タイムワーピングに基づく時系列データの類似検索：次元縮小による効率化

Efficient Search of Similar Time Series under Time Warping with Dimensionality Reduction

大桃 諭♥ 陳 漢雄♦
古瀬 一隆♦ 大保 信夫♠

Satoshi OOMOMO Hanxiong CHEN
Kazutaka FURUSE Nobuo OHBO

時系列データの類似度にユークリッド距離が使用されていたが，時間軸方向のずれに柔軟性を持たせたタイムワーピング距離が近年注目をあびている．しかし，タイムワーピング距離の計算時間は非常に長く，しかも距離公理の三角不等式が成り立たないため，伝統的な索引技法の適用は困難である．そこで本研究では，時系列データに対する新しい次元縮小法と下界関数を提案する．そして，この下界関数をフィルタリングに使用したときの検索効率が，関連研究で提案されている下界関数によるものよりも優れていることを実験によって実証する．

Recently, similarity between time series data measured by time warping (TW) distance which allows out of phase in the time axis has got a lot of attention in recent years. However, since TW violates the triangle inequality, traditional indexing techniques is powerless. To overcome this problem, in this paper we propose a new lower bound function based on dimensionality reduction. This enables efficient indexing hence filtering in similarity search of time series data. We show by experimental results that our lower bound function outperforms other ones proposed in related works.

1. はじめに

時系列データとは，時刻iの時の値をv_iとすると，$V= v_1, v_2 \cdots, v_N$で表される実数のシーケンスである．時系列データの例としては，株価や為替レート，心電図，気温・湿度変化などがあり，ここに挙げた以外にも多く存在する．このように科学，医療，経済，工学などの様々な分野で時系列データが扱われている．さらに近年では，扱うデータ量も非常に多くなってきている．このような現状において，時系列データの効率的な類似検索の必要性が高まっている．例えば，最近の株価の動向と類似したものを過去のデータから見つけ出して，今後の動向を予測するといったことが挙げられる．

♥ 学生会員 筑波大学大学院システム情報工学研究科
oomomo@dblab.is.tsukuba.ac.jp
♦ 正会員 筑波大学大学院システム情報工学研究科
{chx, furuse}@cs.tsukuba.ac.jp
♠ 筑波大学大学院システム情報工学研究科
ohbo@cs.tsukuba.ac.jp

類似検索を行うためには，2つの時系列間の類似度を定義する必要がある．現在，時系列の類似度として最もよく使用されているものはユークリッド距離である．しかしユークリッド距離には，2つの時系列の長さが等しくなければならない，時間軸方向のわずかなずれに対しても大きな影響を受けてしまうなどの問題がある．そこで，ユークリッド距離に代わるものとしてタイムワーピング距離が使用されている．

類似検索に要する計算コストは，次の2つの要因に大きく依存する．1つ目は時系列の長さである．ユークリッド距離は時系列の長さに比例したCPUコストで済むが，タイムワーピング距離は長さの2乗に比例したCPUコストを要する．この問題に対処するために使用される手法の1つが次元縮小である．離散フーリエ変換(DFT)，離散ウェーブレット変換(DWT)，Piecewise Aggregate Approximation (PAA)，Adaptive Piecewise Constant Approximation (APCA) などが使用されている．2つ目はデータベースに含まれる時系列の数である．時系列の数が増えることによって，CPUコストだけではなくI/Oコストも大きく増加する．そこで，R-treeなどのインデックスを使用する手法が提案されている．

しかしこれらの手法の中には，false dismissalを生じてしまうものがある．false dismissalとは，本来検索結果として返されるべき解が返されないことを意味する．このfalse dismissalを生じさせないで計算コストを小さくする手法として使用されているものが，下界関数を使用したフィルタリングである[2]~[4]．

本研究では，タイムワーピング距離を使用して類似検索を行う場合における効率的な検索手法を提案する．提案手法では，与えられた時系列に対してその時系列を包含する領域ボックス列というものを作成する．そして，この領域ボックス列を使用して下界関数を定義する．この下界関数をフィルタリングに使用することで，効率的な検索を可能にする．

