

卒業論文

情報検索における Contextual Bandit を用いた
適合性フィードバックの利用

安江 優人

2022年2月9日

岐阜大学 工学部 電気電子・情報工学科 情報コース
鈴木研究室

本論文は岐阜大学工学部に
学士（工学）授与の要件として提出した卒業論文である。

安江 優人

指導教員：

鈴木 優 准教授

情報検索における Contextual Bandit を用いた 適合性フィードバックの利用*

安江 優人

内容梗概

近年、多数の賃貸物件検索サイトが利用されている。現在の物件検索システムは、探したい地域や最寄の沿線、駅などを選択し、家賃や間取り、駅までの徒歩の時間など希望物件の条件を入力し、該当物件数を絞り込むことが主流となっている。しかし、初期検索で適切な条件を入力しなければ、利用者にとって不必要な物件が多く表示され、利用者が必要な物件を見つけるには困難である。これを解決するために、既存の適合性フィードバックでは主に Rocchio の式が用いられる。適合性フィードバックとは、利用者に検索結果を提示し、検索結果に適合・不適合のフィードバックをしてもらう。フィードバック結果を用いて、利用者に適合とされる検索結果を提示するために検索クエリの更新などを行う。このような利用者との対話を通して、希望する物件条件を推定できないか考えた。Rocchio の式を用いたとしても、検索クエリが表示物件と類似したものであると、複数回フィードバックを行っても検索クエリがあまり変化せず活用のみを行うため、類似した物件しか表示しない。これを解決するために、探索と活用を用いる Contextual Bandit を適用した。探索では、利用者が適合とした物件とは類似しない物件を表示し、利用者の物件選択の可能性を広げる。活用では、探索で得られた結果を用いて利用者に適合する物件を推定し表示する。Rocchio の式を比較手法として、シミュレーション実験と利用者実験の 2 種類の実験を行った。シミュレーション実験では、一人暮らしと 3 人暮らしの利用者を想定して初期検索クエリと物件選択条件を設定した。利用者実験では、利用者の設定を提示し、研究室のメンバー 5 名ずつに実験を行ってもらった。評価はフィードバックされた物件 10 件のうち、適合物件が 7 件になるまでのフィードバック回数とし、収束すると定義した。実験の結果、いくつかの

*岐阜大学 工学部 電気電子・情報工学科 情報コース 卒業論文, 学籍番号: 1183033160, 2022 年 2 月 9 日.

実験で提案手法の方が比較手法よりフィードバック回数が少なかった。最もフィードバック回数の改善が見られたのは比較手法では収束せず、提案手法で収束したフィードバックの平均回数が 9.7 回であった実験だった。シミュレーション実験から、家族 3 人暮らしより、一人暮らしの物件を探す利用者に対して少ないフィードバックで利用者に適合の物件の提案ができると考えられる。しかし、利用者実験で比較手法の方が提案手法よりフィードバック回数が少ない実験が多く見られた。具体的には、物件選択条件による適合物件数が多いと考えられる場合に、比較手法のフィードバック数より提案手法のフィードバック数の方が最大で 16.3 回多い結果となった。物件選択条件による適合物件数に関わらず、少ないフィードバックで利用者に適合する物件の提案を行えるよう改善を行う必要がある。

キーワード

ユーザー支援, バンディットアルゴリズム, Contextual Bandit, ベイズ推定

目次

図目次	v	
表目次	vi	
第 1 章	はじめに	1
第 2 章	基本的事項	4
2.1	適合性フィードバック	4
2.2	Rocchio の式	4
2.3	バンディットアルゴリズム	5
2.4	Contextual Bandit	9
2.4.1	アームの提示	10
2.4.2	事後分布の計算	10
2.4.3	MAP 推定量 $\hat{\theta}_t^{MAP}$ を求める	12
2.4.4	事後分布の近似	14
2.4.5	最適なアームの計算	15
2.5	Bag of Words	15
2.5.1	単語辞書の作成	15
2.5.2	各単語のベクトル化	16
2.5.3	各文のベクトル化	17
第 3 章	関連研究	18
第 4 章	提案手法	20
4.1	検索 Web アプリケーション	20
4.1.1	アプリケーションの流れ	20
4.1.2	検索クエリ入力画面	22
4.1.3	Contextual Bandit の実装	23
4.1.4	物件表示画面	24
4.2	システムの流れ	25

4.3	物件データの前処理	27
第 5 章	評価実験	29
5.1	実験データ	29
5.2	比較手法	30
5.3	評価方法	31
5.4	シミュレーション実験	32
	5.4.1 実験設定	32
	5.4.2 結果	34
5.5	利用者実験	36
	5.5.1 実験設定	36
	5.5.2 結果	36
第 6 章	おわりに	39
	謝辞	41
	参考文献	43
	発表リスト	44

目次

2.1	プレイしていないスロットマシンの確率分布	7
2.2	当たった回数の多いスロットマシンの確率分布	7
2.3	外れた回数の多いスロットマシンの確率分布	7
2.4	$f(x)$ と接線 1	13
2.5	$f(x)$ と接線 2	13
4.1	アプリケーションの流れ	21
4.2	初期検索クエリ入力画面	21
4.3	物件表示画面	22
4.4	チェックの有無の送信	24

表目次

4.1	検索クエリ一覧	23
4.2	架空の物件データ	26
5.1	MAP の計算例	31
5.2	一人暮らしの初期検索クエリ	32
5.3	3人暮らしの初期検索クエリ	32
5.4	該当物件数と各実験結果	34
5.5	実験 7,8 の各手法の結果	37

第1章 はじめに

近年、アットホーム*やSUUMO†など多数の企業が賃貸物件検索サイトを運営し、多くの利用者に活用されている。現在の物件検索システムは、探したい地域や最寄の沿線、駅などを選択し、家賃や間取り、駅までの徒歩の時間など希望物件の条件を入力し、該当物件数を絞り込むことが主流となっている。検索したい物件の場所が決まっていれば、手軽に多くの物件を検索・閲覧することができる。しかし、初期条件で適切な条件が入力できない場合、初期条件と適合する該当物件数が非常に多く、利用者に不必要な物件も比例して多く表示されてしまう。そのため、利用者にとって必要な物件を見つけるには困難であり、労力が必要となる。初期条件で適切な条件が入力できない例として、探したい市と家賃、間取りのみを入力することが挙げられる。こういった状況は、特に引っ越し先の地理情報を把握していない利用者に多くいると考えられる。本研究は、適切な条件を入力できない場合でも、利用者が少ない労力で賃貸物件検索を行うことができる賃貸物件検索 Web アプリケーションの提案を行う。

賃貸物件検索は情報検索・情報推薦の分野に属する。この分野では、昔から多くの研究がなされ、多くの手法が提案されている。本研究では、その中でも適合性フィードバック [1] を対象とする。適合性フィードバックを用いることにより、利用者との対話を通して希望する物件条件を推定できないか考えた。適合性フィードバックの主な手法として Rocchio の式 [1] が適用されている。Rocchio の式を用いて事前実験を行ったが、設定した条件に適合する物件は表示されなかった。適切な条件を入力していないため初期検索の表示物件の 8~9 割が不適合であり、検索クエリを更新しても大きく検索クエリが更新されない。そのため、検索クエリと物件データの類似度が変化しないため表示物件も変化せず、設定した条件に適合する物件を絞ることができなかった。

そこで、利用者に適合する物件を探す際に利用者からのフィードバックを繰り返す部分に着目し、バンディットアルゴリズム [2] に注目した。当たる確率の分からないスロットマシンが複数あり、1 試行で 1 台のスロットマシンしかプレイできな

*<https://www.athome.co.jp/>

†<https://suumo.jp/>

いとす。限られた試行回数の中でスロットマシンから得られる最終的な報酬を最大化するためのアルゴリズムがバンディットアルゴリズムである。バンディットアルゴリズムでは複数のスロットマシンをアーム、プレイしたスロットマシンが当たった際にもらえるものを報酬と定義する。バンディットアルゴリズムには ϵ -greedy 法や Upper Confidence Bound(UCB) [2], Thompson Sampling [3] などの代表的なアルゴリズムが適用されている。

本研究をバンディットアルゴリズムの問題としてモデル化すると、各アームを各賃貸物件、報酬を利用者が行う適合・不適合の判定と置き換えられる。バンディットアルゴリズムでは各アームに関する情報は事前に分からず、各試行で得られるフィードバックの結果のみで最終的な報酬を最大化するアームの探索を行う。しかし、賃貸物件は住所や家賃、築年などの事前情報が分かっている。このままバンディットアルゴリズムを適用しても、事前情報を利用せずにフィードバックの結果のみで表示物件を決めてしまう。そのため、探索を多く行う必要があり、どの物件が利用者に適合するかの推定に時間を要する。そこで、バンディットアルゴリズムの中でも Contextual Bandit に着目した。

Contextual Bandit は、各利用者の年齢や性別、家族構成といった特徴や各賃貸物件の事前情報などの特徴をコンテキストとする。1 試行ごとに結果として受け取ったコンテキストとフィードバックを用いて、事前に仮定した確率分布のパラメータを変化させ、確率分布からアームの推定を行う。Contextual Bandit は、通常のバンディットアルゴリズムより効率的に最大の報酬を得ることができると考えられているバンディットアルゴリズムの一つである。

本研究では、各賃貸物件の事前情報をコンテキストとして扱い、報酬が適合・不適合の 2 値であるため、ロジスティック回帰を用いた Contextual Bandit を適用する。Contextual Bandit には、バンディットアルゴリズムと同様に UCB や Thompson Sampling を用いたアルゴリズムが適用されている。本研究では、精度が一番良い [3] とされている Thompson Sampling を用いた Contextual Bandit を実装した。

実験では、著者が独自に設定した検索クエリと物件選択条件によるシミュレーション実験と、研究室のメンバーに利用者設定を提示し、状況に応じた物件検索を行う利用者実験の 2 種類の実験を行った。評価は、フィードバックされた物件 10

件のうち適合である物件が7件になるまでのフィードバック回数を比較した。実験の目的は、既存手法として Rocchio の式を適用したシステムと比較し、Contextual Bandit を適用したシステムの方がフィードバック回数が少なくなることを確認することである。実験の結果、どちらの実験においても Contextual Bandit を適用したシステムでのフィードバック回数の方が少ない実験条件がいくつか見られた。最もフィードバック回数の改善が見られたのは比較手法では収束せず、提案手法で収束したフィードバックの平均回数が 9.7 回であった実験だった。Contextual Bandit がこのシステムにおいて限られた実験条件ではあるがうまく活用できることが示せた。しかし、Rocchio の式を適用したシステムでのフィードバック回数の方が少ない実験条件もいくつか見られた。具体的には、物件選択条件による適合物件数が多いと考えられる場合に、Contextual Bandit を用いたシステムのフィードバック回数の方が Rocchio の式を適用したシステムでのフィードバック回数より最大で 16.3 回多い結果となった..

