多次元的なログデータマイニングを実現するデータキューブ機構の提案

A New Datacube System Supporting Multi-Dimensional Log-Data Mining

成瀬 正英[1] 大森 匡[2] 星 守[3]

Masahide NARUSE Tadashi OHMORI
Mamoru HOSHI

本稿では, 計算システムや Web サーバのログ分析を目的として, アイテムセットキューブと呼ぶデータキューブ機構を述べる. この機構は, OLAP で用いられている多次元的な分析手法の枠組に沿ってログデータ集合からのデータマイニング (高頻度アイテムセット計算) を行うものである. 本稿では, アイテムセットキューブの概念, 及び, 数値データキューブと併せたログ分析手法を述べ, 実データへの適用結果を示す.

Recently there is much need of understanding logs which are generated by computer systems or web servers. To solve this need, this paper proposes a new datacube system named Itemset Cube. This is used to understand logs by OLAP-style multi-dimensional data mining. Definition and algorithms of the itemset cube and its applications on a real dataset are described.

1. はじめに

近年, 情報システムや Web サーバから生成される膨大なログを分析しシステム内でどのような現象が生じているかを理解したいという要求が高まっている. 通常, ログデータは,「いつ, どこで, 誰によって, どのような事象の組が生じたのか」を表す多属性データである. 従来の分析手法では, データキューブ (数値キューブ) を用いて多属性データを多次元空間に分類集計しておき任意の次元の組についての多様な集計問い合わせに高速に応える手法 (OLAP)[6] や, 調べたい状況におけるログデータ集合から高頻度アイテムセット計算を使って頻出事象の組を探すアイテムセット分析 [4][5] がある.

本稿では, 分析者が数値データキューブを使って着目すべき状況 (すなわち注目すべき属性の組やそこでの制約条件) を探す際に, それと併せて, その状況で生じている高頻度アイテムセットも計算できるようなデータマイニングサーバを提案する. 具体的には, 高頻度アイテムセットを値として持つデータキューブ機構を提案する. そして, 分析者が着目する状況を指定した時に, このキューブを変形して, その状況において生じている現象 (高頻度アイテムセット) を計算する. このように, OLAP で成功してきた多次元分析手法をデータマイニングについても行うデータベースサーバが本研究の目的である. 本稿では, アイテムセットキューブと呼ぶ本機構の概念, 用法, 実データへの適用結果を述べる.

[1]学生会員, 電気通信大学, naruse@hol.is.uec.ac.jp
[2]非会員, 電気通信大学, omori@hol.is.uec.ac.jp
[3]非会員, 電気通信大学

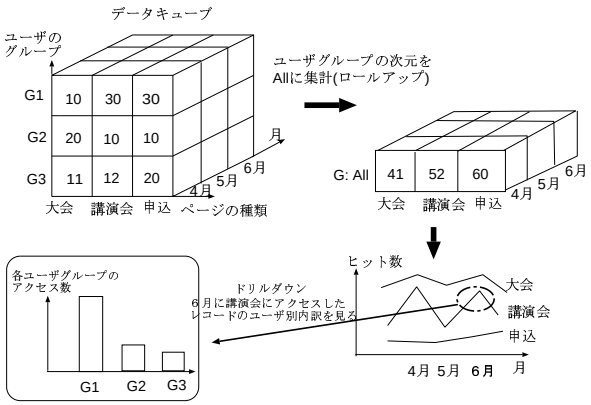

図 1: 数値データキューブによる分析
Fig.1 Numeric analysis by OLAP

2. アイテムセットキューブによる多次元ログ分析

2.1 準備

本稿では Web サーバのアクセスログ分析を例にとってアイテムセットキューブを概説する.

アクセスログ分析では, ログ列は「同一ユーザから一定時間内に連続してアクセスされたページの列」を表すセッションレコードに変形される. 以下, 本稿では, セッションレコードを, [ユーザドメイン, アクセス開始時刻, アクセスの発生した月, アクセスされたページ集合] の 4 属性で表現する. そして, ユーザドメイン (の種類), アクセスの生じた月 (1 月, 2 月, ･･･), アクセスされたページ集合, の 3 属性に着目したログ分析を考える. 以下, レコードという時は, セッションレコードを指す.

次に, 各属性を 1 次元として考えて多次元空間を作り, 各次元をいくつかの区切りに分割してデータキューブ構造を作ってレコードを分類する.