2. タイムワーピング

この章では，タイムワーピング距離の計算方法について簡単に述べる([1])．

長さn,mの2つの時系列$Q=q_1,\cdots,q_i,\cdots,q_n$, $C=c_1,\cdots,c_j,\cdots,c_m$に対して，まずは，以下の式で定義される2つの点$q_i$,$c_j$間の距離$d(q_i,c_j)$を(i,j)要素の値とするn×m行列を作成する．

$$d(q_i,c_j)=(q_i-c_j)^2 \qquad (1)$$

次に，ワーピングパス$W=W_1,W_2,\cdots,W_k,\cdots,W_K$を求める．ワーピングパスは，以下の3つの条件を満たす行列の要素の順列で表現される．ワーピングパスの例を図1に示す。

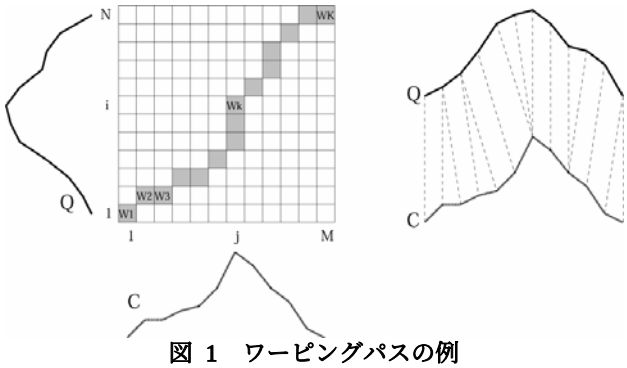

図 1 ワーピングパスの例

Fig.1 An example of warping path

- ☑ **[境界条件]** $w_1=(1,1), w_k=(n,m)$とする．
- ☑ **[連続性]** $w_k=(a,b), w_{k-1}=(a',b')$とすると，$a-a' \leqq 1$ かつ$b-b' \leqq 1$となる．
- ☑ **[単調性]** $w_k=(a,b), w_{k-1}=(a',b')$とすると，$a-a' \geqq 0$ かつ$b-b' \geqq 0$となる．

上記の3つの条件の下で得られるワーピングパスは非常に多く存在するが，それぞれのワーピングパスに対して(2)式で示すワーピングコストと呼ばれるものが存在する．

$$C(W)=\sqrt{\sum_{k=1}^{K} w_k} \qquad (2)$$

この式から得られるワーピングコストの中で最小のものがタイムワーピング距離になる．

$$D_{tw}(Q, C) = \min\{C(W)\} \qquad (3)$$

ワーピングパスは非常に多く存在するので，全てのワーピングパスに対してワーピングコストを計算して最小値を求めるのは現実的ではない．そこで，以下の再帰関数を使用してタイムワーピング距離を計算する．

$$\gamma(i, j) = d(q_i, c_j) + \min \begin{Bmatrix} \gamma(i-1, j-1) \\ \gamma(i-1, j) \\ \gamma(i, j-1) \end{Bmatrix} \qquad (4)$$

$$\gamma(0,0)=0, \gamma(i,0)=\gamma(0,j)=\infty \qquad (5)$$

この再帰関数を使用すると，式(3)は以下のように表される．

$$D_{tw}(Q, C) = \sqrt{\gamma(N, M)} \qquad (6)$$

これは動的プログラミングによってO(NM)で計算可能である．

3. 関連研究

2つの時系列Q, Cに対して本来の距離をD(Q,C)とすると，以下の式を満たす$D_{lb}(Q,C)$が下界関数である．

$$D_{lb}(Q, C) \leq D(Q, C)$$

この下界関数をフィルタリングに使用することで，効率的な検索を行うことができる．例えば，問い合わせQとの距離がe以下の時系列を検索する場合，データベースから取り出した時系列Cに対して下界距離を計算し，eより小さい場合にのみ，本来の距離D(Q,C)を計算すればよい．

また，下界関数には性能の善し悪しがある．1つは下界距離自身の計算コストである．最低でも本来の距離の計算コストよりも小さくなければ，下界関数をフィルタリングに使用することは意味を成さない．計算コストが小さくなるほどフィルタリングに要する時間が短くなるので，結果として検索時間も短くなる．もう1つは下界関数の精度，すなわち下界距離と本来の距離との差である．極端な例であるが，0も当然下界距離になれる。しかし，eがどんなに小さな正の値であっても，結局はD(Q,C)を計算することになってしまい，フィルタリング効果がまったくない．下界距離がD(Q,C)に近いほどフィルタリング効果が大きくなり，検索時間が短くなる．

