

オンライン機械学習並列分散処理フレームワーク Jubatus

ほりかわ けいたろう^{†1} きたやま ゆずる^{†1} おだ さとし^{†1}

堀川 桂太郎 / 北山 禪 / 小田 哲

うえにし こうた^{†1} ハン ジュンギュ^{†1} まきの ひろゆき^{†1}

上西 康太 / 韓 正圭 / 牧野 浩之

いしい まさくに^{†1} うちかわ しょうへい^{†1} きのした しんご^{†2}

石井 方邦 / 内川 昌平 / 木下 真吾

NTT ソフトウェアイノベーションセンタ^{†1}
NTT 研究企画部門^{†2}

本稿では、ビッグデータをリアルタイムに深く分析するための分散処理フレームワークJubatusについて、概要、解くべき技術課題・目指すこと、設計思想、オープンソースコミュニティ活動、現在までの到達点、そして今後の展開を紹介します。

ビッグデータ分析手法

データベース、データマイニング、データ中心設計など、データの価値と重要性は疑う余地がありません。近年、ネットワークの普及に伴い、多種多様なデータが大量に生成・発信され、ビッグデータとして関心を集めています。ICT技術の進歩が、ビッグデータを集めて、高速に分析することを容易化し、この潮流を加速させています⁽¹⁾。

ビッグデータをビジネスに活用するシーンとして、例えば、10数年分の臨床データなど、ある領域において大量にストックされたビッグデータに対し、あらゆる角度から分析することで、新たな発見、気付かなかった規則性を見出すニーズは今も健在です。この傾向は、特定の領域内に閉じた動きに限らず、異なる分野、専門領域をまたがったビッグデータの関係性を対象に、新たな事業可能性、シナジー効果を素早く発見する可能性を含みます。つまり今後、クロスドメイン^{*1}のビッグデータ分析に発展すると考えます。

ここではビッグデータとその分析手法を大きく2つのタイプでとらえます。

- ① 蓄積されたビッグデータ（ストック型）に対する一括高速分析（バッチ処理）
- ② 連続的に発生しているデータの流れ（フロー型）に対する逐次高速分析（リアルタイム処理）

特に②に対して、不確実で正解が見えない多くの課題が偏在する昨今の外部環境において、曖昧ながらも判断や意思決定を下すニーズと期待が高まっています。

本稿では、フロー型ビッグデータのリアルタイム分析を対象とし、さらにオンライン機械学習による深い分析という付加価値を提供するフレームワーク「Jubatus（ユバタス）」を紹介し

Jubatusとは

名前の由来は、俊敏な動物であるチーターの学術名からの命名で、「ユバタス」と読みます。株式会社 Preferred Infrastructure と NTT ソフトウェアイノベーションセンタの共同開発で、日本発ビッグデータオープンソースとしてWebサイトで公開中です^{(2)~(4)}。

Jubatusはビッグデータを素早く深く分析することを目指します。例えばソーシャルメディアの応用として、全

世界で発信されている毎秒8 000以上のつぶやきを自動分類する用途があります。これは「大量」「速い」「深い」の3つの要件を含みます。つまり、毎秒16 Mバイトのデータ流量に対し、自然言語解析と複数カテゴリへの自動分類を、遅滞なく高速実行するものです。

ところがこの3つの要件は基本的にトレードオフ関係であり、すべて両立することは本質的に困難です。Jubatusが解くべき課題を詳細化すると、「深い分析」と「スケーラビリティ」の両立です。ここでいう深い分析とは、自然言語など人間向けの非構造情報を自動的にカテゴライズする分析です。また、情報の推薦、予測、新たな関係性の発見など、明確に定式化された処理とは異なり、人間の作業を代行し得る処理を意味します。技術的には機械学習、人工知能、パターン認識といった領域にかかる課題です。

一方のスケーラビリティは、①処理要求の増大と②データサイズの増大という課題を含みます。①はさらに、スループット（単位時間当りにさばく要求の量）とレスポンス（1件ごとに遅滞なく応答する）に分けられます。一般的にバッチ処理はスループット、リ