例として,「ユーザドメイン」属性を, ドメインの種類 (ac.jp,co.jp, .com 等) によって $\{G_1, G_2, G_3\}$ の 3 つに分割する. 1 レコードは相異なる G_i, G_j に同時に属すことはない. このようにレコードを排他的に分ける分割 (この例では $\{G_i\}(i = 1, 2, 3)$) のことを, **排他的な分割**と呼ぶ. 各値 G_i を, 分割における**区切り**と呼ぶ.「月」属性は, 1 月から 12 月に分割する.

一方, Web ページは, 学会などのコミュニティサイトであれば, 大会 (E_1) や講演会 (E_2), 会員申込み (E_3) などの行事に関するページに分類される. ページ p が E_i に分類される時, E_i を p のページ種類と呼ぶ. これに従って言うと,「(1 レコードでアクセスされた) ページ集合」属性は, 述語「ページ種類 E_i に属すページが当該レコードでアクセスされている」(i=1,2,3) で分割される. (この述語自体も以下では記号 E_i で参照する). ここで, 1 レコードは複数のページをアクセスするから, 結局, 1 レコードは相異なる E_i, E_j に属す可能性がある. このような分割を**非排他的な分割** と呼ぶ. この時, 概念的には, 当該レコードは照応する各 E_i へコピーされて分類されると考える.

2.2 従来のログ分析手法の問題点

2.2.1 数値データキューブによる技法について

ログ分析を多次元的に行う手法の主流は, データキューブを使った OLAP である [6]. 図 1 左上に,「月」,「ユーザドメイン」,「ページ集合」の 3 次元からなる数値データキューブを示す. キューブの各次元 X は, それに与えられた分割 $P_x =$

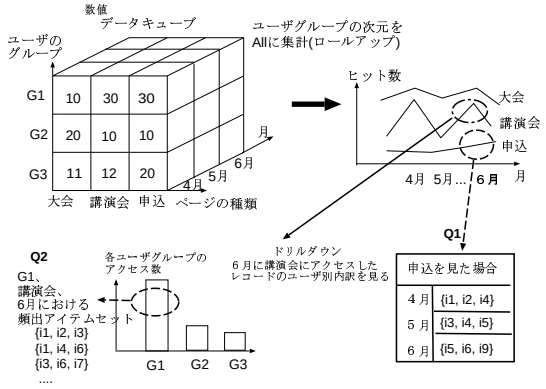

図 2: 数値データキューブと併せたアイテムセット計算
Fig.2 Itemset analysis with OLAP

$\{x_1, x_2, \ldots, x_{k_x}\}$ により区切られる．キューブの最小構成要素であるセルは，各次元 X から区切り 1 つ (これを x_i で表す) を選んだ時に，これら x_i の論理積を全次元について行った論理式として定義される．各セルは，その論理式を満たすレコードの総数 (ヒット数) を格納する．

OLAP では，このようなデータキューブをあらかじめ計算しておき，利用者の問い合わせに対して，該当するセルの値を集計して返す．問い合わせは，図 1 の「ページ種類別 E_1, E_2, E_3 のヒット数を月単位で求めよ」というように，必ずしもデータキューブの構造とは一致しない．この場合，図中，不要な次元である「ユーザドメイン」を全ユーザ (図中の G:All) へロールアップし，元の数値データキューブを変形することで，結果を得ることができる．この処理自体は単純な集計処理である．得られた結果をグラフ表示すると図 1 右下になる．このとき，講演会ページへのヒット数が 6 月に急増していることに着目すると，そこでのユーザ種類別のアクセス数内訳を求めることが考えられる．これがドリルダウン処理であり，元のキューブからスライス，ロールアップにより計算する. 図 1 左下が得られる結果である．

2.2.2 数値データキューブとアイテムセット分析の組み合わせの問題点

一方で，データキューブにより任意の多次元空間上の数値グラフを即計算できる結果として，そのグラフにおける任意の指定状況においてどのような事象の組み合わせが多いのか，をも同時に調べたいという要求が考えられる．図 1 であれば，

Q1:「申込ページを見たレコード集合において，よく見られるページの組み合わせは何か．それは 4，5，6 月と時間が変わるとどう変化するか.」, や,

Q2:「講演会ページを見たレコード数は 6 月に急増している．そのユーザ別アクセス数を計算した時，各ユーザグループ G_1，G_2，G_3 が見た頻出ページ組は何か」

