

まえがき

あなたは最高のユーザ体験を提供することを目指して日々プログラミングをするウェブエンジニアでしょうか？ それとも、ウェブサービスの成長を目指して施策を考えて実行に移すウェブマーケターでしょうか？ どんな形であれウェブサービスの開発に携わるかぎり、ユーザにより良い体験を提供することは無視することのできない命題です。よりウェブサービスが市場のニーズに応えられるものに成長を遂げ、より多くのユーザがより深くサービスを使ってくれるようになることほど嬉しいことはありません。

しかし、目の前にいるユーザは、ウェブサービスを開発するあなたとは異なる人間です。あなたが良かれと思ってウェブサービスに施した変更は、もしかしたら一部のユーザには致命的な悪影響を及ぼしているかもしれません。一方で、あなたがウェブサービスに埋め込んだバグが、一部のユーザには熱烈な歓迎とともに迎え入れられているかもしれません。ユーザがそのサービスを使っている理由はさまざまで、あなたが思いも寄らないようなものかもしれません。結局、そのウェブサービスを知りすぎてしまったあなたは、ウェブサービスを利用する一般的なユーザ像としては機能しなくなってしまうのです。

たとえば、みなさんは2010年頃にローンチされたBurbn（バーバン）というウェブサービスをご存知でしょうか？ Burbnは位置情報に基づいたソーシャルネットワークサービスで、ユーザが近所のレストランやショッピングモールにチェックインしたり、そこで撮影した写真を共有したりできます。開発者の2人は熱心に機能を追加し、もっと多くのユーザにサービスを楽しんでもらえるように努めましたが、その成果はどれもイマイチでした。それどころか開発者が注意深くユーザの行動を観察してみると、ほとんどのユーザは位置情報機能を使っていなかったのです！ユーザが熱心に使っていたのはオマケの写真共有機能、特にそのフィルター機能でした。当時のス

スマートフォンのカメラの画質は今ほど良くなかったので、写真をきれいに編集してくれる機能が求められていたのです。そこで開発者の2人はフィルターと写真共有以外の機能をすべて取り払って、このウェブサービスを再スタートさせることにしました。それが今日のInstagram（インスタグラム）です。いまや世界中で使われているInstagramの開発者でさえ、最初からユーザのニーズが完全に見えているわけではなかったのです。

自分自身の感覚があてにならないのなら、何をよりどころにしてウェブサービスを開発していけばいいのでしょうか？ その最善の方法のひとつは、ユーザに直接聞いてみることでしょう。なぜならユーザが求めているものを知っているのは、そのユーザ自身であるはずだからです。先ほどのInstagramの例でも、もし開発者の2人がユーザの行動を観察することがなかったらこれほどの大成功はなかったでしょう^{†1}。

しかし、だからといってユーザの言うことがすべていつも正しい、ということではありません。なぜなら、誰も自分が求めているものを正確に言語化できるとは限らないからです。こんな機能があつたら便利だと思っていたけれども、実際に触ってみると意外と使いにくかったなんてこともあるでしょう。開発者のあなたがユーザの要求を機能として実装したところで、それを本当に使ってもらえる保証などどこにもないのです。

満たすべき要求もそれを満たす方法も不明瞭ななか、唯一のよりどころとなるのがユーザが実際にとる行動です。なぜなら、それこそがユーザにもたらされた体験が結果として表出したものだからです。ユーザの行動の変化がデータとして表されてはじめて、ウェブサービスの成長を定量的に評価できるようになります。これは開発者もユーザもまだ見ぬ理想のウェブサービスの姿を、データを介して対話しながら探求するプロセスだと考えられます。開発者は、ユーザの行動を観察して得た洞察をもとに仮説を打ち立て、新たな施策としてウェブサービスに導入します。そして、ユーザがその施策に対して反応を起こすことで、ユーザ行動に変化が現れます。開発者はその行動の変化を観察することで仮説を検証し、新たな洞察と仮説につなげます。この仮説検証のサイクルを繰り返すことが、問題も解法も不明瞭ななかでウェブサービスを成長させる着実な方法なのです^{†2}。

そして、この仮説検証のもっともシンプルな形が、A/Bテストです。A/Bテストで

†1 Burbnのように大きくサービスの方向性を変えることをピボットと呼びます。ピボットによって成功を取めたウェブサービスのストーリーについては [Holiday14] や [Ries11] を参照してください。

†2 仮説検証に基づいてウェブサービスを成長させる方法論はリーンスタートアップとして広く知られるようになりました。リーンスタートアップについて詳しくは [Ries11] を参照してください。