*1 クロスドメイン：異なるドメイン間をまたがったデータに対してアクセスすること。

リアルタイム処理ではレスポンスを重視します。②については、データ自体を持たなくて済ませるか、分割して保持するかでアプローチします。

Jubatusは、不確実で曖昧さが残る中で迅速な判断を下すニーズにこたえること、常に判断材料を集め続けて、遅滞なく判断するための設計思想を徹底的に検討しました。具体的には、深い分析（機能）とスケーラビリティ（非機能）を分離します。深い解析の設計は、オンライン機械学習のロジックを「エンジン・CPU」に例えて、着脱可能な分析モジュールとして持続的に拡充可能とします。スケーラビリティの設計は、共通的な基盤「シャーシ・マザーボード」とみて、共通の枠組みで分析モジュールを搭載しスケールさせる設計とします。最終的にはすべての人にスケーラブルな機械学習を提供することが目標です。ハードウェアは廉価なコモディティサーバを多数並べて分散処理させるスケールアウト、ソフトウェアは一部のデータ科学者、プログラマ、専門家向けに限定せず、ビッグデータの使いやすいオンライン機械学習フレームワークを広く分かりやすく提供していく方針です。

適用領域

全世界のWebページの重要度を計算する手法として、GoogleのPage Rankが有名です⁽⁵⁾。Webサイトのリンク構造は比較的緩やかに更新されるためバッチ分析がフィットします。一方、ソーシャルメディア（例えばTwitterのつぶやき）はWebより粒度は小さく、内容は軽く、広いユーザー層から大量、小刻みに発信され、情報の鮮度が

大切という特性を持っています。外部環境の変化や事象に広いユーザー層がどう反応するか、ソーシャルメディアを通じて分析し、事業に応用するケースで、フロー型ビッグデータのリアルタイム分析が有効活用できます。

Jubatusの応用として、Twitter情報の一次分類を用意しました。情報表現が自然言語であり、限られた字数に収めるために、独特の省略、短縮、造語、隠語、癖のある表現が分析対象に含まれます。日本語だけで毎秒2,000以上あるつぶやきの中から、自分に役に立つものを人間の目視で瞬時に抽出、整理することは困難です。そこで、大まかな分類・絞り込み機能を提供する汎用的な一次フィルタが役に立ちます。例えば、「企業」というカテゴリで

分類するデモを図1に示します。東証一部上場約1,600社を分類カテゴリとして、毎秒2,000のつぶやきがどの企業に関連するかを高速分類し、分析アプリケーションに情報提供します。この実装にJubatusのclassifier（多値分類）というオンライン機械学習手法を適用しました。信用性が高い公開情報としてWikipediaなどを教師データとして活用し、分類のための学習モデルを自動構築しました。キーワード一致や関連ワードによる記事分類と異なり、n次元特徴ベクトル化された空間^{*2}を複数カテゴリに分割する分離平面のベクトル方程式をオンライン学習により

*2 n次元特徴ベクトル化された空間：nは日本語を特徴づける単語数で、ここでは約2,000程度を目安としました。



図1 ソーシャルメディアの一次分類（企業1,600社にカテゴリライズ）

更新します。

また、外部からビッグデータを取り込み続けることで勝手に賢くなり、未知の単語も適切に分類していく効果が期待できます。一次分類するカテゴリは「企業」に限らず、「国、都道府県、市町村」「著名人」「商品」など幅広く応用可能です。

アーキテクチャと機能の概説

Jubatusの設計思想概要を図2に示します。前述のとおり、Jubatusは機械学習エンジン群とそれらを共通に支える高速フレームワークで構成されています。従来の機械学習エンジン単体では、データサイズは小～中規模、バッチ処理、個別の開発が必要なケースが一般であるのに対して、Jubatusは多様なエンジンを高速フレームワークに搭載し、許容範囲内で誤差を容認しつつビッグデータを高速処理する仕組みを共通仕様で開発できる点が差異化ポイントです⁽⁶⁾。2012年6月現在、3回のアップデートを重ねて、分析アルゴリズムのラインアップ増強、支えるフレームワークの強化を図っています。これまでに対応した分析アルゴリズムについて表1に概説します。

