心座標を図 1 の黒丸とする. 中心座標の一般形 t を以下に示す.

$$t = (t_1, t_2, \ldots, t_n),$$
$$t_i = (2k_i + 1) \times r, (k_i = 0, 1, \ldots, u_i). \tag{9}$$

t_i は i 番目の時系列の時刻である. r を半径, k_i を座標位置を移動させるカウンタとする. ここで T_i を i 番目の時系列長とすると u_i は式 (10) で計算される.

$$u_i = \lfloor \frac{T_i}{2r} \rfloor - 1. \tag{10}$$

n 個の時刻座標各々をループで制御する代りに一つの無限ループを用いて以下の手順で実現する.

step 0 $k_1 = k_2 = \ldots = k_n = 0$

step 1 k_1 を 1 づつカウントアップ

step 2 k_1 が上限 u_1 に達したら

- k_2 を 1 カウントアップ
- k_1 を 0 に初期化, goto step 1

step 3 k_2 が上限 u_2 まで達したら

- k_3 を 1 カウントアップ
- k_1, k_2 を 0 に初期化し, goto step 1

この手順で全ての超立方体の中心座標を生成配置できる. 時系列の長さ T_i は時系列が与えられた時点で確定できる.

3.3 超立方体の分割法

初期超立方体の中心で求めた距離 $d(t)$ が判定条件を満たさない場合には超立方体の分割を行い, その新しい中心座標と半径を用いて再帰処理を行う. 図 2 の矢印で示された丸が新しい超立方体の中心座標を示すようにその新中心座標は元の超立方体の中心 t と半径 r から求められる. n 個の時系列なので分割で生成する超立方体の個数は 2^n となる. それらの中心座標は現在の中心座標 t に 2^n 個の変位ベクトルを加えることで得られる. ここで変位ベクトルは n 次元空間の $(\pm 1, \pm 1, \ldots, \pm 1)$ の 2^n の組み合せの j 番目の方向ベクトル v_j に半径 $r/2$ をかけて得られる. 従って座標 $t + (r/2)v_j$ $(j = 1, 2, \ldots, 2^n)$ を中心とする 2^n 個の RDDS の再帰を呼び出すことになる. ここで n 次元空間の方向ベクトル v_j は時系列の個数が与えられた時点で一度だけ作成する. その方法として, v_j は 2^n 個なので 2 進数の組み合せを用いて作成する.

超立方体の半径 r が偶数の時は分割によって不足が生じることはないが r が奇数の時は不足領域が生じるので半径の偶数化, もしくは全数探索を行う必要がある [7]. 従って分割を進め $r = 1$ の時にさらに分割する際には全数探索を行うこととする.

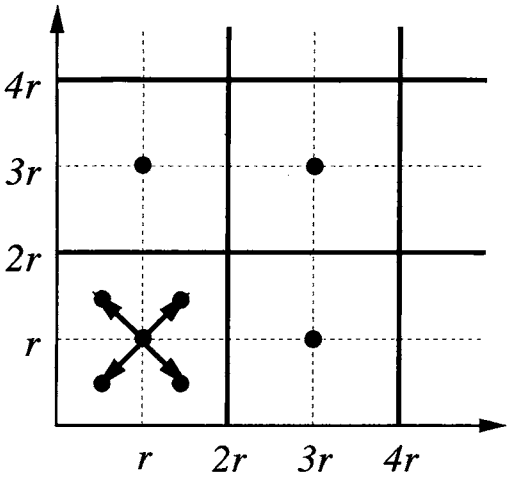


図 2: 新しい超立方体の中心座標.

3.4 2つの時刻の組み合せの表現法

RDDS-n の実装方法を 2 つ提案する. 第 1 の方法は文献 [5] で述べた方法であり, 第 2 の方法はそれを改良した方法である. 方法 1 と方法 2 の違いは, 距離計算部分の実装方法である. RDDS-n の探索処理部で最も時間がかかる処理を gprof で分析した. 結果, RDDS-3 では距離計算部のみで約 285 秒かかっているのに対し, 方法 1 では約 407 秒かかっていた. 従って距離計算部に着目して検討した. 以下に改善の方法を示す.