Yiらは[5]において以下のような下界関数を定義している．第1章で述べたように，タイムワーピング距離の計算では一方の時系列の点は他方の時系列のいずれかの点と必ず対応させなくてはならない．このことを利用して，一方の時系列の点の中で他方の時系列の最大の点より大きいもの，もしくは最小の点より小さいものを見つけて，それらの点の値の差を用いて下界関数を定義している．図で表すと，図2の点線で表される距離の和が下界距離になる．この手法はCPUコストがO(N+M)で済むが，データベースから全ての時系列を取り出す必要があるのでI/Oコストは減少しない．

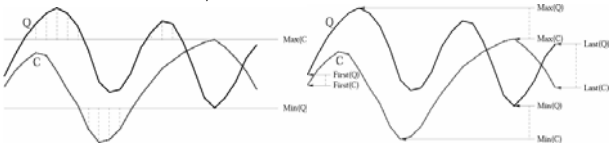

図 2 Yiによる下界関数（左）とKimによる下界関数（右）

Fig.2 Lower bound by Yi (left) and Kim (right), respectively

Kimらも[4]においてそれぞれの時系列から4次元の特徴ベクトルを抽出する．特徴ベクトルは時系列の最初の値First(C)，最後の値Last(C)，最大値Max(C)，最小値Min(C)から成る．そして2つの特徴ベクトルが与えられたとき，それぞれの要素の値の差の中で最大のものが下界距離になる．図2では最後の値の差Last(Q)-Last(C)が下界距離となる．この特徴ベクトルを4次元空間上に配置してR-treeなどのインデックスを構築することで，計算コストを減少させている．この手法はデータベースが大きい場合には有効であるが，下界関数自身の精度はよくない．

Keoghらは[2][3]において時系列を固定長な領域ボックスに区切りMBRを用いる下界関数を提案した．この方法は下界関数の精度をあげた一方，事前に領域ボックスの数を決めなければならないため規則性のない時系列データには無駄が生じる．

4. 提案手法

我々は時系列データにあわせた可変長な領域ボックスを提案し，それを用いてKeoghの下界関数([3])を改良する．つぎにまず，ある時系列$Q=q_1, \cdots, q_n$が与えられたときに，その時系列を包含する領域ボックス列$R^Q=r^Q_1, \cdots, r^Q_N$を作成する方法を述べる．

4.1 領域ボックス列の作成

まずは，時系列Qを重なりのない複数の部分時系列$Q=<q_1, \cdots, q_h>, <q_{h+1}, \cdots, q_k>, \cdots, <q_{k+1}, \cdots, q_N>$,に分割する．そして，それぞれの部分時系列に対して1つの領域ボックスを作成する．1つの領域ボックスは上限U，下限L，サイズSの3つの要素から成り，部分時系列$<q_{h+1}, \cdots, q_k>$に対して作成された領域ボックスをr^Q_iとすると，以下のように定義される．

$r^Q_i.U=max\{q_{h+1}, \cdots, q_k\}$, $r^Q_i.L=min\{q_{h+1}, \cdots, q_k\}$, $r^Q_i.S=k-h$,

領域ボックス列の作成において中心となる処理は部分時系列への分割である．そして，この分割方法が検索効率に大きな影響を与える．影響を与える1つの要因は領域ボックスの数N'であり，もう1つは領域ボックスの面積$\sum_{i=1}^{N'}(r_i.U - r_i.L)\, r_i.S$である．下界距離の計算方法については次の章で詳しく説明するが，数が少なくなるほど下界距離の計算コストは小さくなり，面積が小さくなるほど下界距離によるフィルタリング効果は上がるが，数と面積はトレードオフの関係にある．そこで，本研究では図3のアルゴリズムを用いて領域ボックス列を生成する．

1. Input: 許容振幅率 $\rho \in (0,1]$
2. $\tau = \rho \cdot (\max(Q)-\min(Q))$
3. $i=1, n=1$
4. find minimum $j (i \leqq j < N)$ such that
   $\max\{q_i, \cdots, q_j\} - \min\{q_i, \cdots, q_j\} \leq \tau$