という問い合わせである．

図 2 は，これらの問い合わせを，OLAP による分析の進行と併せて発行した状況を示している．図中，右下部が Q1，左下が Q2 に対応する．このような分析を行うには，着目した状況を満たすレコード集合をまず求め，そこから，調べたい対象となる事象を 1 アイテムとして，高頻度アイテムセットを計算するしかない．Web ログデータ分析の場合，調べたい事象は，「どのページをアクセスしたか」であるので，ページを 1 アイテムとして，同一レコードに高頻度に出現するアイテムセットを求めることになる．Apriori[5] や FP-growth などの列挙アルゴリズムがあるが，計算 1 回あたりのメモリ消費量や CPU 時間の高い技法である．従って, このままでは, OLAP のような自由度の高い多次元問い合わせをアイテムセット分析についてオンライン的に実行できない．

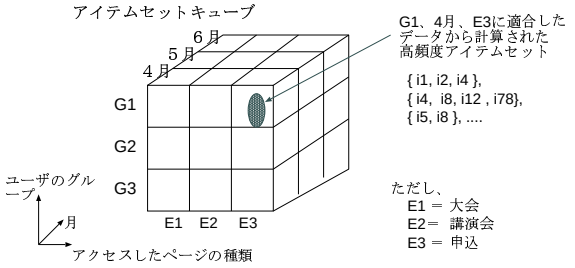

図 3: アイテムセットキューブ
Fig.3 Itemset Cube

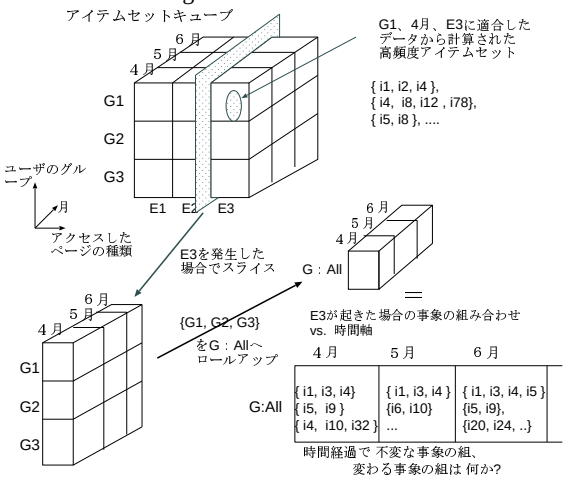

図 4: アイテムセットキューブの変形例
Fig.4 Query processing by an itemset cube

2.3 提案：アイテムセットキューブ

前節で述べた問題に対して，著者らは，高頻度アイテムセットを各セルの値として持つようなデータキューブモデルを提案している．これを「アイテムセットキューブ」と呼ぶ．アイテムセットキューブは，2.1 節の前提と用語の下で，以下のように定義される．

1. データキューブの各セルは，従来の数値データではなく，そのセルの条件を満たすようなレコード集合から計算された高頻度アイテムセットを持つ．ここで, セル c においてアイテムセット I が支持率 s(%) の下で高頻度アイテムセットであるとは, c に当該するレコード総数を N_c とすると, このレコード集合における I の出現回数が $N_c \times s$ 回より大きい場合を指す. ただし, s はキューブを構成する全てのセルにおいて共通の定数として与える．□

アイテムセットキューブをログデータ集合から計算する演算を実体化と呼ぶ．ロールアップとドリルダウンも，従来の数値キューブと同様の変形操作として定義される．(形式的定義は文献 [2] を参照)． ただし, アイテムセットキューブのロールアッ

プなどの変形操作は，セルの値が支持率 s での高頻度アイテムセットであるため，数値データキューブのような単純な集計処理では実現できない．

図 3 は，ユーザドメイン (G_1, G_2, G_3 の 3 グループ)，月 (1-12 月)，アクセスされたページの種類 (「E_i に関するページがアクセスされた」，E_1=大会，E_2=講演会，E_3=申込) の 3 属性で実体化したアイテムセットキューブの例である．

図 4 は，図 3 のキューブを用いて，問い合わせ Q1 =「E_3 に関するページがアクセスされた場合において生じる高頻度アイテムセットを，月別に示せ」を処理する場合である．すなわち,「ページの種類= E_3」でスライスを行い,「ユーザドメイン」次元を「全ユーザ」(G:All) へロールアップすることで問い合わせに応じた結果を計算する．