Jubatusの本質を引き出す用途と

は、差し迫った状況において正解を明日まで待てないスピード判断の支援があげられます。データ量の観点では、特定領域に閉じず、異なるドメイン間にまたがり、新たな関係性を分析・発見できる効果が期待できます。

分散処理アーキテクチャ

分散処理基盤の詳細な構成を図3に示します。ビッグデータのストリームは左から右へ流れます。クライアントは、複数のユーザ・プロセス、プロキシ・プロセスから構成されます。プロキシ・プロセスが、クライアント・サーバ間に介在し要求を転送することで、アプリ側はサーバの数・所在・稼動状況など詳細を意識せずにアプリケーションロジックに集中できます。ユーザ・プロセスはJubatusクライアントAPIを用いてスクリプト言語や汎用プログラミング言語で記述できます。

オープンソースコミュニティに対する外部の貢献により、Python、Ruby、Javaなど多様な言語バインディングが利用可能です。プロキシ・プロセスとサーバ・プロセスの通信はmessage pack⁽⁷⁾ベースのRPC (Remote Procedure Call) が基本です。非ブロック型I/Oにより、通信、同期制御を効

率化します。また、Zookeeper⁽⁸⁾プロセスが、プロキシ・プロセスとサーバ・プロセス間の連携や、分散サーバ間調停役を果たし、サーバ死活監視・リーダの選定を実施します。

Jubatusの分散処理の特徴をバッチ処理と比較します(表2)。

例えば、サーバを1000台並べて分散処理する場合、バッチ処理 (Map-Reduce) では1000台が同時にMapし、その後一斉にReduceして結果をまとめるのに対し、Jubatusでは1台1台が自律的に学習と分析を繰り返しています。

分散サーバ間の並列処理において、深い分析とスケーラビリティを両立するための技術的工夫がMIX処理です^{(6),(9)}。MIX処理は、自学自習と答え合わせの勉強会に例えて説明できます。データ1つひとつの学習・分析のたびに、1000台全員で答え合わせすると、全体として待ち合わせ時間の膨大なロスを生じますが、適度な頻度で

表1 クライアントAPIの一例

分析	代表的なAPI
分類	学習要求
	分類要求
帰	学習要求
	類推要求
推薦	行データ (user ID/items) の更新・削除
	類似行の検索
	推進アイテムを抽出
統計	データの更新
	合計、偏差、最大・最小、エントロピー、モーメント計算
グラフ	ノード生成・削除・更新・参照
	エッジ生成・削除・更新・参照
	中心性スコア計算
	2ノード間最短路計算