$\max\{q_i,\cdots,q_{j+1}\} - \min\{q_i,\cdots,q_{j+1}\} > \tau$

5. **if** j dose not exist **then** $j=N$
6. $r_i^q.U = \max\{q_i,\cdots,q_j\}$, $r_i^q.L = \min\{q_i,\cdots,q_j\}$
   $r_i^q.S = j - i + 1$
   if j==N **return** R^Q, **otherwise goto** 4

図 3 領域ボックス列作成アルゴリズム

Fig.3 Region box series creation algorithm

このアルゴリズムに従って，τ=8 として領域ボックス列を作成する様子を図 4 に示す．点 A から点 B までの部分時系列の最大値と最小値の差は 7 であり，τ 以下の値となっているが，点 B の次の点である点 C まで進むと差は 10 になり，閾値より大きくなる．よって，点 A から点 B までが 1 つの領域ボックスとなり，点 C が次の領域ボックスの開始点になる．

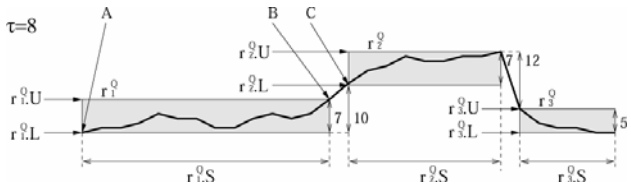

図 4 領域ボックス列の例

Fig.4 An example of region box series.

閾値 τ の値によって領域ボックスの数や面積は決まるが，時系列の取り得る値の範囲が事前にわかっていない限り，閾値 τ を直接指定することは困難である．そこで，許容振幅率ρというものをパラメータとし，このρから閾値 τ を計算する（図 3 の 2 行目）．ρは時系列の取り得る値の範囲に関係なく，0 から 1 の値で指定することができる．ρが小さいほど領域ボックスの面積は小さくなるが数は多くなる．面積と数はトレードオフの関係にあり，より効率的な検索を行うためにはユーザがρに適切な値を指定する必要がある．

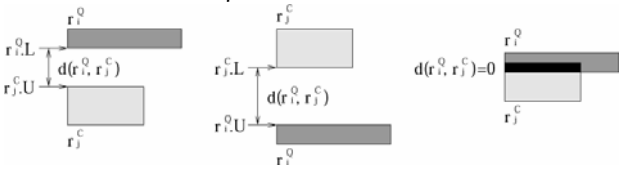

図 5 領域ボックス間の距離

Fig.5 Distance between region boxes.

4.2 改良下界関数および検索アルゴリズム

2 つの時系列 $Q=q_1,\cdots,q_i,\cdots,q_N$, $C=c_1,\cdots,c_j,\cdots,c_M$ に対して，それぞれの領域ボックス列 $R^Q=r^Q_1,\cdots,r^Q_{N'}$, $R^C=r^C_1,\cdots,r^C_{M'}$ が作成されたとする．この 2 つの領域ボックス列を使用して，タイムワーピング距離 $D_{tw}(Q,C)$ の下界関数 $D_{tw\_lb}(R^Q, R^C)$ を定義する．本文は領域ボックスが可変長であるため，本質的には異なるが，表現上の便宜から[3] と同じ形式の計算式を利用する．まずは式(1)で表される 2 点間の距離の代わりとなる，図 5 で示すような 2 つの領域ボックス r^Q_i, r^C_j 間の距離 $d'(r^Q_i, r^C_j)$ を定義する．

次に再帰関数(4)，(5)をもとに，下界関数のための新たな再帰関数 γ'_d を定義する．

$$d'(r_i^Q, r_j^C) = \begin{cases} (r_i^Q.L - r_j^C.U)^2, & if \quad r_i^Q.L - r_j^C.U \\ (r_i^C.L - r_j^Q.U)^2, & if \quad r_i^C.L - r_j^Q.U \\ 0, & otherwise \end{cases}$$

$$\gamma'_r(i,j) = \min\begin{cases} \gamma'_d(i-1,j-1) + d'(r_i^Q, r_j^C) \bullet r_j^C.S \\ \gamma'_c(i-1,j) + d'(r_i^Q, r_j^C) \\ \gamma'_r(i,j-1) + d'(r_i^Q, r_j^C) \bullet r_j^C.S \end{cases}$$