本研究の貢献は、適切な条件を入力できない利用者の労力を必要最小限にし、賃貸物件検索ができる賃貸物件検索システムとして Contextual Bandit の適用を提案したことである。これにより、今後の賃貸物件検索システムの改善の一提案になると考えられる。また、適合性フィードバックに Contextual Bandit を適用することは、賃貸物件検索システムだけではなく、商品検索や情報検索などの検索問題への解決策の一つとして考えられる。

本論文の構成は以下の通りである。2 章では本研究で用いた技術について述べる。3 章では関連研究について述べる。4 章では、提案手法の Contextual Bandit を用いた検索 Web アプリケーションとそのシステムについて述べる。5 章ではシミュレーション実験と利用者実験の 2 種類の実験について述べる。最後に 6 章では本論文のまとめと今後の課題について述べる。

第2章 基本的事項

2.1 適合性フィードバック

情報検索の分野において、昔から主な研究対象の一つとして扱われている。適合性フィードバックの流れは以下となる。

- (1) 利用者が検索クエリを文章やキーワードで指定する。
- (2) システムが指定された検索クエリと類似した検索結果を利用者に提示する。
- (3) 利用者が検索結果に対して求めているものであれば適合、求めているものであれば不適合のフィードバックを行う。
- (4) システムでフィードバックを用いて検索クエリの更新を行う。
- (5) 更新した検索クエリでの検索結果を利用者に提示する。

(3)~(5) で利用者とシステムの対話を繰り返すことにより、利用者が求める検索結果の提示を目的とする。

2.2 Rocchio の式

適合性フィードバックの研究で主に用いられている手法の一つである。Rocchio の式は、2.1 節の (4) の検索クエリの更新で用いる。検索クエリを利用者に適合する情報に近づけるように更新を行う。Rocchio の式の検索クエリの更新式は式 (2.2.1) である。ただし、 Q は検索クエリベクトル、 $x^+(x^-)$ は適合 (不適合) データベクトル、 $N^+(N^-)$ は適合 (不適合) データ数、 $X^+(X^-)$ は適合 (不適合) データの集合、 α, β, γ はハイパーパラメータとする。

$$Q_{i+1} = \alpha Q_i + \beta \frac{1}{N^+} \sum_{x^+ \in X^+} x^+ - \gamma \frac{1}{N^-} \sum_{x^- \in X^-} x^- \quad (2.2.1)$$

三つのハイパーパラメータは、それぞれ 0~1 の範囲の値をとる。三つのパラメータの値の大きさによって、検索クエリ、適合データ、不適合データの重みを変更できる。つまり、どれを重要とするかを決定できる。一般的には、元の検索クエリベクトルの値を保ちつつ、検索クエリベクトルを適合データベクトルに近づけるために $\alpha > \beta > \gamma$ となるようにパラメータの値を設定する。

2.3 バンディットアルゴリズム

バンディットアルゴリズムは、広告配信やレコメンドシステムなどで適用されている。広告配信ではある広告のクリック率の最大化を目的とし、レコメンドシステムでは利用者に購入してもらええるアイテムの推薦を目的としている。

例として、当たる確率が分からない複数のスロットマシンがあり、1 試行につき 1 台のスロットマシンのみプレイできるとする。限られた試行回数の中で、スロットマシンから得られる最終的な報酬を最大化するために考えられたアルゴリズムがバンディットアルゴリズムである。ここで報酬とは、スロットマシンをプレイした際に当たって得られるものとする。バンディットアルゴリズムでは、各スロットマシンに関して一切情報が分からない。1 試行ごとにそれまでプレイした際の結果を入力として、次の試行でどのスロットマシンをプレイすることにより最終的な報酬を最大化できるかを考える。

バンディットアルゴリズムでは探索と活用という二つの考えがある。探索では、まだ情報が少ないスロットマシンを多くプレイして情報を得る。活用では、探索により得られた情報を用いて、最終的な報酬を最大化するスロットマシンをプレイする。バンディットアルゴリズムでは、次の試行でどちらの考えを適用するかが重要となる。

バンディットアルゴリズムの主な手法として以下の三つが挙げられる。

- ϵ -greedy 法 [2]

ϵ の確率で探索を行い、 $1 - \epsilon$ の確率で活用を行う。探索では、ランダムにスロットマシンをプレイする。活用では、それまでに最も多くの報酬を獲得したスロットマシンをプレイする。

ϵ の値を大きくすると、探索を多く行うことになり、最も多くの報酬を獲得しているスロットマシンをプレイすることができない。反対に ϵ の値を小さくすると、探索する回数が少ないため、最適ではないスロットマシンを何度もプレイする可能性がある。このように報酬を最大化できるかは ϵ の値によって変わる。そのため、初めの試行では ϵ の値を大きくして探索を多く行い十分探索を行なった後に、 ϵ の値を小さくして活用を多く行うなどの工夫が考えられている。

- Upper Confidence Bound(UCB)[2]

$$u_t^i = \bar{u}_t^i + \sqrt{\frac{2 \log N}{N_t^i}} \quad (2.3.1)$$

u_t^i を t 回目のスロットマシン i の UCB スコア, \bar{u}_t^i を t 回目までにスロットマシン i を引いたことにより得られた報酬の平均, N を試行回数, N_t^i を t 回目までのスロットマシン i を引いた回数とする.

式 (2.3.1) のように 1 試行ごとにそれまでのプレイ結果からそれぞれのスロットマシンについて UCB スコアを求め, 最もスコアの高いスロットマシンをプレイする. プレイ回数の少ないスロットマシンに関する情報は限られている. 式 (2.3.1) では, 情報が少ないスロットマシンのプレイ回数を増やすために補正項 $\sqrt{\frac{2 \log N}{N_t^i}}$ を加えている. これにより, プレイ回数がより少ないスロットマシンは補正項の分母の値が小さく, 補正項全体の値が大きくなるため情報を手に入れられるようにしている. 補正項があることにより試行回数が少ないうちは探索を多く行い, 全てのスロットマシンを満遍なくプレイすると, 過去の報酬の平均 \bar{u}_t^i が最も高いスロットマシンをプレイする活用を行う.

- Thompson Sampling[3][4]

報酬がある確率分布から生成されると仮定する. 1 試行ごとにそれまでのプレイ結果によって事前に仮定した確率分布が変化すると考え, 新たな確率分布を求める. 求めた確率分布からランダムにサンプリングを行い, 最も大きいサンプリング値であったスロットマシンをプレイする.

試行回数が少ないうちは, プレイしていないスロットマシンの当たる期待値はすべて図 2.1 のような同じ確率分布に従うため, 探索を行う. プレイ回数を重ね当たった数の多いスロットマシンの確率分布は, 図 2.2 のように当たる確率の高い方に変化する. 外れた回数の多いスロットマシンの確率分布は, 図 2.3 のように当たる確率の低い方に変化する. よって, プレイ回数を重ねると, 期待値の高い確率分布をもつスロットマシンをプレイする活用を行う. 変化する確率分布の計算については, 以下で詳しく述べる.

今まで述べたのは, バンディットアルゴリズムの理論的な内容である. 以下から

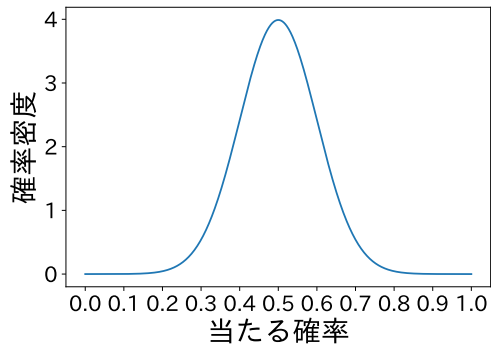


図 2.1 プレイしていないスロットマシンの確率分布

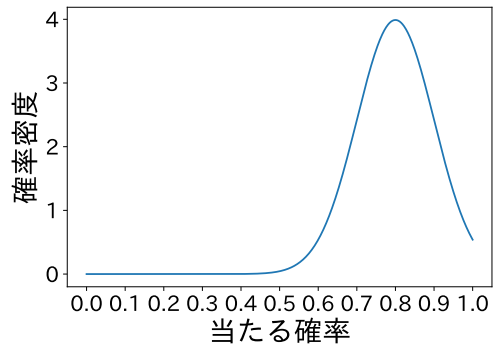


図 2.2 当たった回数の多いスロットマシンの確率分布

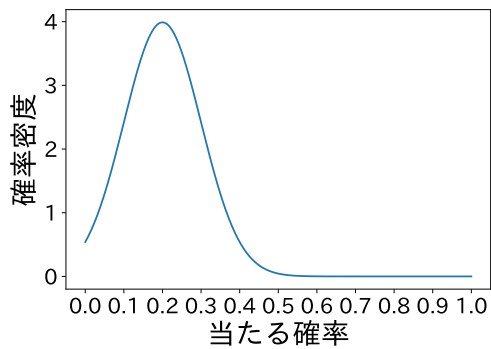


図 2.3 外れた回数の多いスロットマシンの確率分布

は、現実世界の問題に対してどのようにバンディットアルゴリズムが適用されているかを述べる。

例として広告配信への適用について述べる。冒頭でも挙げたように、広告配信の目的は特定の広告のクリック率を最大にすることである。クリック率を最大化することにより、コンバージョン数を増加させることや利用者のデータをより多く収集できるなど多くのメリットがある。コンバージョン数を増加させることにより、広

告を配信している EC サイトや企業の直接的な利益を上げることにつながる。コンバージョンとは、商品購入や商品に関する問い合わせなど広告を配信する企業が目的としている行動を利用者が行うことである。利用者のデータを収集し、どの利用者層に多くクリックされているかを集計する。集計結果から、ターゲットの利用者により多くクリックしてもらうために広告のデザインやフォントなどの改善を行い、広告の品質を向上させることができる。