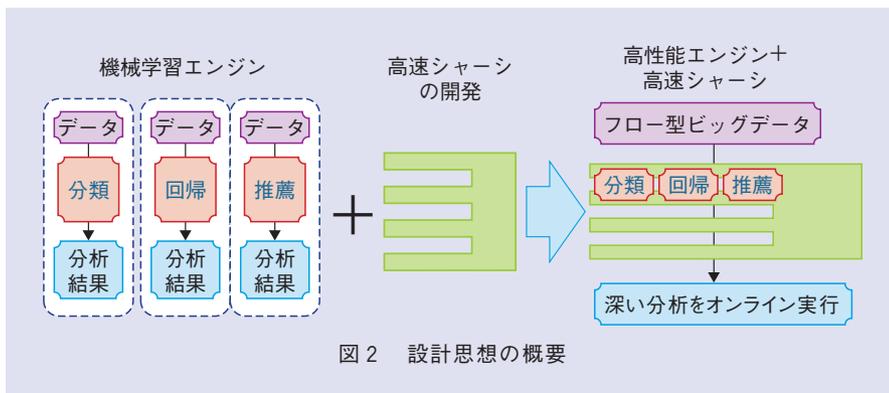


図2 設計思想の概要

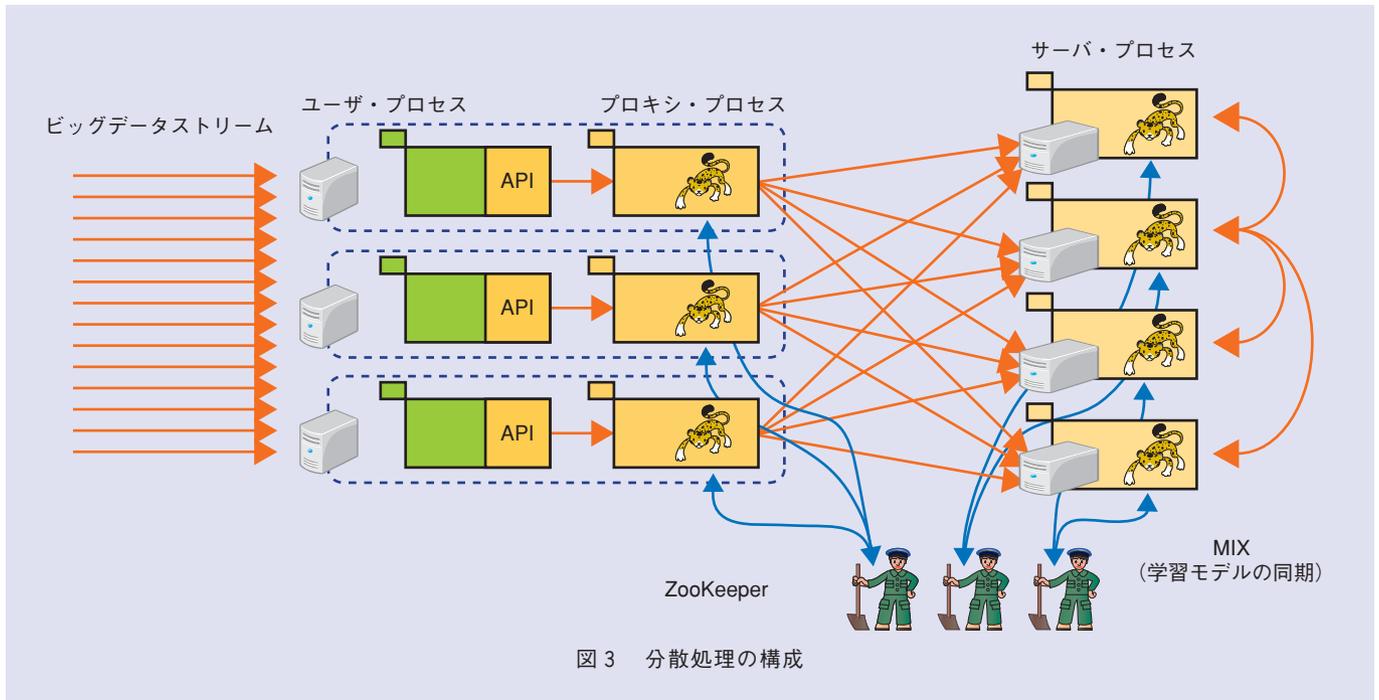


図3 分散処理の構成

表2 Jubatusの分散処理の特徴

	バッチ処理	Jubatus
処理能力	ストック型：ためて一気に計算	フロー型：ためずに小まめに計算
稼働時間	定期便方式：開始・終了が明確	24時間×7日：開始・終了がない
処理のタイミング	今のデータは次の便まで待たされる	今のデータを待たずにすぐ計算

答え合わせする（＝同期をとる）ことで、全体の性能を低下せずに学習と分析を実行できます。

次にMIX処理と、統一的なインタフェースUMAを紹介します。それぞれ、

- ・Update：更新系クエリ，ビッグデータ追加・変更，非同期実行
- ・Mix：分散サーバ間のゆるやかな分析結果の共有
- ・Analyze：クライアントからの参照クエリ，分析リクエストを実行を意味する抽象インタフェースです。分析ロジックに依存しない共通インタフェースを実現するねらいがあります。今後も継続的に機械学習エンジンを拡