方法 1 では, 距離計算で用いる時系列の組み合せ

$$(t_1,t_2),(t_1,t_3),\ldots,(t_i,t_j),\ldots,(t_{n-1},t_n),(i\le j).$$

をその都度ループを用いて生成していた. 一方, 方法 2 では前処理として時系列の組み合せを生成しておく. 実際の処理の違いを RDDS-3 を用いて説明する.

d を組み合せの合計の距離, d_1 を出現ベクトル間の距離として,

$$d(\boldsymbol{t})=\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} d_1(\boldsymbol{p}_{t_i}^{(i)},\boldsymbol{p}_{t_j}^{(j)}).$$

RDDS-3 は以下のように 3 通りの組み合せを指定して距離計算を行う.

RDDS-3 の距離の計算方法

```
d=d_1(p[0][x],p[1][y])
 +d_1(p[0][x],p[2][z])
 +d_1(p[1][y],p[2][z])
```

`p[i][x]` は i 番目の時系列の時刻 $\boldsymbol{x}$ の出現ベクトルを意味している. 一方, 方法 1 では 2 重ループを用いて組み合せを導き出していた.

方法 1 の距離の計算方法

```
loop i (0..n-1)
  loop j (i+1..n-1)
    d += d_1(p[i][t[i]],p[j][t[j]])
```

ここで `p[i][t[i]]` は i 番目の時系列の時刻 t_i の時刻ベクトル $\boldsymbol{p}_{t_i}^{(i)}$. 上の 3 つの時系列の場合には (i,j) の組み合せは $(0,1)$, $(0,2)$, $(1,2)$ の 3 通りであり, 2 重ループを用いなくともこれを生成することが可能である. この距離計算の部分を方法 2 では以下の様に変更した.

方法 2 の距離の計算方法

```
loop k (0..n*(n-1)/2-1)
  d += d_1(p[tau[k].i][t[tau[k].i]],
           p[tau[k].j][t[tau[k].j]])
```

ここで tau は $n(n-1)/2$ 通りの組み合せを格納しておく構造体であり tau は事前に生成しておく. 詳しく述べれば以下の様に組み合せを格納する構造体 $\boldsymbol{\tau}_i$ の配列を定義し時系列の組み合せを格納する.

$$\boldsymbol{\tau}_0,\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_i,\ldots,\boldsymbol{\tau}_{\frac{n(n-1)}{2}-1},$$

$$\boldsymbol{\tau}_0=(0,1),\boldsymbol{\tau}_1=(0,2),\ldots,\boldsymbol{\tau}_{\frac{n(n-1)}{2}-1}=(n-1,n).$$

この様にすることで方法 2 は 1 重ループのみを用いて距離計算部を実現できる. これは $n=3$ の場合, RDDS-3 の距離計算と同じ処理をしていることになる. 方法 1, 2 の処理時間の違いを実験により評価する.

4 RDDS-n の評価

4.1 実験条件

評価実験に CampusWave データベース [8] の第 1 回から第 5 回までの収録データを用いた. これは会津若松市内の FM 局の音楽リクエスト番組であり, 2 名の女性パーソナリティの対話音声, リクエスト曲, CM 音声などを含んでおり, 音声長は各々約 1 時間である. 分析条件を表 1 に示す. n を時系列数として, $\theta=0.3,0.6,1.0(=n(n-1)/2\times 0.1)$, $W=768,1024,1280(=n\times 2^8, n=3,4,5)$, $L=625$, 最小半径を $W_{\min}=0$ とした. ここで第 2.1 項の式 (8) より, W を超菱形の半径とすると $\frac{W}{n}=r=2^8$ は超立方体の半径である.

4.2 実験結果

4.2.1 探索処理時間の比較

図 3-(a), (b), (c) は時系列の個数 $n=3,4,5$ での探索処理時間の比較である. 最大の時系列長を 30 分までとした.

表 1: 音響分析及び実験条件.