配信する広告のクリック率を上げるために、同じ商品の広告でデザインやフォントなどを変えたものを複数準備する。今回は Thompson Sampling を用いて広告配信の最適化を行うとする。Thompson Sampling では、報酬が生成される確率分布にはガウス分布やベルヌーイ分布などが用いられる。広告配信の例では、報酬がクリックするかしないかの 2 値であるので報酬の分布をベルヌーイ分布とする。ベルヌーイ分布の事前分布は、事後分布が解析的に計算できるようにベータ分布とする。ベータ分布 $\text{Beta}(\alpha|\beta)$ とは、確率 θ を用いて以下の式により表される確率分布のことである [4]。ただし、 α , β はハイパーパラメータ、 Γ はガンマ関数とする。

$$p(\theta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

事前分布は仮定した確率分布であり、事後分布はクリックの有無により事前分布から変化する確率分布である。配信する各広告を $a_1, a_2, a_3, \dots, a_k$, i を $1, 2, 3, \dots, k$, 広告 a_i を n_i 回配信した際に得られた累積の報酬を R_{n_i} , 広告 a_i から得られる報酬の期待値を μ_i , 広告 a_i を t 回目に配信した時の報酬 r_{t,a_i} はクリックされたら 1, クリックされなかったら 0 とする。

利用者に広告 a_i を n_i 回配信し、クリックされた回数を m_i 回とするとこれまでに受け取った結果を用いて、広告 a_i の報酬の期待値 μ_i の変化する事後分布をベイズの定理を用いて計算する。

$$\begin{aligned} p(\mu_i | R_{n_i}) &= \frac{p(\mu_i) p(R_{n_i} | \mu_i)}{p(R_{n_i})} \\ &\propto p(\mu_i) \prod_{s=1}^{n_i} p(r_{s,a_i} | \mu_i) \end{aligned}$$

このままでは計算が困難であるため、両辺に対数をとると以下になる。

$$\begin{aligned}
\log p(\mu_i | R_{n_i}) &\propto \log p(\mu_i) + \prod_{s=1}^{n_i} \log p(r_{s,a_i} | \mu_i) \\
&\propto \log \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu_i^{\alpha-1} (1 - \mu_i)^{\beta-1} + \prod_{s=1}^{n_i} \log \mu_i^{r_{s,a_i}} (1 - \mu_i)^{1-r_{s,a_i}} \\
&\propto (\alpha - 1) \log \mu_i + (\beta - 1) \log(1 - \mu_i) + \\
&\sum_{s=1}^{n_i} r_{s,a_i} \log \mu_i + \sum_{s=1}^{n_i} (1 - r_{s,a_i}) \log(1 - \mu_i) + \text{const} \\
&\propto (m_i + \alpha - 1) \log \mu_i + ((n_i - m_i) + \beta - 1) \log(1 - \mu_i) + \text{const} \\
&\propto \text{Beta}(m_i + \alpha | n_i - m_i + \beta)
\end{aligned}$$

よって、求める事後分布は、ベータ分布のハイパーパラメータ α , β にそれぞれクリックされた回数 m_i とクリックされなかった回数 $n_i - m_i$ を加算したベータ分布に従う。次の利用者に配信する広告は、広告 $a_1 \sim a_k$ についてそれぞれ上記の事後分布を求め、事後分布からパラメータをサンプリングし、最も大きいサンプリング値であった広告を配信すれば良い。

2.4 Contextual Bandit

バンディットアルゴリズムでは、各スロットマシンについて事前に情報が分かっておらず、プレイしたスロットマシンの結果のみを用いて次にプレイするスロットマシンを選択していた。Contextual Bandit では、各スロットマシンについて事前に何かしらの情報が分かっているとシコンテキストと定義する。プレイしたスロットマシンの結果とコンテキストを用いて、次にプレイするスロットマシンを選択する。

Contextual Bandit は、バンディットアルゴリズムと同様に UCB や Thomson Sampling を用いたアルゴリズム [3] が適用されている。2.4 節では、本研究で適用する報酬が 2 値である Thomson Sampling を用いた Contextual Bandit について詳しく述べる。

Contextual Bandit の流れは以下である。各アームを $a_1, a_2, a_3, \dots, a_n$, i を $1, 2, 3, \dots, n$ とする。アームとは、バンディットアルゴリズムでのスロットマシン

ンと同様に複数の選択肢を表し、一般化のために今後このように表現する。 t 回目に i 番目のアーム a_i を提示し、利用者やアーム a_i の t 回目のコンテキストを x_{t,a_i} とする。本研究では各アームを各賃貸物件、コンテキストを各賃貸物件の住所や家賃、間取りなどの物件情報とモデル化できる。

- (1) 利用者にアーム a_i を提示する。
- (2) 利用者の特徴や (1) のアーム a_i の特徴を表すコンテキスト x_{t,a_i} と利用者の適合・不適合、 t 回目までの試行で得た結果を用いて、利用者に適合とされるアームを推定するために事後分布を計算する。
- (3) 事後分布を解析的に求めることはできないため、事後分布からニュートン法により MAP 推定量 $\hat{\theta}_t^{MAP}$ を求める。
- (4) (3) により求めた MAP 推定量 $\hat{\theta}_t^{MAP}$ を用いて事後分布を近似し、事後分布からパラメータをランダムにサンプリングする。
- (5) サンプリングしたパラメータを用いて最適なアームを求める。

以下からはこの手順について [5] を参考に詳しく述べる。

2.4.1 アームの提示

試行 t 回目に利用者にアーム a_i を提示し、アーム a_i に対して適合・不適合のフィードバックをしてもらう。アームが物件とすると、利用者は提示された物件が自分の希望物件であれば適合、そうでなければ不適合のフィードバックを行う。

2.4.2 事後分布の計算

t 回目のフィードバックを受けた時のパラメータを θ_t 、 t 回目のフィードバック後の各アームの報酬を r_{t,a_i} とする。 $t+1$ 回目に利用者から適合・不適合であるとフィードバックされる各アームの報酬の確率はそれぞれ以下の式になる。ここでの報酬とは、利用者から各アームに対してフィードバックされる適合・不適合のこと

である。以降、適合、不適合を 1,0 で表現する。

$$P(r_{t,a_i}|\theta_t) = \begin{cases} \frac{e^{\theta_t^T x_{t,a_i}}}{1+e^{\theta_t^T x_{t,a_i}}} & (r_{t,a_i} = 1) \\ \frac{1}{1+e^{\theta_t^T x_{t,a_i}}} & (r_{t,a_i} = 0) \end{cases}$$

利用者に適合とされるアームを求めるには、パラメータ θ_t と各アームのコンテキスト x_{t,a_i} を入力とすると、 $r_{t,a_i} = 1$ となる確率を最大にするコンテキストが分かれば、そのコンテキストのアームが利用者に適合とされるアームである。よって、上記より t 回目のフィードバック後の最適なアームは次のように求められる。

$$a(t) = \arg \max_{a_i} \frac{e^{\theta_t^T x_{t,a_i}}}{1 + e^{\theta_t^T x_{t,a_i}}} = \arg \max_{a_i} \theta_t^T x_{t,a_i} \quad (2.4.1)$$

よって、パラメータ θ_t を求めることで利用者に適合とされるアームを推定することができる。パラメータ θ_t を求めるには、報酬の確率の事後分布を求める必要がある。

ここで、パラメータ θ_t と事後分布について説明する。事後分布とは、ある回数までの結果を元に報酬の確率からベイズの定理を用いて導き出される確率分布のことである。 t 回目までの事後分布を求める式は以下である。

$$p(\theta_t | \{r_{s,a_i}\}_{s=1}^t) = \frac{p(\theta_t) \prod_{s=1}^t p(r_{s,a_i} | \theta_s)}{\prod_{s=1}^t p(r_{s,a_i})}$$

パラメータ θ_t とは、上記の求めた事後分布からランダムにサンプリングした値である。

実際にパラメータ θ_t を求めていく。先ほど述べた報酬の確率を用いて、ベイズの定理から事後確率最大化推定 (MAP 推定) を行い、事後分布を求める。ただし、 σ はハイパーパラメータ、パラメータ θ_t の事前分布を $N(0, \sigma^2)$ とする。 $p(\theta_t)$ が事前分布、 $\prod_{s=1}^t p(r_{s,a_i} | \theta_s)$ が、 t 回目までのフィードバックを受けた物件の報酬の確率である。

$$\begin{aligned}
p(\theta_t | \{r_{s,a_i}\}_{s=1}^t) &= \frac{p(\theta_t) \prod_{s=1}^t p(r_{s,a_i} | \theta_s)}{\prod_{s=1}^t p(r_{s,a_i})} \\
&\propto p(\theta_t) \prod_{s=1}^t p(r_{s,a_i} | \theta_s) \\
&\propto p(\theta_t) \prod_{s=1}^t \left(\frac{e^{\theta_s^T x_{s,a_i}}}{1 + e^{\theta_s^T x_{s,a_i}}} \right)^{r_{s,a_i}} \left(\frac{1}{1 + e^{\theta_s^T x_{s,a_i}}} \right)^{1-r_{s,a_i}} \\
&\propto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\theta_t^T \theta_t}{2\sigma^2}} \prod_{s=1}^t \frac{1}{1 + e^{\theta_s^T x_{s,a_i}}} \prod_{s:r_{s,a_i}=1} e^{\theta_s^T x_{s,a_i}}
\end{aligned}$$

上記の式を θ_t の関数として最大化することにより推定を行う。このままでは計算が困難であるため、両辺に負の対数をとると以下になる。式 (2.4.2) より、 t 回目までの全てのフィードバックの結果を用いて事後分布の推定を行う。

$$\begin{aligned}
-\log p(\theta_t | \{r_{s,a_i}\}_{s=1}^t) &= \frac{\theta_t^T \theta_t}{2\sigma^2} + \sum_{s=1}^t \log(1 + e^{\theta_s^T x_{s,a_i}}) \\
&\quad - \sum_{s:r_{s,a_i}=1} \theta_s^T x_{s,a_i} + \text{const} \quad (2.4.2)
\end{aligned}$$

2.4.3 MAP 推定量 $\hat{\theta}_t^{MAP}$ を求める

式 (2.4.2) より、 θ_t についての 1 次微分、2 次微分を求めると以下になる。ただし、 I_d は単位行列とする。

$$\begin{aligned}
G(\theta_t) &= \frac{\theta_t}{\sigma^2} + \sum_{s=1}^t \frac{e^{\theta_s^T x_{s,a_i}} x_{s,a_i}}{1 + e^{\theta_s^T x_{s,a_i}}} - \sum_{s:r_{s,a_i}=1} x_{s,a_i} \\
H(\theta_t) &= \frac{I_d}{\sigma^2} + \sum_{s=1}^t \frac{e^{\theta_s^T x_{s,a_i}} x_{s,a_i} x_{s,a_i}^T}{(1 + e^{\theta_s^T x_{s,a_i}})^2}
\end{aligned}$$