このように，あらかじめ必要な分割を設定しアイテムセットキューブを実体化しておき，問い合わせに応じてスライス，ロールアップすることで，OLAP と同形式のオンライン的なアイテムセット分析ができる．

2.4 演算方式について

アイテムセットキューブを実現するためには，(i) 実体化が効率良く実現可能であること, 及び, (ii) 概念階層の有無によらずロールアップが効率的に実現可能であること, が必要である．

実体化の高速化技法としては, 排他的分割に対しては単にログデータを分割して別個に計算すれば良い．一方, 非排他的分割によって決められたセル n 個のアイテムセット計算では, 同じログデータ集合が複数のセルに共通に含まれるため, セルごとに Apriori を使うと, セル間で同じアイテムセット列挙処理が発生する. これを避けるため, 著者らは, Cubic Apriori 法 (CA 法) と呼ぶ手法を提案している [1]．CA 法は, 非排他的な分割によって規定されたセル $C_1, C_2, ...C_n$ を同時に実体化する技法であり, 次で与えられる：

[Cubic Apriori 法] 今，アイテムセット I に，各セル $C_i (i = 1, ..., n)$ での，I の出現回数を c_i として，$v(I) = [c_1, c_2, ..., c_n]$ を与える．

手順 1．各レコード r について長さ 1 のアイテムセット I_1 の数え上げを行い，r が満たすセル C_i についてのみ c_i をインクリメントする．全レコードスキャン後にセルあたりの頻度足切りを行い，長さ 1 の高頻度アイテムセット集合 L_1 を確定する．

手順 2．長さ $k(\geq 2)$ の候補アイテムセット I_k を長さ $k-1$ の高頻度アイテムセット L_{k-1} から生成する．I_k が候補となるのは，あるセルにおいて，I_k の長さ $k-1$ の真部分集合が全て高頻度の時に限られる．

手順 3．各レコード r について，手順 1 と同様に候補 I について r が満たすセル C_i についてのみカウンタ c_i をインクリメント．全レコードスキャン後に，頻度足切りを行い，L_k を確定．L_k が空になるまで，$k++$ して手順 2 へもどる．[]

CA 法は，セルごとに個別にアイテムセット計算をしない．そのため，各セルにレコードを分類してからセルごとに Apriori を実行する手法 (Naive 法と呼ぶ) に比べ, 長さ 2〜3 の高頻度アイテムセット列挙処理が 1 回で済む．文献 [3] では, 非排他的分割によるセル k 個で規定された 1 次元のアイテムセットキューブの実体化時間を評価している. アイテム数 1 万, データ数 30 万, レコード平均長 20, 潜在的アイテムセット長 4 で生成した人工データで支持率 1%, Pentiumu 4 (2.4GHz)+1GB メモリ機を用いてセル数 10 から 30 個までの実体化を行った. (ただし, セルの分け方は，第 i 番目の区切り＝「アイテム 300i 番〜300i+299 番のうちいずれかのアイテムを含む」で定義). その結果, セル数=10〜30 に対し, CA 法が 200 秒程度でほぼ一定, Naive 法では 400 秒から 1000 秒程度となり, CA 法の有効性が確認されている．

一方, ロールアップは，排他的に分割されたセル集合について高速に行うことができる．この場合，ロールアップの対象となるセルに含まれる高頻度アイテムセットを全て併合した結果を候補アイテムセットとして，一度だけデータベーススキャンを行い，高頻度アイテムセットを決定するだけでよい．文献 [3] の試験では, 上のデータを 3 万件のデータセット 10 個に排他的にわけ, この次元を d_1 とし, もう 1 つの次元 d_2 を実体化時と同じ非排他的分割によるセル数 5 として, 2 次元のアイテムセットキューブを d_1 に沿って All へとロールアップした. その結果, CA 法で 30 万件を再実体化する場合と, 3 万件ごとの計算結果からマージしてロールアップ処理をした場合とで, 実行時間が 100 秒 vs. 25 秒, 最大メモリ使用量が 1/200 になっている．