標本化周波数	16kHz
窓長	256 点 (16ms)
フレーム更新周期	256 点 (16ms)
窓関数	ハミング窓
高域強調	$(1-0.97\,z^{-1})$
LPC 分析	14
LPC ケプストラム分析	16
音声長	約 1 時間 (約 $T = 225,000$)
セグメント長	10 秒 ($L = 625$ フレーム)
VQ 符号帳サイズ	$M = 32$

ただし $n = 5$ においては時間がかかりすぎてしまうので時系列長を 480 秒までとした. 図 3-(a), (b), (c) をみると, どの時系列数においても方法 2 は方法 1 より約 2 倍速く動作していることがみてとれる. 従って第 3.4 項で述べた 2 重ループの除去が処理時間の向上につることが判る. 方法 1 と方法 2 の処理時間の比を図 4 に示す. この図から, 音声長によらず方法 2 が方法 1 より処理時間が 2.5 倍程度速いことがわかる. また $n = 3$ の場合に比べて, $n = 4, 5$ の方が改善効果が大きいことが判る.

gprof による分析においても, 距離計算部分で方法 1 では約 407 秒かかっているのに対し, 方法 2 では約 291 秒と改善された.

また, 方法 2-(a) から判るように RDDS-3 法と同程度の処理時間となった. これから, RDDS-n 法が任意の個数の時系列に対して動作でき, それぞれの時系列数に合わせて実装する必要がなく, 個別に実装する場合と同一以上の処理時間を実現できたことを示している.

4.2.2 探索処理時間の推定

RDDS-n の初期の超立方体の個数は, T を時系列長, r を半径, n を時系列数とすると,

$$(T/2r)^n \tag{11}$$

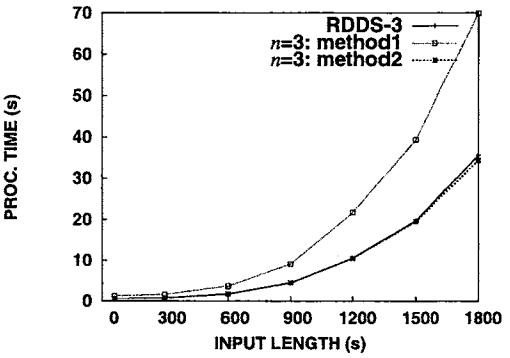
となる. ここで探索処理時間は距離計算回数で決定されるとすると, 各立方体での平均距離計算回数を a_n とし, a_n が時系列長に関係ないとすると, 距離計算回数は

$$a_n(T/2r)^n \tag{12}$$

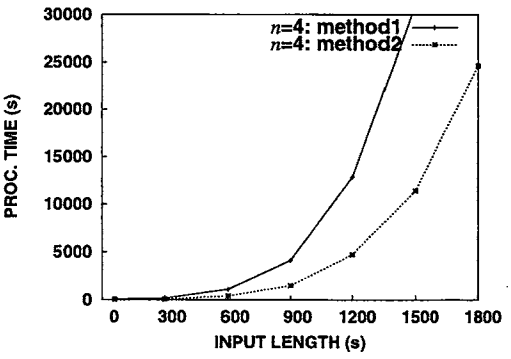
となり, 探索処理時間は

$$y = b_n T^n, \quad (b_n = \frac{a_n}{(2r)^n}) \tag{13}$$

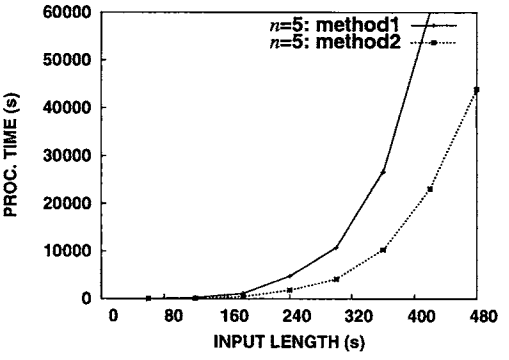
となることが判る. 式 (13) より, b_n の値を最小 2 乗法で予測することにより平均距離計算回数 a_n が判る. 図 5 から,



(a) $n = 3$



(b) $n = 4$



(c) $n = 5$

図 3: 方法 1 と方法 2 の探索処理時間の比較.

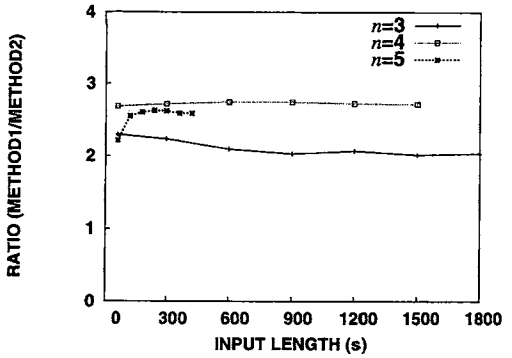


図 4: 方法 1 と方法 2 の探索処理時間の比 ($n = 3, 4, 5$).