1 次微分を求めたので、 $G(\theta_t) = 0$ を満たす MAP 推定量 $\hat{\theta}_t^{MAP}$ が求まればそのまま θ_t となる。しかし、 $G(\theta_t) = 0$ を満たす $\hat{\theta}_t^{MAP}$ を解析的に求めることはできない。そこで、反復を繰り返すニュートン法により $\hat{\theta}_t^{MAP}$ を数値的に求める。

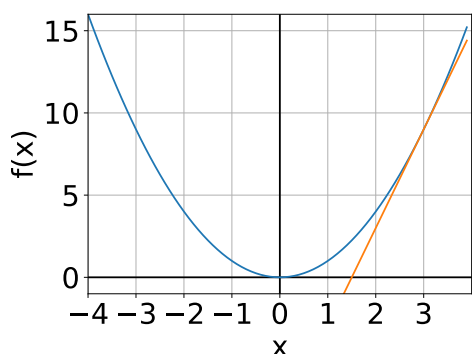


図 2.4 $f(x)$ と接線 1

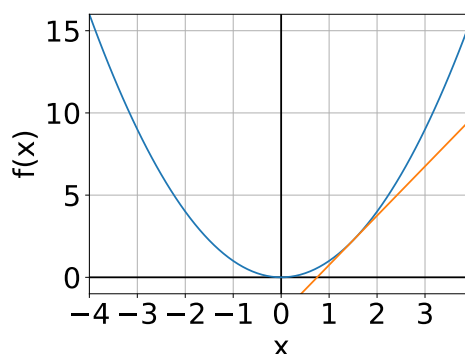


図 2.5 $f(x)$ と接線 2

ここで、ニュートン法について述べる。図 2.4,2.5 の 2 次関数のグラフを $f(x)$ とする。この 2 次関数の $f(x) = 0$ を満たす x を求めたい。一般的には、 $f(x)$ を微分して $f'(x)$ を求め、 $f'(x) = 0$ を計算することにより $f(x) = 0$ を満たす x を求めることができる。これが解析的に $f(x) = 0$ を満たす x を求める方法であるが、今回は解析的に求められないため数値的に求める。ニュートン法の考え方は、ある点で引いた接線の x 軸の切片が、接点の x より $f(x) = 0$ を満たす x に近づくというものである。

図 2.4 のように $x = 3$ で接線を引くとする。接線の x 軸の切片は $x = 1.5$ であるので、 $f(x) = 0$ を満たす x に近づいている。 $f'(x)$ は接線の傾きを求めており、直線の傾きは 2 点間の x の増加量で y の増加量を割ったものであるので、以下の関係式が成り立つ。

$$f'(3) = \frac{f(3) - 0}{3 - 1.5} \Leftrightarrow 1.5 = 3 - \frac{f(3)}{f'(3)} \quad (2.4.3)$$

図 2.5 のように、次に先ほどの接線の x 軸の切片である $x = 1.5$ で接線を引く。式 (2.4.3) と同様に計算をすると以下になる。

$$f'(1.5) = \frac{f(1.5) - 0}{1.5 - 0.75} \Leftrightarrow 0.75 = 1.5 - \frac{f(1.5)}{f'(1.5)} \quad (2.4.4)$$

式 (2.4.4) より次は $x = 0.75$ で接線を引いて同様に計算を行えば良い。この計算

を繰り返すことにより, $f(x) = 0$ を満たす x を近似的に求めることができる. 式 (2.4.3),(2.4.4) を一般化すると式 (2.4.5) になる.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.4.5)$$

よって, $G(\theta_t) = 0$ を満たす MAP 推定量 $\hat{\theta}_t^{MAP}$ を求めるために, 式 (2.4.5) を当てはめると以下になる.

$$\hat{\theta}_t(j+1) \leftarrow \hat{\theta}_t(j) - \frac{G(\hat{\theta}_t(j))}{H(\hat{\theta}_t(j))}$$

この更新をどれだけ繰り返すかは, 状況に適した回数を選択することやある閾値を設定して, 更新した値がその閾値よりも小さくなった時点で更新を止めるなどによって決定する.

2.4.4 事後分布の近似

ニュートン法により求めた MAP 推定量 $\hat{\theta}_t^{MAP}$ のまわりで式 (2.4.2) の 2 次近似を行う. $G(\hat{\theta}_t^{MAP}) \approx 0$ であるので, テイラー展開を利用すると以下になる.

$$\begin{aligned} -\log p(\theta_t | \{r_{s,a_i}\}_{s=1}^t) &\approx \\ \frac{1}{2}(\theta_t - \hat{\theta}_t^{MAP})^T H(\hat{\theta}_t^{MAP})(\theta_t - \hat{\theta}_t^{MAP}) &+ \frac{(\hat{\theta}_t^{MAP})^T \hat{\theta}_t^{MAP}}{2\sigma^2} + \\ \sum_{s=1}^t \log(1 + e^{(\hat{\theta}_t^{MAP})^T x_{s,a_i}}) - \sum_{s:r_{s,a_i}=1} &(\hat{\theta}_t^{MAP})^T x_{s,a_i} + const \end{aligned} \quad (2.4.6)$$

ここで, 多変量ガウス分布の確率密度関数に負の対数をとると式 (2.4.7) になる.

$$\begin{aligned} -\log \frac{1}{(\sqrt{2\pi})^n \sqrt{|\Sigma|}} \exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)) &= \\ \frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) &+ const \end{aligned} \quad (2.4.7)$$

式 (2.4.6) の一項目と式 (2.4.7) を比較すると同じような形をしている. よって, 求めたい事後分布を多変量ガウス分布に近似する. 二つの式を比較すると, マップ推定量 $\hat{\theta}_t^{MAP}$ を平均 μ , 2 次微分の逆行列 $(H(\hat{\theta}_t^{MAP}))^{-1}$ を分散共分散行列 Σ

とみなすことができる。よって、多変量ガウス分布 $MultiNorm(\mu, \Sigma)$ を用いて、 $MultiNorm(\hat{\theta}_t^{MAP}, (H(\hat{\theta}_t^{MAP}))^{-1})$ とすることにより近似でき、事後分布を求めることができる。

2.4.5 最適なアームの計算

近似した事後分布からパラメータ θ_t をランダムにサンプリングする。パラメータ θ_t が求まったので、式 (2.4.1) より最適なアームを求めることができる。そして、アームを利用者に提示する (1) に戻る。

(1)~(5) までの流れを繰り返すことにより、利用者に適合とされるアームを推定する。

2.5 Bag of Words

自然言語処理では、機械学習や類似度計算などをする際に自然言語のままデータを用いることはできない。そこで、Bag of Words により自然言語で書かれた文をベクトルへ変換を行う。文をベクトル化するためには文を形態素解析により単語ごとに分ける必要がある。文を単語ごとに分けたデータをそれぞれ以下とする。単語の例として、“岐阜 ” や “電車 ” , “私 ” などが挙げられる。

$$x_{q^{(1)}} = (q_1^{(1)}, q_2^{(1)}, \dots, q_n^{(1)}), \quad x_{q^{(2)}} = (q_1^{(2)}, q_2^{(2)}, \dots, q_n^{(2)})$$

例文として、(1) 「私は岐阜県が好きで岐阜県に住む。」と (2) 「私はパンが好き。」を扱う。上記と同様に形態素解析により単語に分けると以下になる。

$$\begin{aligned} x_{(1)} &= (\text{私}, \text{は}, \text{岐阜県}, \text{が}, \text{好き}, \text{で}, \text{に}, \text{住む}), \\ x_{(2)} &= (\text{私}, \text{は}, \text{パン}, \text{が}, \text{好き}) \end{aligned}$$

2.5.1 単語辞書の作成

単語ごとに数値を割り振り辞書を作成する。 $x_{q^{(1)}}, x_{q^{(2)}}$ の単語を用いて単語辞書 d_1 を作成すると以下になる。数値は以下のように単語ごとに識別するために、重複

しないよう順番に割り振る.

$$d_1 = \{ "q_1^{(1)}" : 1, "q_2^{(1)}" : 2, \dots, "q_n^{(1)}" : n, \\ "q_1^{(2)}" : n + 1, "q_2^{(2)}" : n + 2, \dots, "q_n^{(2)}" : 2n \}$$

同様に例文の単語辞書 d_2 を作成すると以下になる.

$$d_2 = \{ "私" : 1, "は" : 2, "岐阜県" : 3, "が" : 4, \\ "好き" : 5, "で" : 6, "に" : 7, "住む" : 8, "パン" : 9 \}$$

2.5.2 各単語のベクトル化

作成した単語辞書を用いて単語ごとに one hot vector を作成する. one hot vector とは, 単語ごとにその単語が存在する要素には 1, それ以外の要素には 0 をもつベクトルのことである. ベクトルの次元数は単語の総数であり, 単語辞書の value がそのまま one hot vector を作成する際の要素番号となる. よって, 単語辞書 d_1 を用いて, 単語ごとの one hot vector は以下になる.

$$v_{q_1^{(1)}} = [1, 0, \dots, 0, 0, 0, \dots, 0], v_{q_2^{(1)}} = [0, 1, \dots, 0, 0, 0, \dots, 0], \dots, \\ v_{q_n^{(1)}} = [0, 0, \dots, 1, 0, 0, \dots, 0], v_{q_1^{(2)}} = [0, 0, \dots, 0, 1, 0, \dots, 0], \\ v_{q_2^{(2)}} = [0, 0, \dots, 0, 0, 1, \dots, 0], \dots, v_{q_n^{(2)}} = [0, 0, \dots, 0, 0, 0, \dots, 1]$$

同様に単語辞書 d_2 を用いて, 例文の単語ごとの one hot vector は以下になる.

$$v_{私} = [1, 0, 0, 0, 0, 0, 0, 0, 0], v_{は} = [0, 1, 0, 0, 0, 0, 0, 0, 0], \\ v_{岐阜県} = [0, 0, 1, 0, 0, 0, 0, 0, 0], v_{が} = [0, 0, 0, 1, 0, 0, 0, 0, 0], \\ v_{好き} = [0, 0, 0, 0, 1, 0, 0, 0, 0], v_{で} = [0, 0, 0, 0, 0, 1, 0, 0, 0], \\ v_{に} = [0, 0, 0, 0, 0, 0, 1, 0, 0], v_{住む} = [0, 0, 0, 0, 0, 0, 0, 1, 0], \\ v_{パン} = [0, 0, 0, 0, 0, 0, 0, 0, 1]$$