充しますが、一貫したインタフェースで分かりやすいフレームワークを設計・維持していくのが重要です。

開発効率の向上

Jubatus開発のアジリティを向上し、開発者自らを楽にするために、設計やツールを充実化しています。

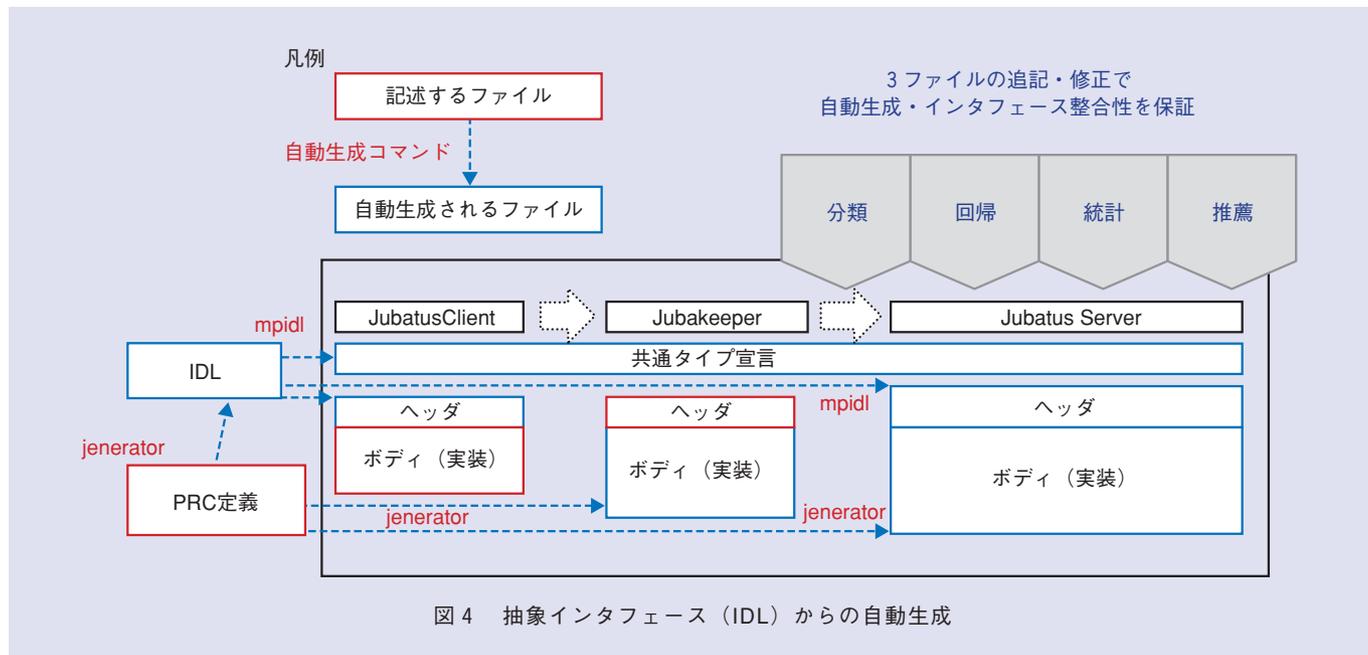
(1) 特徴ベクトル変換

機械学習の入力形式として、どんなデータも特徴ベクトル化する必要があります。逆に、特徴ベクトル化が成功すれば、機械学習アルゴリズムが後は適切に処理します。その重要な特徴ベクトル化を、プログラムロジックとしてハードコードせずに、外出しの設定

ファイルで定義させるツールを用意しました。

(2) IDL, jenerator

Jubatus 0.01版では、解析ロジックの追加に伴い、その都度、手作業で7つのファイルを編集・確認する必要がありました。そこで私たちはインタフェースの整合性を自動的に保ち、インタフェース修正作業を局所化する仕組みを考案しました。インタフェースをIDL（Interface Description Language）ファイルで定義し、jeneratorと呼ぶツールを通すことで7つのファイルの雛形スケルトンを自動生成させます（図4）。その結果、100行の定義ファイルから約3000行のコードを自動生成し、IDLによるソースコードメンテナンス作業の集約化が進みました。またIDLを読むだけでAPIの意味や使い方が理解容易化し、開発効率を大きく向上できました。