$$\gamma'_c(i,j) = \min\begin{cases} \gamma'_d(i-1,j-1) + d'(r_i^Q, r_j^C) \bullet r_i^C.S \\ \gamma'_c(i-1,j) + d'(r_i^Q, r_j^C) \bullet r_i^C.S \\ \gamma'_r(i,j-1) + d'(r_i^Q, r_j^C) \end{cases}$$

$$\gamma'_d(i,j) = \max\begin{cases} \gamma'_r(i,j) \\ \gamma'_c(i,j) \end{cases}$$

$$\gamma'_d(0,0) = 0, \quad \gamma'_r(i,0) = \gamma'_c(0,j) = \infty$$

この再帰関数を使用して下界関数を以下のように定義する．

$$D_{tw\_lb}(R^Q, R^C) = \sqrt{\gamma'_d(N', M')}$$

これは動的プログラミングによって O(N'M')で計算可能である．紙面の関係で証明を省略するが、このように定義された関数は以下の式を満たし，よって下界として機能する．

$$D_{tw\_lb}(R^Q, R^C) \le D_{tw\_lb}(Q, C)$$

この下界関数を用いて類似検索に対するフィルタリングが可能である．類似検索には大きく分けると近傍検索と範囲検索の 2 種類がある．ここでは k=1 のときの最近傍検索アルゴリズムを説明するが，k>1 の場合の近傍検索や範囲検索もアルゴリズムを少し変更することで行うことができる．

実際に検索を行う前に，データベース中の全ての時系列に対して領域ボックス列を作成してファイルに格納しておく．このファイルを圧縮ファイルと呼ぶことにする．問い合わせ時系列 Q が与えられると，次のアルゴリズムを用いて最近傍検索を行う．

1．Qから領域ボックス列 R^Q を作成する．

2．圧縮ファイルから領域ボックス列 R^C を 1 つ取り出して下界距離 $D_{tw\_lb}(R^Q, R^C)$ を計算する．

3．下界距離がこれまでの最小距離より小さい場合には，データベースから時系列を取り出して本来の距離 $D_{tw}(Q,C)$ を計算する．これも最小距離より小さい場合には最小距離を更新する．

4．圧縮ファイルから次の領域ボックス列を取り出して同様の処理を行う．これを全ての時系列に対して行って，最終的に最小距離となった時系列を検索結果として返す．

5. 実験

今回の実験では株価データセットと $v_t=v_{t-1}+r_t$（r_t は[-1,1]の乱数)から生成される RandomWalk 合成データセットの 2 種類を使用する．合成データセットは長さが 256，512，1024 の RandomWalk データをそれぞれ 100，1000，10000 個生成して作成することで，合計 9 つの合成データセットを作成した．株価データセットは S&P 500 (http://biz.swcp.com/stocks/)から 479 個の銘柄の始値，高値，安値，終値の 4 種類のデータ 252 日分（データ長が 252，データ数が 479×4）

からなる．問い合わせデータは，データセット中のデータと同じ長さのRandomWalkデータを新たに生成して使用する．また，それぞれのデータに対して平均値が 0，標準偏差が 1 になるように正規化を行う．

表 1　フィルタリング率の比較

Table 1　Comparison on filtering rate

dataset	Synthetic dataset			stock
Len. data	256	512	1024	252
Num. data	10000	10000	10000	1916
Serial	0	0	0	0
Yi's	17.7	25.0	23.9	10.6
Kim's	36.8	26.5	15.9	24.4
RB (ρ)	30.2 (0.20)	32.3 (0.20)	40.1 (0.20)	37.3 (0.20)

表 2　ディスクアクセス数の比較

Table 2　Comparison on disk access

dataset	Synthetic dataset			stock
Len. data	256	512	1024	252
Num. data	10000	10000	10000	1916
Serial	20040	40040	80040	3780
Yi's	20040	40040	80040	3780
Kim's	19581	37286	76233	4400
RB (ρ)	11505 (0.13)	16313 (0.13)	26402 (0.09)	2102 (0.13)