2.5.3 各文のベクトル化

一つの文ごとに使用されている単語の one hot vector を足し合わせたものが Bag of Words である。その際に、単語の要素の値はその単語の文中の出現回数が入る。よって、文ごとの Bag of Words は以下になる。

$$v_{q^{(1)}} = [1, 1, \dots, 1, 0, 0, \dots, 0], \quad v_{q^{(2)}} = [0, 0, \dots, 0, 1, 1, \dots, 1] \quad (2.5.1)$$

同様に例文の Bag of Words は以下になる。“岐阜県” は (1) の文中に 2 回出現しているため、要素の値は 2 になる。

$$v_{(1)} = [1, 1, 2, 1, 1, 1, 1, 1, 0], \quad v_{(2)} = [1, 1, 0, 1, 1, 0, 0, 0, 1]$$

第3章 関連研究

本村ら [6] は、物件検索に受容度という概念を導入して賃貸物件検索システムを構築している。例として家賃を取り上げる。家賃の上限を5万円、下限を3万円と設定したとする。利用者が3万円~5万円の間の家賃において、どれほど受け入れられるかを0~1で表す。3万円は0.8、5万円は0.3のように利用者自身に入力させるものが受容度である。これにより、一般的な上限・下限の検索ではなく、その中で利用者の好みの幅を持たせることができる。この研究では、初期条件で提示する駅からの距離、築年数、賃料、面積の4項目に対して上限と下限を入力した後に、各項目の受容度を1次関数を用いて受容度関数として利用者に入力させている。受容度関数は、単に一次関数で表されるような単調増加や単調減少するものではないため、曲線で入力を行う仕様になっている。また、各項目で入力した受容度曲線を乗算または加算することにより、どの項目を重視しているかが分かり、統合受容度を計算することができる。そして、統合受容度が高い順に利用者に物件の提案を行っている。

中野ら [7] は、オンライン機械学習を用いた賃貸物件検索推薦サービスの提案を行っている。一般的に機械学習はバッチ処理により行われるため、大量の学習データを保持するための大規模なデータベースなどの環境が必要であったり、学習に多くの時間が必要なため、学習結果をシステムに反映するのに時間を要する。これらの課題は、リアルタイムにフィードバックを受け推薦を行う Web サービスにとって致命的である。そこで、オンライン機械学習を用いることにより、得られた結果からすぐに学習を行い、学習結果を利用者の推薦へ反映させることができる。この研究では、Jubatus というオンライン機械学習を行うためのミドルウェアを利用してシステムの実装を行っている。システムの流れは、物件が各最寄り駅ごとで分けられた探索空間上に存在し、同じ探索空間上の遠い場所に位置する二つの物件を利用者に提示する。利用者が提示された物件に対してフィードバックを行い、フィードバックの結果を用いて Jubatus の分類器で学習を行う。学習結果から探索空間を徐々に狭めていき、利用者の希望する物件条件を推定し適合物件数を絞り、利用者に物件の推薦を行う。

本研究と関連研究は、フィードバックの逐次処理の点は中野らと同じである。中

野らの提案手法は、利用者が二つの物件のフィードバックを最大8回のみ行うことで、利用者の物件条件と適合物件を絞ることができる。しかし、探索空間は最寄駅ごとで構成されるため、最寄駅を初期条件で入力しなければ検索することができない。また、提示される物件は最寄駅がすべて初期条件で入力した最寄駅の物件のみとなる。こうしたことから、最寄り駅をどこにするか決定している利用者が使用するシステムであると考えられる。一方、本稿の提案手法では、最寄駅を初期条件で入力しない場合にも検索を行うことができる。また、物件を表示する際の制限はないため、利用者にとって意外な物件を提示することができる。しかし、確率的に表示物件を決めているため、物件選択条件によっては複数回フィードバックを行ったとしても利用者に適合する物件が表示されないことがある。これについては、5.4節や5.5節で述べる。

第 4 章 提案手法

4.1 節では実装した検索 Web アプリケーションについて述べる。その後、4.2 節では Contextual Bandit を適用したシステムの流れについて述べる。最後に 4.3 節では物件データをシステム内での計算に利用するために行った前処理について述べる。

4.1 検索 Web アプリケーション

4.1.1 アプリケーションの流れ

賃貸物件検索 Web アプリケーションは、Ruby on Rails* を用いて実装を行った。データベースは SQLite† を用いた。アプリケーションの表示画面は初期検索クエリを入力する画面 1 (図 4.2) と物件を表示し、表示物件に対して利用者に適合・不適合のフィードバックをしてもらう画面 2 (図 4.3) の 2 種類である。画面 1, 2 は表示画面の一部である。

アプリケーションの流れと対応する表示画面は図 4.1 になる。以下に図 4.1 の詳細を述べる。

- (1) 利用者が初期検索クエリを入力し送信する。
- (2) 利用者の入力した初期検索クエリと初期検索クエリで指定した都道府県の全物件データを比較し、一致率を計算して上位 10 件の物件を表示する。
- (3) 表示物件が利用者の希望物件に適合の場合はチェックを付け、不適合の場合はチェックを付けずに送信する。
- (4) それまでの利用者の全フィードバックの結果と住所や家賃、築年などの物件情報を用いて Contextual Bandit を適用したシステムで計算を行う。
- (5) 計算の結果、次に利用者に適合されると推定する物件 10 件を表示する。

(3)~(5) を繰り返すことにより、利用者に適合する物件を絞り込む。

*<https://rubyonrails.org/>

†<https://www.sqlite.org/index.html>

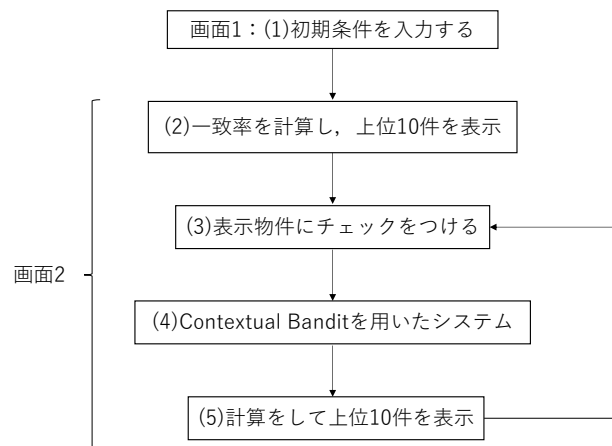


図 4.1 アプリケーションの流れ

都道府県	所在地名1	沿線名称	駅名称	徒歩(分)	バス(分)	停歩(分)	バス停名称	物件価格・賃料(円)	管理費(円)
岐阜県	岐阜市	無	無	0	0	0	無	50000	0

図 4.2 初期検索クエリ入力画面

上記の流れの中で重要となるのが (2) である。初期検索結果のうち、利用者に適合である物件がない場合に、フィードバックを用いて Contextual Bandit を適用したシステムで利用者に適合とされる物件をうまく推定することができなかった。これに関する具体的な結果は 5.4.2 節で述べる。また、4.1.2 節で初期検索結果に利用者に適合とされる物件を表示する方法について述べる。

都道府県	所在地名1	所在地名2	沿線名称	駅名称	徒歩(分)	バス(分)	停歩(分)	バス停名称	物件価格・賃料(円)	管理費(円)	共益費(円)	契約期間(年)	定期借家フラグ	更新料区分	更新料(月数)	築年	築中古区分	所在階	取りタイプ
岐阜県	岐阜市	西玉宮町1丁目	東海道本線	岐阜	6	0	0	無	65000	0	0	3	0	0	0	1997	2	3	1K
岐阜県	岐阜市	加納新本町3丁目	東海道本線	岐阜	8	0	0	無	39000	0	3000	3	0	0	0	1994	2	4	1K
岐阜県	岐阜市	加納新本町3丁目	東海道本線	岐阜	8	0	0	無	39000	0	3000	3	0	0	0	1994	2	5	1K

図 4.3 物件表示画面

4.1.2 検索クエリ入力画面

利用者が入力する検索クエリは表 4.1 である。図 4.2 のように、利用者が検索したい物件に関する条件を入力し送信すると、入力した値がカラムごとにデータベースに保存される。初期検索は、利用者が入力した初期検索クエリと初期検索クエリで入力した都道府県の全ての物件データを比較し、一致率が高い物件 10 件の物件番号を求める。求めた 10 件の物件番号に一致する物件をデータベース内で検索して表示する。図 4.3 は、表 5.2 の実験 3 の初期検索クエリを入力した場合の初期検索結果の一部である。

システムを実装した当初は、初期検索クエリと物件データを 4.3 節で述べる前処理を行い、初期検索クエリと各物件データの cosine 類似度を求め値が大きい物件を表示していた。しかし、例として愛知県一宮市の物件を検索した際に、名古屋市の物件しか表示されないことが見られた。これは県庁所在地以外の住所の物件を探している利用者にとっては不必要な物件である。また、4.1.1 節で述べたように、初期検索結果で利用者に適合する物件が無い場合、利用者に適合とされる物件を推定

するができなかった。そこで、利用者が求める住所の物件を初期検索で表示するために今回はデータの一一致率を用いた。

4.1.3 Contextual Bandit の実装

4.1.1 節で検索 Web アプリケーションを Ruby on Rails を用いて実装したと述べたが、Contextual Bandit を用いて最適な物件を求める部分は Python で実装を

表 4.1 検索クエリ一覧

検索クエリ	備考
都道府県	
所在地名 1	市（政令指定都市は〇区まで）
沿線名称	
駅名称	
徒歩（分）	駅までの徒歩の時間
バス（分）	駅までのバスの時間
停歩（分）	バス停までの徒歩の時間
バス停名称	
物件価格	
管理費	
共益費	
契約期間	
築年	
間取りタイプ	ワンルームや 1K,1LDK など
建物構造	鉄骨造や RC,SRC など
部屋面積（平米）	
駐車場有無	
駐車場料金	

<input type="checkbox"/>	岐阜県	岐阜市	加納本町4丁目	東海道本線	岐阜	7	0	0	無	72000	0	6000	2	0	1	0	2019	1	3	1 S K
<input type="checkbox"/>	岐阜県	岐阜市	八代2丁目	東海道本線	岐阜	0	24	3	福光球場前	33000	0	3000	1	0	1	1	1988	2	3	3 D K
<input checked="" type="checkbox"/>	岐阜県	岐阜市	加納栄町通4丁目	東海道本線	岐阜	7	0	0	無	50000	0	5000	2	0	1	1	2005	2	1	1 K