性能評価

分析ロジックを拡充しながら、事前性能検証を行いました。Jubatusのスケラビリティを評価します。

- ① 分類：世界規模のTwitterつぶやき (8 000 ツイート/秒) をコモディティサーバ 2 ~ 3 台で分類。10 秒間で得たデータからバッチ処理の 90% の正解率に到達。教師データ量をさほど多く必要としないケース。
- ② 推薦：購入履歴のリアルタイム更新では、1 サーバ当たり 10 万回/秒の更新が可能。3 000 万ユーザに対するお勧め検索で、レスポンス 0.1 秒で推薦可能 (Mahout の約 10 倍高速)。サーバ台数増加に対し、スループット (登録データ件数/秒) が線形にスケールすることを確認。
- ③ グラフマイニング
 - ・ 1 億エッジ追加：コモディティサーバ

パ10台×5分で実施。

- ・ エッジ追加のサーバ台数増によるスケラビリティ：(1, 2, 4, 8) 台 ⇒ (3, 6, 13, 25) 万エッジ/秒 (スループット)。
- ・ 分析のレイテンシ：{update, mix, centrality}: 0.1 ~ 0.3 μ秒/ノード, {shortest-path}: 0.1 ~ 数 10 μ秒/ノード (約 1 000 ホップのパス) ⇒ グラフデータベース製品 (neo4j) の数 100 ~ 1 000 倍高速。
- ・ データサイズ：1 台 (8 GB メモリ) 当たり 1 600 万強のエッジ保持 ⇒ 50 GB で 1 億エッジ保持。

オープンソース状況と今後の予定

2012年7月現在の到達点は、3回のバージョンアップとOSS公開の実施です。専門家、プログラマ、開発者向けに使いやすい基盤がそろってきました。また、データサイエンティスト向けに、試したい分析アルゴリズムを載

せて、高速にスケールする基盤も整ってきました。引き続き、オープンソースコミュニティの活性化に取り組みます。

2012年7~9月にかけ、Jubatus分析アイデアコンテスト (Jubatus Challenge Japan 2012) を実施しました^{(10), (11)}。このコンテストのねらいは、オープンイノベーションの積極活用により、Jubatusのポテンシャルを最大限に引き出すアイデアとビジネスシーンの探索です。重要なポイントは、Jubatusエコシステムの設計です。Jubatusの基盤技術、機械学習アルゴリズム、ビッグデータ、データサイエンティスト、いずれも不足することなく関係する主体が有機的に連携する中で化学反応を触媒するような場、つまり図5に示すようなリアルタイム機械学習分析のエコシステムを設計することが重点課題です。