3. 実データへの適用

本節では，実データを用いてアイテムセットキューブによる分析の有効性を示す．使用したデータは，ある会員制学術団体の活動を広く公開する Web サイトのアクセスログ 1 年間 (2002 年) であり，ログ数 213863 行である. ただし, 同団体の特徴として, 非会員からのサイトコンテンツ閲覧も自由とし, 大会や講演会, チュートリアルなどの行事参加に際して会員登録を求める制度であった. 検索エンジン等の影響除去後, セッションレコードに変形して 17293 件を得た．

数値データキューブ及び, アイテムセットキューブの実体化は，ユーザのドメイン種別，月，イベント種類, の 3 次元で行った. なお，各次元に与えられた属性の分割は次のとおりである (i) ドメイン種別 : ac ドメイン，co ドメイン，com ドメイン，その他のドメイン, の 4 分割 (排他的) (ii) 月 : 1 月〜12 月の 12 分割 (排他的), (iii) イベント種類 : 講演会のいずれかを見た，大会のいずれかを見た，(会員登録用の) 申込みを見た，all(前出のいずれかに属する), の 4 分割 (非排他的). 結果, 数値キューブ, アイテムセットキューブともに, 4x12x4 = 192 個のセルから成る. アイテムセットキューブの実体化は, アイテムセット長 4 まで行った．

図 5 は，数値データキューブのユーザドメインを全ユーザへとロールアップで集計し,「イベント種類」と「月」の 2 次元にして表示した数値グラフである．これと併せて, アイテムセットキューブもロールアップ処理をして同じ 2 次元のキューブに変換し, さらに「月」属性を四半期単位 (1Q,2Q,3Q,4Q) へとロールアップした. その結果を使って, 同図で楕円で囲まれた部分に, 各々, 講演会に属すページをアクセスした場合の第 1, 第 2, 第 4 四半期, あるいは, 第 4 四半期に申込ページをアクセスした場合に生じている高頻度アイテムセットのうち支持率上位で自明でないものを表示した．

図 5 の数値変化に注目すると, 講演会を見たレコード数が 7 月-9 月の第 3 四半期 (3Q) から以降の第 4 四半期 (4Q) にかけて大きく増加し，それに合わせて all も増加していることがわかる. そこで, この時期に講演会を見たレコード群がどのようなページを組み合わせて見ているかを調べる. 図 5 からは，講演会を見たレコード群が 4Q では kouenkai1023 や taikai24 と組み合わせて，datamining や oracle02oct3 といったこの時期の特別なイベントを見ていることがわかる．

次に, 講演会を見たレコードが増えた理由を調べるため, ユーザドメイン種別の次元を追加する．数値キューブを用いて, 講演会を見たレコード群について,「月」属性の条件を 3Q，4Q とし, ユーザドメイン次元を追加してドリルダウンする. その結果が図 6 である．

この図から，com ドメインと other ドメインが講演会を見たユーザの大半を占めることがわかる．そこで，other ドメインと com ドメインが実際に行った動作を調べるため，アイテム

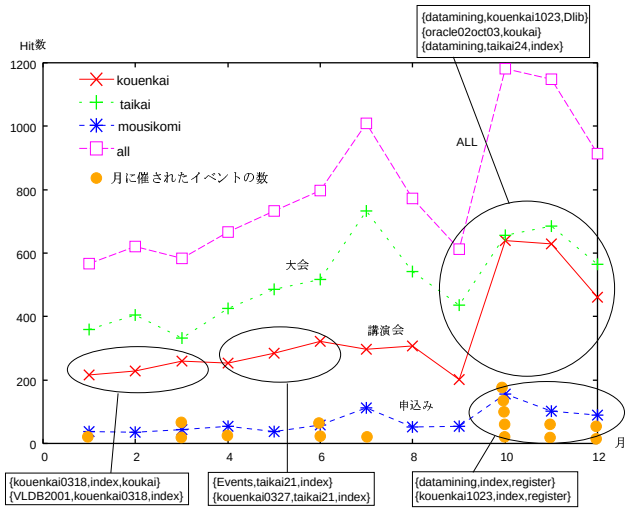

図 5: 各イベント種類の月別ヒット数
Fig.5 Monthly hit counts of event classes

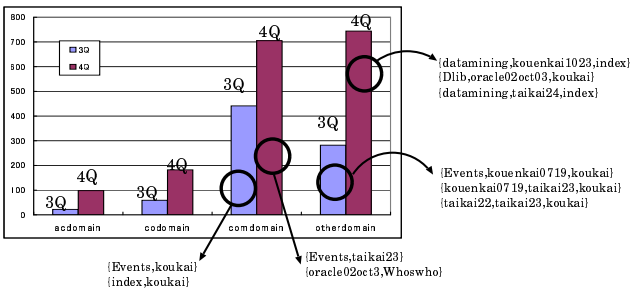

図 6: 講演会を見たレコード群の第 3, 4 四半期におけるドメイン別ヒット数とそのアイテムセット
Fig.6 hit counts and itemsets of users' domains