送信

図 4.4 チェックの有無の送信

行った[‡]。図 4.4 のようにチェックの有無が送信され、チェックの有無と物件番号がデータベースに保存された後に、この二つと利用者が入力した都道府県を引数として、外部コマンドで Python のファイルを Ruby のプログラム内で実行する。計算の結果、最適な物件の物件番号を Ruby 側のプログラムに出力している。

4.1.4 物件表示画面

表示する物件は、データベース内で 4.1.3 節で出力される物件番号を用いて検索を行い、合致したものを選択している。使用したデータセットには、物件の画像や図で描かれた間取りなどの画像データは含まれていないため、図 4.3,4.4 のように物件情報の文字列のみ表示している。図 4.4 のように利用者は各物件の情報を見て、適合であれば物件の横にあるチェックボックスにチェックを入れ、不適合の場合はチェックを入れない。すべてのフィードバックを終えて送信すると、チェックの有無と物件番号を対としてデータベースに保存される。

[‡]https://nbviewer.org/github/hagino3000/notebooks/blob/master/MLP_bandit/Chap7_binary_reward.ipynb

4.2 システムの流れ

2.4 節で一般的な Contextual Bandit の流れについて述べた。Contextual Bandit をシステムに適用するために本研究を Contextual Bandit の問題としてモデル化すると、各アームを各賃貸物件、コンテキストを物件情報とすることができる。2.4 節での流れを元に、提案手法のアルゴリズムは [3][5] を参考に以下となる。ただし、 I_0 は次元数がコンテキストと同じゼロベクトル、 i はハイパーパラメータ、試行を N 回行うとする。アルゴリズム内で用いている変数は 2.4 節で定義している。物件データのコンテキスト x_{t,a_i} は、式 (4.3.3) のように物件データを前処理して用いている。今回行った実験では、 $\sigma = 0.01$ 、 $i = 20$ とした。

Algorithm 1 Contextual bandit with logistic

Input: σ

for $t = 1, 2, 3 \dots N$ **do**

$$\theta_t = I_0$$

{(1)}

repeat

$$G(\theta_t) = \frac{\theta_t}{\sigma^2} + \sum_{s=1}^t \frac{e^{\theta_t^T x_{s,a_i}} x_{s,a_i}}{1 + e^{\theta_t^T x_{s,a_i}}} - \sum_{s:r_{s,a_i}=1}^t x_{s,a_i}$$

$$H(\theta_t) = \frac{I_d}{\sigma^2} + \sum_{s=1}^t \frac{e^{\theta_t^T x_{s,a_i}} x_{s,a_i} x_{s,a_i}^T}{(1 + e^{\theta_t^T x_{s,a_i}})^2}$$

$$\hat{\theta}_t \leftarrow \hat{\theta}_t - H(\hat{\theta}_t)^{-1} G(\hat{\theta}_t)$$

until i

$$\theta_t = \text{MultiNorm}(\hat{\theta}_t, (H(\hat{\theta}_t))^{-1}) \text{ {(2)}}$$

$$\text{Select rent } a = \arg \max_{a_i} \theta_t^T x_{t,a_i} \text{ {(3)}}$$

end for

t 回目のフィードバックを受けると、 t 回目までにフィードバックを受けた物件の適合、不適合の判定 1,0 とその物件のコンテキストのベクトル x_{t,a_i} を入力として、アルゴリズムの (1) で繰り返し $\hat{\theta}_t$ を更新し推定を行う。 $G(\theta_t)$ と $H(\theta_t)$ の 2 項目は、 t 回目までのフィードバックを受けた全ての物件のコンテキストを代入する。 $G(\theta_t)$ の 3 項目は、 t 回目までのフィードバックされた物件のうち、利用者に適合とされた物件のコンテキストのみ代入する。

i 回 $\hat{\theta}_t$ の更新を終えると、アルゴリズムの (2) で事後分布を求めてパラメータ θ_t をサンプリングする。アルゴリズムの (3) で求めた θ_t と各物件のコンテキストのベクトルを用いて、式 (2.4.1) を計算して値の大きい 10 件の物件の物件番号を出力する。

ここからは、表 4.2 の架空の物件データを用いてアルゴリズムの流れを述べる。ここでは $i = 5$ 、利用者に物件が 2 件提示されるとする。

利用者が初期検索クエリを入力し、初期検索結果として物件 1,5 が利用者に提示され、物件 1 には不適合、物件 5 には適合のフィードバックを行ったとする。二つのフィードバックがシステムに送信されるとアルゴリズムの (1) の計算を行う。このフィードバックが 1 回目であるので、(1) では物件 1,5 のコンテキストを用いて $\hat{\theta}_t$ の推定をニュートン法により行う。 $G(\theta_t)$ と $H(\theta_t)$ の 2 項目は、物件 1,5 のコンテキストを代入する。 $G(\theta_t)$ の 3 項目は、適合であった物件 5 のコンテキストのみ代入する。

5 回 $\hat{\theta}_t$ の更新を終えると、アルゴリズムの (2) でパラメータ θ_t の事後分布を求め、 θ_t のサンプリングを行う。物件 1 が不適合、物件 5 が適合であったので、アルゴリズムの (3) で適合とされる物件を推定する計算を行った際に、物件 5 に類似する物件との値が大きくなるように事後分布が近似され、 θ_t のサンプリングが行われる。

よって、物件 1~5 のコンテキストと θ_t を用いて、それぞれ (3) の計算を行う。その結果、値が大きくなるのが物件 2,3 となり、この 2 件の物件が利用者に提示される。

この流れを利用者からフィードバックを受ける度に繰り返し行う。

表 4.2 架空の物件データ

物件	都道府県	所在地名 1	駅まで徒歩	家賃	築年	間取り
1	愛知県	名古屋市北区	3	80000	2010	1K
2	愛知県	春日井市	8	59000	2000	1K
3	愛知県	春日井市	5	65000	2014	1K
4	愛知県	名古屋市北区	15	65000	2007	1K
5	愛知県	春日井市	10	50000	1995	1K

4.3 物件データの前処理

パラメータ θ_t を計算する過程でコンテキストとして物件データの物件情報を用いているが、住所や間取りなどの文字列データと家賃や築年などの数値データが混合している。このままではアルゴリズム内の計算には利用できないため、文字列データと数値データごとに前処理を行いベクトルにする。それぞれ前処理したベクトルを一つのベクトルに結合して物件のコンテキストとする。

“岐阜市” や “東海道本線”, “1K” などの文字列データは Bag of Words によってベクトル化を行い、最終的なベクトルは式 (2.5.1) のようになる。物件情報は上記で挙げたように単語ごとに分かれているため、形態素解析をする必要はなかった。各物件の文字列データの中で県名はすべて同じであり、市や区以下の住所のベクトル化を行うと各物件情報のベクトルの次元数が大きくなり計算コストがかかる。また、適切な条件が入力できないことを考えると市や区以下の住所を入力しないと考えた。これについては、5.4.1 節で詳しく述べる。そこで、この 2 項目は削除した上でベクトル化を行った。

数値データは家賃や築年、駅までの徒歩の時間などで最大値・最小値に大きく差があるため、各都道府県の物件データのカラムごとに正規化を行った。物件データ $q^{(1)}$ の数値データ $x_{q^{(1)}}$ と、物件データ $q^{(1)}$ と同じ都道府県の全データの中で、データ $x_{q^{(1)}}^1$ と同じカラムのデータ X^1 を以下とする。

$$\begin{aligned}x_{q^{(1)}} &= [x_{q^{(1)}}^1, x_{q^{(1)}}^2, \dots, x_{q^{(1)}}^n] \\X^1 &= [x_{q^{(1)}}^1, x_{q^{(2)}}^1, \dots, x_{q^{(N)}}^1]\end{aligned}$$

データ $x_{q^{(1)}}^1$ の正規化は以下の式により行った。ただし、 $x_{q^{(1)}}^{1'}$ は $x_{q^{(1)}}^1$ を正規化した値である。

$$x_{q^{(1)}}^{1'} = \frac{x_{q^{(1)}}^1 - \min(X)}{\max(X) - \min(X)} \quad (4.3.1)$$

同様にそれぞれのカラムごとに正規化を行うと、 $x_{q^{(1)}}^1$ を正規化した $x_{q^{(1)}}^{1'}$ は以下になる。

$$x_{q^{(1)}}' = [x_{q^{(1)}}^{1'}, x_{q^{(1)}}^{2'}, \dots, x_{q^{(1)}}^{n'}] \quad (4.3.2)$$

t 回目の試行の物件データ $q^{(1)}$ において、正規化を行った式 (4.3.2) と式 (2.5.1) のように Bag of Words によりベクトル化を行った文字列のベクトル $v_{q^{(1)}}$ を結合して、以下のようにコンテキストのベクトル $x_{t,q^{(1)}}$ とした。ただし、 $y_{q^{(1)}}^i$ は単語の出現回数、 i は単語辞書に登録されている単語の value、物件データの総単語数が $2k$ 個あるとする。

$$x_{t,q^{(1)}} = [y_{q^{(1)}}^1, y_{q^{(1)}}^2, \dots, y_{q^{(1)}}^k, y_{q^{(1)}}^{k+1}, y_{q^{(1)}}^{k+2}, \dots, y_{q^{(1)}}^{2k}, x_{q^{(1)}}^{1'}, x_{q^{(1)}}^{2'}, \dots, x_{q^{(1)}}^{n'}] \quad (4.3.3)$$

第5章 評価実験

5.1 節では実験で使用したデータについて、5.2 節では実験を行うための比較手法について述べる。5.3 節では、実験の評価方法について述べる。5.4 節では著者が独自に決めた初期検索クエリと物件選択条件による実験について、5.5 節では実装したアプリケーションで研究室のメンバーに物件検索を行ってもらった実験について述べる。

今回の実験は、既存手法として Rochhio の式を適用したシステムと比較し、Contextual Bandit を適用したシステムの方がフィードバック回数が少なくなることを確認するために行った。フィードバック回数が少ないということは、利用者の物件を探す負担を減らすことができていることを示す。

実験は、シミュレーション実験と利用者実験の2種類を行った。シミュレーション実験では、一人暮らしと3人暮らしの利用者を想定して初期検索クエリと物件選択条件を設定した。利用者の状況を複数準備することで、提案手法の方がフィードバック回数が少なくなる状況が何であるかを見るために実験を行った。利用者実験では、利用者の状況をこちらで提示し、実際の利用者による賃貸物件検索を想定して実験を行った。シミュレーション実験では物件選択条件の変更を途中で行っていないが、実際の物件検索において必ずしも条件に100%適合する物件が表示されることはなく、条件を変更しないことはあり得ないと考えられる。この実験では、利用者の途中の条件変更を可能とし、最初に設定した条件に適合しない物件ばかりが表示された場合に、利用者が条件変更後に、提案手法がその条件に適合する物件を提示することができるか確認した。