はウェブサービスに訪れるユーザをランダムにAとBの2つのグループに分けて、ひとつのグループには施策を導入していないバージョンのウェブサービスを、もうひとつのグループには施策を導入したバージョンのウェブサービスを提供します。この2つのグループの間のユーザ行動の違いを観察することで、施策の効果を評価します。ユーザインタフェースの見た目の変更にかぎらず、表示内容そのものを決定するアルゴリズムの変更など、バックエンドを含めたあらゆる施策に使えます。

このように言うとお変簡単なアイデアに聞こえますが、実はこの裏にはさまざまな奥深いトピックが眠っています。いったいどれだけのユーザ数のデータをもとに判断すればよいのか、そもそも何を計測すればいいのか、その指標は本当にウェブサービスの健全な成長を意味しているのか、今試している施策が考える全てのなか……これらの多くの問題にはまだ明確な答えが出ていません。というよりも、さまざまな厳しい仮定を置かないかぎりまともに扱えないと言ったほうが正確でしょう。

そんななか、ユーザの行動について得られたデータから少しでも誠実な結果を得るために強力な武器となるのが統計学、そして機械学習の知識です。統計学および機械学習の知識を用いることで、さまざまな要因（人間であるユーザの個性も含まれます！）によるばらつきが入ったデータからでも、ある一定の結論が出せるようになります。また、分析の裏側に置いたさまざまな仮定も定量的に扱うことで、分析の限界を理解する手助けにもなります。

本書では、開発者とユーザの間で行われる仮説検証のサイクルを通してウェブサービスの何らかの指標を最大化もしくは最小化することをウェブ最適化と定義し、ウェブ最適化を適切に行うための数的手法を解説します。本来「ウェブ最適化」という言葉は広い意味を持つものであり、ウェブページの表示にかかる時間を最小化すること（パフォーマンス最適化）や、検索エンジンでの表示順位を最小化すること（検索エンジン最適化）なども含みますが、本書ではこういった話題は扱いません。

筆者はもともと基本的な統計の知識もおぼつかないウェブエンジニアでしたが、ユーザにより良い体験を提供するためにデータに基づいた判断をしようと研究するうちに、いつのまにかこの学問に入門していました。この本は、そんなウェブエンジニアの私がA/Bテストをはじめとする数理的な仮説検証の手法を研究するうちに歩んだ道筋を一緒にたどりながら、機械学習および統計学の基礎に入門する本です。ウェブサービスやモバイルアプリなどのプロダクトの開発に何らかの形で携わる方々に、ウェブ最適化を通して新たな機械学習の世界の見方を提供できれば幸いです。

対象とする読者

本書が対象とする読者は、下記のような方々です。

統計学もしくは機械学習に入門したいと考えているウェブエンジニア

普段はウェブサイトのフロントエンド開発もしくはサーバサイド開発を中心に
行っているが、業務で直接機械学習に触れる機会があまりないエンジニアの
方々を想定しています。本書のサンプルコードを実際に動かして理解を深める
ことで、機械学習システムをウェブサイトのユーザ体験向上に応用できる知識
と感覚を得てもらうことを目的としています。

ウェブマーケティングに携わるウェブ担当者、ウェブマーケター

普段からウェブサイトの訪問者数拡大や収益率向上に向けて施策を考え、実行
しているウェブ担当者・マーケターの方々を想定しています。本書で紹介する
さまざまなウェブ最適化のシナリオを見ることで、自身が直面している問題に
適したアルゴリズムおよびそのアイデアを理解してもらうことを目的としてい
ます。よりよい施策の策定や、チーム内での効率的なコミュニケーションの助
けとなれば幸いです。

機械学習の応用、特に人間とのインタラクションへの応用に興味のある学生

画像認識や自然言語処理などのトピックは機械学習の応用として代表的です
が、一方でウェブ最適化に見られるような、人間とのインタラクションを扱っ
たシステムへの応用はあまり取り上げられる機会が多くないように思います。
本書で紹介する最適化アルゴリズムおよびその発展の流れを追うことで、他の
書籍とは異なる角度で機械学習技術を眺めてもらえれば幸いです。

本書のサンプルコードはPython 3によって書かれています。Pythonでのプログラ
ミングが初めての方は、『Python チュートリアル』[Rossum11]などの書籍やウェブ
サイトを通じて簡単に学んだ上で読みすすめることをおすすめします。本書で前提と
する数学の知識は、高校数学で学ぶ線形代数（ベクトル、行列）と微分・積分程度で
す。適宜必要な数学は補足して説明しますので、安心して読み進めていただければと
思います。特に、本書を読み進める上で必要な線形代数の知識は「付録A 行列演算
の基礎」にまとめてありますので、適宜そちらもご参照ください。