セットキューブを, 数値キューブの時と同じ演算列を使ってドリルダウンし，高頻度アイテムセットを求めた．図 6 における太線丸枠で示されたアイテムセットがその結果である. これを見ると, other ドメインの 4Q における動作は, 図 5 の 4Q における講演会アクセスユーザの動作と一致するが, com ドメインのそれは一致しない. また, 図では略したが, other ドメインの 3Q から 4Q にかけての動作は, ac や co ドメインのそれと似た動作をとっていることもわかった．ac や co ドメインは会員登録ずみのユーザであることも別途わかっている. これらのことより, 9 月〜12 月の講演会を見たレコード群の増加は，other ドメインが主要因であることがわかった.

4. おわりに

本稿では，膨大なログを理解するために OLAP で用いられている多次元的分析手法をデータマイニングについて行うデータキューブ機構「アイテムセットキューブ」を提案し, その実データへの適用を述べた．ログ分析を行う際に，数値のデータキューブとアイテムセットキューブを併用することで，様々な条件で数値のグラフを描き，グラフから発見した着目点での高頻度アイテムセットを即座に得て分析を行うことができた．この分析手法により，従来の数値による分析ではわからないログの特性を理解することができた．ロールアップの安定した高速化, 及び, 本機構とアプリケーション固有のデータマイニングとの接続などが現在の課題である.

[文献]

[1] T.Ohmori, Y.Tsutatani, M.Hoshi, A Novel Datacube Model Supporting Interactive Web-log Mining, IEEE 1st Int. Symp. Cyber Worlds, pp.419-427, 2002.

[2] 助川，大森，星，蔦谷，Web ログ分析における高頻度アクセスパターン検出を支援するデータキューブモデル, DEWS(データ工学ワークショップ)2003 1-A-01, 電子情報通信学会, 2003.

[3] 成瀬，大森，星，多次元的なログデータマイニングを実現するデータキューブ機構の提案と評価，DEWS2005，3C-i10, 電子情報通信学会, 2005.

[4] B.Prasetyo, et al.， Naviz:User Behavior Visualizations System using Web Access Log, 情報処理学会第 64 回全国大会 6X-01, 2002.

[5] R.Agrawal, et al.， Fast Algorithms for Mining Association Rules, Proc. 20th Very Large Data Bases, pp.487-499, 1994.

[6] S. Chaudhuri, U.Dayal, An Overview of Data Warehousing and OLAP Technology, SIGMOD Record, Vol.26 (No.1), pp.65-73, 1997.

成瀬 正英　Masahide NARUSE
2005 電気通信大学大学院情報システム学研究科修士課程了, 工修. 現在, (株) 日立システム& サービスに勤務し, システム自動管理等の分野に従事. 柔道参段. 情処学生会員, 日本データベース学会学生会員.

大森 匡　Tadashi OHMORI
1990 東京大学大学院工学系研究科情報工学専門課程博士課程了, 工博. 三菱電機, 京都大学を経て 1994 より電気通信大学大学院情報システム学研究科助教授. 関係データベースシステムの高性能化・高機能化, トランザクション処理の研究に従事. ACM, 情処, 信学会 各会員. 旧 SIGMOD 日本支部幹事 ('99-'02). 訳書 (共訳)「トランザクション処理- 概念と技法 -」(グレイ, ロイタ著, 喜連川監訳, 日経 BP).

星 守　Mamoru HOSHI
東京大学大学院計数工学専攻修士課程了, 工博. 電子総合技術研究所, 千葉大学を経て 1992 より電気通信大学大学院情報システム学研究科教授. データ構造とアルゴリズム, 情報理論等の研究に従事. ACM, IEEE, 情処, 信学会 各会員. 訳書 (共訳)「グラフの理論 I〜III」(ベルジュ著, サイエンス社), 著書「データ構造」(昭晃堂) など.