表 3　実行時間の比較

Table 3　Comparison on execution time

dataset	Synthetic dataset			stock
Len. data	256	512	1024	252
Num. data	10000	10000	10000	1916
Serial	35.6	136.4	543.9	6.3
Yi's	29.6	105.3	419.9	5.7
Kim's	22.9	102.7	459.9	4.8
RB (ρ)	6.26 (0.10)	17.31 (0.09)	49.31 (0.06)	1.16 (0.09)

本研究で提案した領域ボックスを使用した下界関数をフィルタリングに使用する RB 検索手法と 3 つの手法の検索効率を比較する．すなわち，下界関数でのフィルタリングを行わずに，直接タイムワーピング距離を計算する逐次検索手法，第 3 章で述べた Yi らや Kim らの下界関数によるフィルタリング検索手法である．領域ボックス列を作成するときに指定する許容振幅率 ρ には，0.01〜0.20 の 0.01 刻みの値を使用する．50 個の異なった問い合わせデータで最近傍検索を行って平均結果を計測し，表１〜３でその一部の結果を示す。

まずフィルタリング率（表１）について比較した．この値が大きいほど実際のタイムワーピング距離の計算が減り，検索効率が良いと判断できる．RB 検索手法では許容振幅率を大きくするとフィルタリング率が小さくなっている．これは，許容振幅率を大きくすると領域ボックスの面積が大きくなるので，下界距離が小さくなってフィルタリング効果が下がるためと考えられる．しかし，許容振幅率を 0.2 まで大きくしてもフィルタリング率は 30〜40%であり，Yi や Kim の 10〜40%に比べて高いことがわかる．

つぎにディスクアクセス数（表２）について比較した．いうまでもなく，この値が小さい検索効率が良いと判断できる．RB 検索手法では，圧縮ファイルへのアクセスとデータベースへのアクセスのバランスが取れた箇所でディスクアクセス数が最小になる．ほかの方法より顕著に減少した．

最後に検索効率を総合的に評価するため問い合わせを与えてから検索結果が得られるまでの実行時間（表３）についても比較した．データの数や長さにもよるが，Yi や Kim では最大でも逐次検索手法の 65%程度までしか実行時間が短くならないが，RB 検索手法では 30%〜10%程度まで実行時間が短くなっている．

6. おわりに

本研究では，時系列から可変長領域ボックス列を作成して Keogh の下界関数を新しく定義し，false dismissal を生じることなく効率的な類似検索を行うことを試みた．最近傍検索の実験によって，提案手法が関連研究の手法よりも優れた性能を持っていることが実証された．今後の課題としては，今回は時系列を領域ボックス列に次元縮小して圧縮ファイルを作成しただけなので，R-tree などのインデックスを構築して下界距離の計算回数をさらに減らすことが考えられる．

[文献]

[1] J. B. Kruskall and M. Liberman, ``The Symmetric Time-Warping Problem: From Continuous to Discrete,"
Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, pp.125-161, Addison-Wesley, 1983

[2] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani, ``Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," In Proc. of 2001 ACM SIGMOD, 2001.

[3] E. Keogh, ``Exact Indexing of Dynamic Time Warping" In Proc, VLDB, 2002., pp. 406-417.

[4] S.-W. Kim, S. Park, and W. W. Chu, ``An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases," In Proc. of 17th ICDE, 2001.

[5] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, ``Efficient Retrieval of Similar Time Sequences under Time Warping," In Proc. of 14th ICDE, 1998.

大桃 論　Satoshi OOMOMO
2003 年筑波大学情報学類卒業．現在，同大大学院博士課程システム情報工学研究科在学中．データベースシステムに関する研究に従事．日本データベース学会学生会員．

陳 漢雄　Hanxiong CHEN
1993 年筑波大学大学院博士課程工学研究科了．現在，同大電子・情報工学系講師．データベースシステムに関する研究に従事．工博．日本データベース学会正会員．

古瀬 一隆　Kazutaka FURUSE
1993 年筑波大学大学院博士課程工学研究科了．現在，同大電子・情報工学系講師．データベースシステムに関する研究に従事．工博．日本データベース学会正会員．

大保 信夫　Nobuo OHBO
1968 年東京大学理学部卒．1970 年同大大学院修士課程了．同年同大理学部助手．1980 年筑波大学電子・情報工学系講師．1995 年同大電子・情報工学系教授．現在に至る．データベースシステムに関する研究に従事．理博．