5.1 実験データ

アットホーム株式会社が国立情報学研究所を通じて提供しているアットホームデータセット*を使用した。このデータセットには、アットホームが運営する全国不動産情報ネットワークに2015年1月1日から2019年12月31日の期間に登録

*アットホーム株式会社 (2020): アットホームデータセット. 国立情報学研究所情報学研究データリポジトリ. <https://www.nii.ac.jp/dsc/idr/athome/>

された賃貸物件や売買物件などのデータが記録されている。実験データには、2019年1月1日から2019年12月31日の1年間に登録されたデータを利用した。賃貸物件数は全国で約491万件であった。一つの賃貸物件には、住所や家賃、築年、間取り、駅までの徒歩の時間など106のカラムがある。

実際には物件番号が重複しているデータや、利用者が賃貸物件を選択する際に必要ないと考えられる土地権利や都市計画、築月などの80カラムを削除し、カラム数26、全国で約35万件のデータを用いて実験を行った。

5.2 比較手法

1章で述べた適合性フィードバックの既存手法である Rocchio の式を比較手法として用いた。

Rocchio の式の検索クエリの更新式は式 (2.2.1) である。式 (2.2.1) では、検索クエリと物件データをベクトル化して検索クエリの更新を行う。物件データのベクトル化は、4.3 節で述べたようにベクトル化を行った。本稿では、 Q_0 を利用者が初期検索時に入力した初期検索クエリとする。初期検索クエリについてもベクトル化を行う必要がある。文字列のベクトル化は、4.3 節で作成した単語辞書を用いて Bag of Words によってベクトル化を行った。数値の正規化については、初期検索クエリの都道府県と合致する都道府県のデータのカラムごとに求めた最大値・最小値を利用して式 (4.3.1) を用いて正規化を行った。最終的に、式 (4.3.3) のように文字列のベクトルと正規化した数値データのベクトルを結合し、初期検索クエリのベクトルとした。

利用者に提示する物件は、更新した検索クエリベクトル Q と物件データベクトル x_a のコサイン類似度を求め、類似度の大きい上位 10 件とした。コサイン類似度を求める式は以下である。

$$\frac{Q^T x_a}{|Q||x_a|}$$

今回の実験では、パラメータの値を経験的に一番良かった $\alpha = 1$, $\beta = 0.3$, $\gamma = 0.1$ として実験を行った。

5.3 評価方法

原島ら [8] は、適合性フィードバック前のランキングと適合性フィードバック後のランキングを比較して手法の評価を行っている。ランキング精度を評価するために Precision@K(P@K) や Mean Average Precision(MAP) などの方法を利用している。P@K はランキング上位 K 個の中で、利用者が適合であるとしたものの割合を求める。MAP は、適合であるものの順位で先ほどの P@K を算出し、不適合であるものの順位の P@K は 0 とする。求めた P@K の和を適合数で割った平均で評価を行う。よって表 5.1 の MAP は、 $(1 + \frac{2}{4}) \div 2 = 0.75$ となる。

本研究は、いかに利用者の物件を探す負担を減らすことができるかを目的としている。P@K は主に一度のフィードバックでのランキングの精度を評価するものであり、MAP はランキングの評価を行うため適切ではないと考えた。そこで、評価方法は利用者がフィードバックを行った回数とした。この評価方法であれば、フィードバック回数が少ないほど、利用者が希望物件を探す負担が少ないと評価することができる。

具体的な評価方法は、利用者が最初に適合である物件をフィードバックした試行からフィードバックの回数を計測した。フィードバックされた物件 10 件のうち、適合である物件が 7 件以上を規定物件数として実験終了とし、これを収束すると定義した。収束するまでのフィードバック回数を評価とする。すべての実験において必ず規定物件数に達することはなかったため、フィードバック回数が 30 回になった場合はそこで実験終了とし、適合物件数が足りないので収束しないと定義し評価とした。

表 5.1 MAP の計算例

ランキング	商品	適合・不適合	P@K
1 位	a	適合	$\frac{1}{1}$
2 位	b	不適合	0
3 位	c	不適合	0
4 位	d	適合	$\frac{2}{4}$
5 位	e	不適合	0

5.4 シミュレーション実験

5.4.1 実験設定

一人暮らしと家族3人暮らしの2パターンを想定し、初期検索クエリと物件選択条件を設定した。探す物件の住所は、物件が多数の場所と少数の場所で評価が変化するかを確認するために、愛知県名古屋市北区、愛知県春日井市、岐阜県岐阜市の3パターンを想定した。総物件データ数は、愛知県が42,533、岐阜県が2,346であった。それぞれの初期検索クエリと物件選択条件は表5.2,5.3の通りである。

ここで初期検索クエリの決定方法について述べる。本研究で対象としているのは、初期検索で適切な条件で入力できない場合である。ここでの適切でないとは、初期条件が必要最小限であることを意味する。そこで、必要最小限の初期条件を求めるために、初期検索クエリの入力するカラムを変更して初期検索の事前実験を行った。その結果、駅名称を入力すると該当物件数が絞られると判明した。同じ市や区の物件でも、駅名称によって場所が限定されると考えられる。同様の理由で市や区以下の住所に関しても入力を行わなかった。

一人暮らしの未入力の数値データに関しては、駅までの徒歩、駅までのバス、バス停までの徒歩、管理費、共益費、所在階、部屋の面積、緯度、経度を各都道府県

表 5.2 一人暮らしの初期検索クエリ

	都道府県	所在地名 1	家賃	契約期間	築年	間取り	駐車場
1	愛知県	名古屋市北区	50000	2	2000	1K	無
2	愛知県	春日井市	50000	2	2000	1K	無
3	岐阜県	岐阜市	50000	2	2000	1K	無

表 5.3 3人暮らしの初期検索クエリ

	都道府県	所在地名 1	家賃	契約期間	築年	間取り	駐車場
4	愛知県	名古屋市北区	80000	2	2000	1K	有
5	愛知県	春日井市	59000	2	2000	1K	有
6	岐阜県	岐阜市	61000	2	2000	1K	有

のカラムごとでデータの平均値を求め、それを代入した。

3人暮らしの未入力の数値データに関しては、駅までの徒歩、駅までのバス、バス停までの徒歩、管理費、共益費、所在階、部屋の面積、駐車場料金、緯度、経度を各都道府県のカラムごとでデータの平均値を求め、それを代入した。また、物件価格に関しては、各所在地名1の全2LDKの物件で物件価格の平均値を求め、初期検索クエリとして利用した。

それぞれの実験での物件選択条件は以下になる。

- 実験 1

駅までの徒歩：15分以内

支払う金額：合計 50000~55000 円以内

築年：1981 年以降

部屋の広さ：20 平米以上

- 実験 2,3

駅までの徒歩：15分以内

支払う金額：合計 55000 円以内

築年：1981 年以降

部屋の広さ：20 平米以上

- 実験 4

駅までの徒歩：20分以内 or 駅までのバス：10分以内

支払う金額：合計 90000 円以内

築年：1981 年以降

部屋の広さ：56 平米以上

駐車場有無：有 or 無

- 実験 5

駅までの徒歩：20分以内

支払う金額：合計 69000 円以内

築年：1981 年以降

部屋の広さ：56 平米以上

駐車場有無：有 or 無

- 実験 6

駅までの徒歩：20 分以内 or 駅までのバス：20 分以内
 支払う金額：合計 71000 円以内
 築年：1981 年以降
 部屋の広さ：56 平米以上
 駐車場有無：有 or 無

物件選択条件は該当物件数が 30~40 個になるよう設定した。実験 1~6 の該当物件数は表 5.4 の通りである。

5.4.2 結果

比較手法は、表示物件が変化しなければ検索クエリベクトルも変化しないため、実験は一回のみ行った。しかし、提案手法は、パラメータ θ_t をランダムにサンプリングしているため、表示物件が同じだとしてもパラメータ θ_t が異なることにより、次の表示物件も異なる。そこで、提案手法の実験は 50 回試行を行い、収束した実験のフィードバック (FB) 回数の平均値で評価を行った。各手法の結果は表 5.4 である。

表 5.4 から今回の実験設定において、実験 3 以外で比較手法より提案手法の方がフィードバック回数が少ない結果となった。

表 5.4 該当物件数と各実験結果

実験	物件数	比較手法	提案手法	
		FB 回数	FB 回数	収束しなかった試行
1	33	収束せず	22.3	46
2	39	収束せず	12.1	16
3	33	3	10.2	0
4	30	収束せず	15.8	46
5	33	収束せず	18.8	29
6	41	収束せず	19.0	20

比較手法は、実験 3 以外収束しなかった。検索クエリベクトルを更新し、それぞれの物件データベクトルとコサイン類似度を求めた際に、毎回表示物件が変化していなかった。これは検索クエリベクトルが既に理想の検索クエリベクトルに近づいており、更新を行ってもあまり検索クエリベクトルが変化しないからではないかと考えられる。

一方、提案手法では、実験 1 と実験 4 では収束しなかった回数が 50 回中 46 回とほぼ収束することはなかった。

実験 1 では、該当物件数を他の実験と揃えるために物件選択条件の合計金額に上限だけでなく下限も設定している。実験 2,3 とはこの条件だけ違うため、フィードバック回数・収束しなかった試行がともに実験 2,3 と比べ多くなったと考えられる。上限と下限の両方を設定することが提案手法では難しいのではないかと考えられる。

実験 2 はシミュレーション実験の中では最もフィードバック回数が改善された。具体的には、比較手法では収束せず、提案手法では 12.1 回となった。

実験 3 では、提案手法が比較手法よりフィードバック回数が多くなったが、他の実験と比較すると収束しなかった試行が 0 個であった。岐阜県の全物件数は 2,346 と愛知県と比較すると少なく、探索する物件数自体が少ないためすべて収束したと考えられる。