$n = 3$ の時の最小 2 乗推定値が実験で得られた処理時間と同程度であることがわかる. 表 2 に, 方法 2 の $n = 3, 4, 5$ の b_n とデータの長さを 30 分とした推定処理時間を示す. $n = 5$ の場合には約 1 年かかると推定される. 従って現時点では, 時系列の個数を増加させると現実的な処理時間での動作は得られないことになる. b_n の値は n が大きくなるに連れて半減している. 平均距離計算回数 a_{n+1} を式 (15) で予測できる.

$$\frac{a_{n+1}}{a_n} = \frac{(2r)^{n+1}b_{n+1}}{(2r)^n b_n} = (2r)\frac{b_{n+1}}{b_n} \approx (2r)\frac{1}{2} = r, \quad (14)$$

$$a_{n+1} \approx r a_n. \quad (15)$$

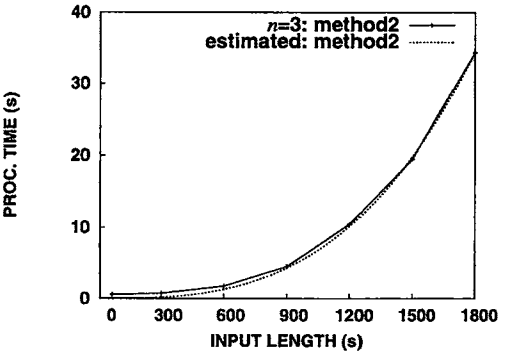


図 5: 方法 2 の $n = 3$ の処理時間と予測される関数.

5 むすび

本報告では RDDS-n 法の時系列に依存しない実現法について述べ, その探索処理時間を評価した. 実現法を 2 つ提案し, 方法 1 と方法 2 を比較し, 方法 2 の処理時間が従来の実装法である RDDS-3 と同程度以上であることを示し, RDDS-n の有効性を示した. 方法 2 では距離計算で用いる時系列の組み合せを事前に生成しておくことでループを削減し, 処理時間を向上させた.

表 2: 時系列の個数と 30 分の音声に対する予測処理時間の関係.

時系列数 (n)	方法 2	
	b_n	予測処理時間 (s)
3	5.881×10^{-9}	34.30
4	2.328×10^{-9}	24443.65
5	1.732×10^{-9}	3.27×10^7

時系列数の増加に伴い, 処理時間が爆発的に増加するので, より探索の効率化, 高速化が必要である. 今後は, 探索出力のクラスタリング [9], RDDS-n 法の 2 つの時系列への帰着法について検討していく.

謝辞 日頃有益な助言, 討論いただくヒューマンインターフェース学講座の皆様に感謝致します.

参考文献

[1] 柏野, 他, ヒストグラム特徴系列に基づく長時間音響信号の高速探索, 音学講論, 2-9-24, pp.561-562 (1998-09).

[2] 西村, 他, アクティブ探索法による時系列データ中の一致区間検出 —参照区間自由時系列アクティブ探索法—, 電子情報通信学会論文誌, D-II, Vol.J84-D-II, No.8, pp.1826-1837 (2001-08).

[3] 杉山, 複数時系列中の類似セグメント探索法の提案と評価, 音声研資 SP2005-196, pp.71-76 (2006-03).

[4] 酒井, 杉山, 複数時系列中の類似セグメント高速探索法 — 3 つの時系列に対する実装と評価 —, 音声研資, SP2007-09 (2007-06).

[5] 菅井, 杉山, 任意個数の時系列に含まれる類似部分探索法, 音学講論, 1-3-13 (2007-09).

[6] 数学辞典, 岩波書店.

[7] 杉山, 類似セグメント高速探索法における球被覆の検討, 音学講論, 1-P-22 (2007-03).

[8] 内田, 杉山, CampusWave 音声データベースの作成, 電気関係学会東北支部連合大会, 2A-6 (2000-08).

[9] 杉山, RDDS 探索出力のクラスタリング法, 音学講論 (2008-03). (発表予定)