今後の展開

従来のビッグデータ分析ビジネスは、

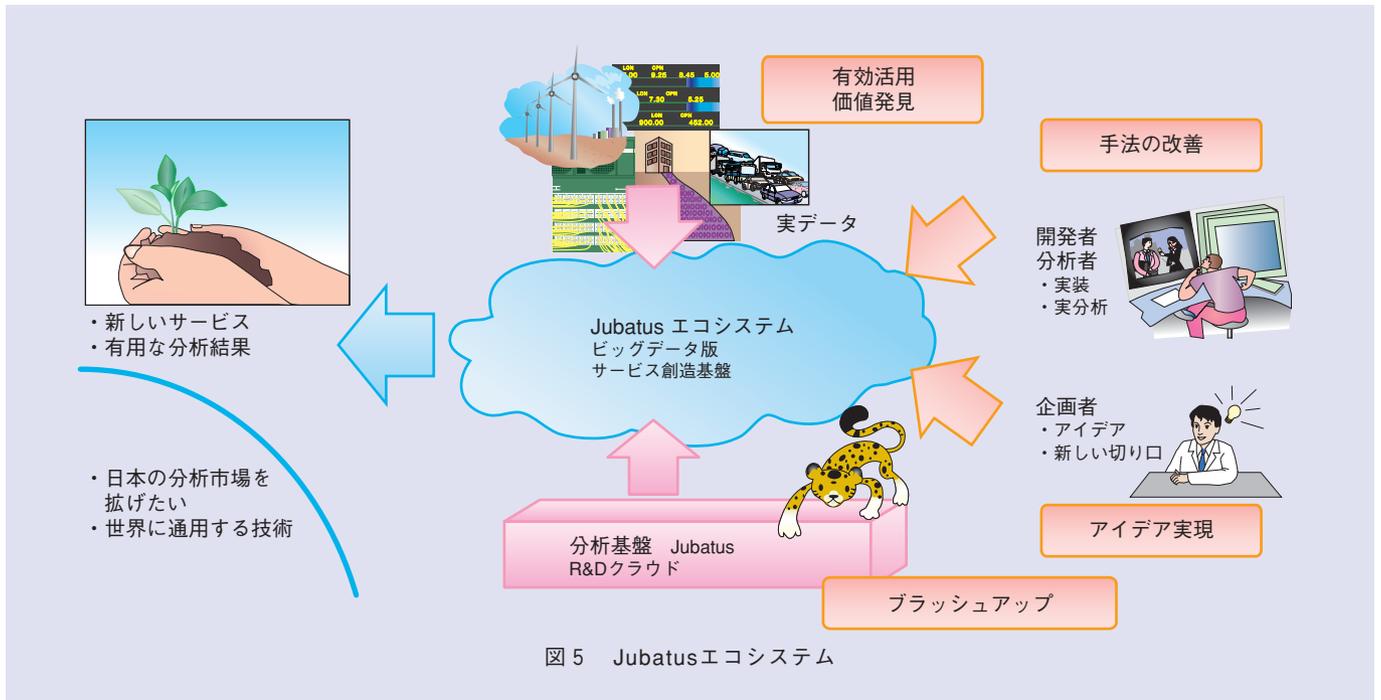


図5 Jubatusエコシステム

データマイニングにより今まで気付かなかった関係性発見を事業利用すること、その際、バッチ処理分析では10数年分の臨床データなど、ストックされた大量データがありきで試行錯誤的に多様な角度から分析し、新たな発見を得ることが中心でした。これからは、下記の観点から研究開発をしていくことが重要となります。

- ① クロスドメイン関係性分析：特定領域に閉じたビッグデータ分析に対し、異なる分野、専門領域をまたぐビッグデータの関係性を対象に、新たな事業可能性、シナジー効果を発見。
- ② リアルタイム分析：フロー型データを対象に分析し、迅速に判断することで、スピード経営、競争優位性の創出。
- ③ 深い分析，リアルタイム，ビッグデータ分析：厳密な分析結果を明日・明後日まで待てない，人間

の判断（意思決定ルート）を介したアクションでは遅い， μ 秒単位の自動判断において，あらかじめ与えられた決定的ロジックでない，状況の変化を速やかに反映した自動判断（機械学習ベース）へのニーズにこたえる高付加価値の技術開発。

今後も持続的に，オープンソース開発と，オープンイノベーションを加速し，実用事例の展開を推進していきます。

■参考文献

- (1) <http://www.ngs-forum.jp/event/seminar/report/detail.php?cno=w2012040401>
- (2) <http://jubat.us/>
- (3) <https://twitter.com/jubatusofficial>
- (4) <https://github.com/jubatus/jubatus>
- (5) <http://www-db.stanford.edu/~backrub/pageranksub.ps>
- (6) http://www.itrc.net/report/meet31/data/meet31_cis_ini_2.pdf
- (7) <https://github.com/msgpack/msgpack-rpc>
- (8) <http://zookeeper.apache.org/>
- (9) 小田・中山・上西・木下：“Jubatus: Big Dataのリアルタイム処理を可能にする分散処理技術,” 信学技報, Vol.111, No.409, IN2011-126, pp.35-40, 2012.
- (10) <http://www.ntt.co.jp/news2012/1207/120710a.html>

(11) <http://www.facebook.com/JubatusChallenge2012>



(後列左から) 石井 方邦/ 堀川 桂太郎/
北山 禪/ 木下 真吾
(前列左から) 上西 康太/ 韓 正圭/
内川 昌平/ 小田 哲/
牧野 浩之

Join US ! Let's Jubatus (<http://jubat.us/>)
仲間、募集中です。皆様のオープンソース
コミュニティへの参加をお待ちしています。

◆問い合わせ先

NTT ソフトウェアイノベーションセンタ
分散処理基盤技術プロジェクト
TEL 0422-59-3970
FAX 0422-59-5652
E-mail horikawa.keitaro@lab.ntt.co.jp