本書の構成

本書の構成は図1のようになっています。

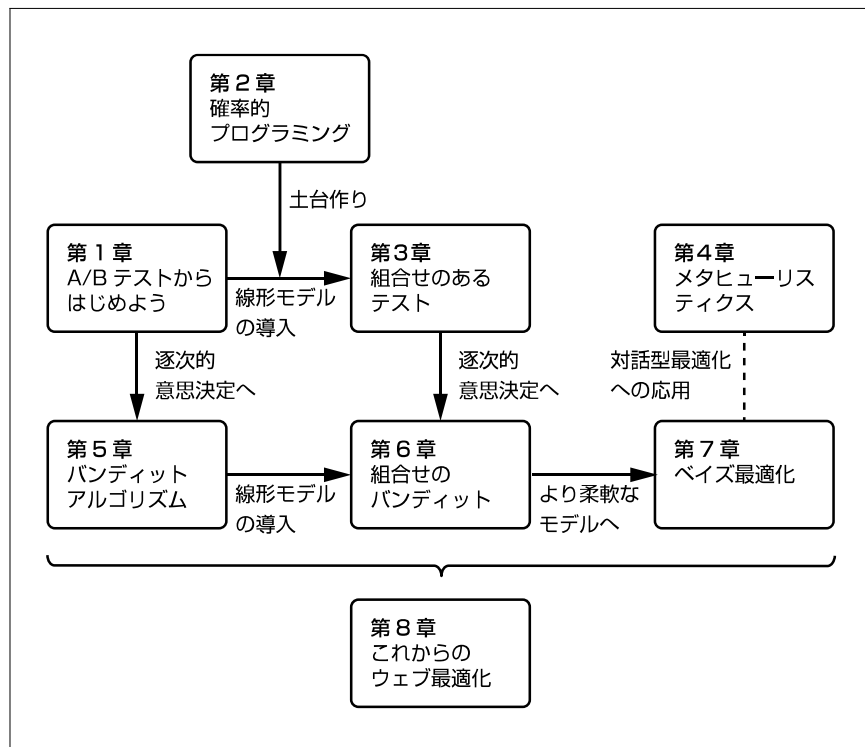


図1 本書の構成

「1章 A/Bテストからはじめよう: ベイズ統計による仮説検定入門」では、仮説検証のもっとも単純な形であるA/Bテストを取り上げ、得られたデータから数理的手法、特にベイズ推論に基づいて意思決定をする手法を説明します。

「2章 確率的プログラミング: コンピュータの助けを借りる」では、数式の展開よりもプログラミングを中心的に使ってベイズ推論する方法として、マルコフ連鎖モンテカルロ法 (MCMC) を取り上げます。ここでは特にMCMCライブラリである

PyMC3^{†3}を使って柔軟な統計モデリングをする方法を学びます。

「3章 組合せのあるテスト: 要素に分解して考える」では単純なA/Bテストを発展させて、施策がなんらかの組合せの構造を持っている場合を取り上げます。ここでは線形モデルを導入することで、少ないデータでも効率的に仮説検証をするための方法を紹介します。

「4章 メタヒューリスティクス: 統計モデルを使わない最適化手法」は、ここまでで紹介した統計的手法とは異なるアプローチで最適な施策を探索する手法として、メタヒューリスティクスを取り上げます。なんらかのモデルを仮定しないアプローチであっても効率の良い探索が可能であることを見ると同時に、人間とコンピュータが対話しながら最適な解を見出す対話型最適化への応用についても考えます。

「5章 バンディットアルゴリズム: テスト中の損失にも向き合う」ではふたたび冒頭のA/Bテストの問題に戻り、すべての実験データが与えられてから意思決定を下すのではなく、実験しながらも逐次的に意思決定を下すアプローチへの発想の転換を扱います。そして、このような問題の定式化として多腕バンディット問題およびその解法を説明します。

「6章 組合せのバンディット: バンディットアルゴリズムと統計モデルの出会い」では、線形モデルを導入したバンディット問題を考えます。これは、3章で扱った内容を逐次的な意思決定を下すバンディットアルゴリズムに拡張することに相当します。これによってより高速な最適化、さらにはパーソナライゼーションに手法を拡張できることを解説します。

「7章 ベイズ最適化: 連続値の解空間に挑む」では、線形モデルをさらに柔軟なモデルであるガウス過程に拡張することで、より複雑な問題にもバンディットアルゴリズムを応用できることを説明します。その結果4章と同様に、問題に特定の仮定を置くことが難しいような問題、特に人間の感性を扱うような対話型最適化問題への解法としてベイズ最適化が期待されることを解説します。

最後に、「8章 これからのウェブ最適化」ではウェブ最適化にまつわる今後の課題を取り上げます。本書で紹介する手法は、数理的な問題として人間であるユーザとウェブサイトの相互作用を扱うために、さまざまな仮定を含んでいます。それらの仮定をひとつひとつ解きほぐしながら、ウェブ最適化の未解決の問題を概観します。

†3 PyMC3 Documentation <https://docs.pymc.io/>