実験 4 では、物件選択条件に駅までの徒歩とバスの時間のどちらかで当てはまる物件が含まれるようにした。一方、実験 5 では、実験 1~3 と同様に駅までの徒歩の時間のみを物件選択条件に設定している。その条件の違いで、同じ愛知県内の物件から該当物件を探索しているが、収束した回数に差が出たと考えられる。また、実験 4 に関しては、実験の都合上、初期検索で物件選択条件に適合する物件が表示されず、すべて不適合であった。Contextual Bandit は適合・不適合の結果両方を得ることで多数の選択肢から絞り込むため、適合の物件が表示されるまでに多くのフィードバックを要し、収束する試行が実験 5,6 と比べて少なかったと考えられる。

実験 6 は実験 4 と同様の物件選択条件にしているため実験 3 と比べ、該当物件数が多いにも関わらず、収束した回数が少なかったと考えられる。

実験 1~6 のフィードバック回数を比較すると、家族 3 人暮らしより一人暮らしの利用者に対して提案手法がフィードバック回数 15 回以内に利用者に適合である

物件を提示できると考えられる。

5.5 利用者実験

5.5.1 実験設定

実験設定と検索クエリの一部を以下のようにあらかじめ指定し、研究室のメンバー 5 名ずつに物件検索を行ってもらった。検索クエリは、指定したカラム以外は利用者に自由に決めてもらい実験を行ってもらった。指定した実験設定と検索クエリは以下の通りである。

- 実験 7：岐阜駅から徒歩 5～10 分圏内の大学に入学し、一人暮らしを始めるとする。

都道府県：岐阜県

所在地名 1: 岐阜市

契約期間：2 年

築年：2000 年

- 実験 8：就職して、名古屋駅付近で仕事をするようになった。

都道府県：愛知県

契約期間：2 年

築年：2000 年

5.5.2 結果

実験回数は、比較手法を実験 1～6 と同様に一回のみ行い、提案手法を 10 回試行してもらった。フィードバック回数は、収束した実験のフィードバック回数の平均で評価を行った。それぞれの手法の結果は表 5.5 である。

どちらの実験においても、比較手法の方が提案手法よりフィードバック回数が少ない利用者がある結果となった。これらの利用者は、物件選択条件が支払う合計金額と駅までの徒歩の時間のみを設定していることが多く、条件が厳しくなく該当物件数が多いため比較手法の方がフィードバック回数が少なくなったと考えられる。

提案手法と比較手法のフィードバック回数を比較すると、比較手法で5回、提案手法で21.3回の16.3回が最も差があった。

一方、提案手法の方がフィードバック回数が少なくなった利用者は、物件選択条件が複雑であった。例としては、所在階が1or2階であったり、駐車場のある物件が良いといった物件選択条件にしている利用者が見られた。最もフィードバック回数を改善できた実験は、比較手法で収束せず、提案手法で9.7回となった実験であった。

また、実験7,8を比較すると、全ての利用者の収束しなかった試行の総数は同数であるが、若干実験8の方が提案手法のフィードバック回数が少ない。実験8では、愛知県の物件の中から希望物件を検索している。岐阜県の物件数と比較すると約18倍愛知県の物件数の方が多い。そのため、提案手法はより多くの物件の中から利用者に適合する物件を提示することの方が少ないフィードバック回数で行えると考えられる。

提案手法では、表示物件のフィードバックをしてから次の物件を表示するまでに約30秒～1分かかった。同様のことがシミュレーション実験でも見られた。4.3節で各物件情報のベクトルの次元数が大きくなり計算コストがかかることを防ぐため

表 5.5 実験 7,8 の各手法の結果

実験	利用者	比較手法	提案手法	
		FB 回数	FB 回数	収束しなかった試行
7	1	9	17.1	2
	2	収束せず	9.7	4
	3	2	9.9	1
	4	2	9.7	1
	5	5	21.3	3
8	6	11	9.8	1
	7	収束せず	11.8	4
	8	13	12.1	1
	9	1	12.0	4
	10	1	14.4	1

に、市や区以下の住所のベクトル化を行わなかったと述べた。しかし、それでもアルゴリズム内での計算コストが大きかったとみられる。特に愛知県の物件データは42,533個あるため、ベクトル化した際に一つのベクトルの次元数が1,081次元と大きくなった。そのため、4.2節のアルゴリズム内の $H(\theta_t)$ は $1,081 \times 1,081$ の行列となる。事後分布の近似をする際に $H(\theta_t)$ の逆行列を求める必要があり、この部分の計算コストが大きいと考えられる。これにより、次の物件表示に約1分ほどの時間がかかると考えられる。これは検索システムとしては致命的であり、改善を行う必要がある。

第6章 おわりに

本稿では、初期条件が適切に入力できない場合でも、利用者が少ない労力で賃貸物件検索ができる賃貸物件検索アプリケーションの提案を行った。賃貸物件を探す利用者が全て詳細な条件を考えているわけでない。また、引っ越しを検討している利用者の中には、少なからず引っ越し先の地理情報を把握していない人がいると考えられる。そういった利用者が、少ない労力で希望の賃貸物件を探せるシステムとして Thompson Sampling を用いた Contextual Bandit を適用したアプリケーションを提案した。t 回目までのフィードバックの結果とフィードバック物件のコンテキストを用いてパラメータの事後分布を求め、事後分布からランダムにパラメータのサンプリングを行う。そのパラメータと物件のコンテキストを用いて各物件が利用者に適合とされる確率を計算し、確率の高い上位 10 件の物件を表示するシステムを実装した。比較手法として適合性フィードバックの既存手法の Rocchio の式と提案手法を比べた際に、フィードバック回数が少なくなることを確認するために 2 種類の実験を行った。評価方法は、表示された物件 10 件のうち、適合である物件が 7 件になるまでのフィードバック回数として評価を行った。

シミュレーション実験では、一人暮らしと 3 人暮らしの利用者を想定して初期検索クエリと物件選択条件を設定し、実験を行った。特に実験 1,4 の収束する回数が他の実験と比べて少なかった。実験 1 のみ合計金額に上限と下限を設定していたため、この条件をシステムで推定することが難しいと考えられる。また、実験 4 では、初期検索で物件選択条件に適合する物件が一つもなかった。利用者に適合とされる物件が一つでも無いと、適合する物件の推定ができないと分かった。

利用者実験では、架空の利用者設定を研究室のメンバーに提示し、実際の利用者が賃貸物件検索を行うことを想定して実験を行った。4 人の利用者で提案手法のフィードバック回数が少ない結果となった。しかし、物件選択条件によって比較手法の方が提案手法よりフィードバック回数が最大で 16.3 回少ない結果が出ており、提案手法の課題が見られた。

2 種類の実験で提案手法の方が比較手法よりフィードバック回数が少なくなることが確認できた。ただし、物件選択条件によって比較手法の方が提案手法よりフィードバック回数が少なくなる結果も見られた。シミュレーション実験のうち

フィードバック回数を考慮すると、家族3人暮らしより、一人暮らしの物件を探す利用者に対して少ないフィードバック回数で利用者に適合の物件の提案ができると考えられる。

また、式 (2.4.6) と (2.4.7) を比較すると、かなり大雑把な事後分布の近似をしていることが分かる。正確な事後分布を推定していないため、2種類の実験で提案手法のフィードバック回数や収束しない試行が多くなったと考えられる。これに関しては、Bianca ら [9] が事後分布を求める際に近似するのではなく、解析的に求める方法を提案している。この実装を行うことにより、提案手法のフィードバック回数や収束しない試行を少なくできる可能性があると考えられる。

本稿での結果を受けて、情報検索の分野において適合性フィードバックを対象とする場合、賃貸物件検索に限らず何かしらの情報検索において Contextual Bandit がうまく働く場面があると考えられる。ただし、利用者実験のようにあまり効果的に働かない可能性もあるため、Contextual Bandit の適用には多くの実験を行い、適切な場合を見極めた上で選択する必要がある。

謝辞

本研究を進めるにあたり、指導教官の鈴木先生には研究が行き詰まった際には、多くのアドバイスや研究を進める上でのヒントを下さり無事に卒業研究を終えることができました。特に、卒業研究を始めた際には、私はどのような研究を行うかまったく決まっていませんでした。ですが、鈴木先生から助言を頂き、自分の過去を振り返って私が興味を持った研究に取り組むことができました。また、卒業論文の執筆の際には、細かい部分まで添削をして下さったおかげでよい論文を完成させることができました。ありがとうございました。

秘書の佐野さんや現在は産休で休んでいらっしゃる井尾さんには、色々な手続きの際に丁寧に教えてくださり、滞りなく進めることができました。また、佐野さんには研究室のメンバーで昼食を食べる機会を設けていただき、普段研究室内ですることのないような会話をすることで親睦を深めることができました。

利用者実験を行うにあたっては、鈴木研究室の皆様にはそれぞれの研究があり忙しい中被験者を務めて下さり、貴重なデータを集めることができました。利用者実験の中で提案手法の実験は計算に時間がかかるため、皆さんの貴重な時間を多く私の研究に割いていただきました。4年生の皆さんには、お互いの研究について意見を言い合ったり自分が面白いと思った話などをすることで、仲を深めることができました。ご協力ありがとうございました。

アットホーム株式会社からは、貴重な物件データを提供していただきました。お礼申し上げます。

最後に、本研究ならびに学業、生活にわたるすべてのことにおいて経済的・心身的に支えて下さった家族に感謝し、お礼申し上げます。高校3年間には、朝は早く夜は遅い時間に駅に送り迎えをしてくださいました。両親のおかげで、生活に不自由なく勉強に集中することができました。大学4年間には、岐阜大学が実家から遠いこともあり、初めて一人暮らしをすることになりました。初めてということもあり、私は最初のうちはとても不安でしたが日々生活面や健康面などを気にかけてくださり、最初に感じていた不安はすぐになくなり、大学4年間で大きな病気をすることもなく勉強に集中でき、そしてアルバイトなど勉強に関係しないこれから必要となる良い経験を多くすることができました。これから、学生という立場か

ら卒業し社会人の一人となります。また一人暮らしをすることになりますが、大学4年間の経験を活かして生活をしていくことができると思います。私がこれまでに経験したことを活かし、新しいことはすぐに学び自分の中に取り入れ、早く1人前の社会人となり、親孝行ができるように精進して参ります。

参考文献

- [1] 松井治樹, 伊藤潤, 李相協, 平澤茂一. Rocchio-based フィードバック手法に基づく情報検索. FIT(情報科学技術フォーラム)2003. D-006.
- [2] 川田涼平, 藤田桂英ほか. 複数回交渉のための多腕バンディットに基づくメタ戦略. 第 81 回全国大会講演論文集, Vol. 2019, No. 1, pp. 275–276, 2019.
- [3] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, Vol. 24, pp. 2249–2257, 2011.
- [4] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.
- [5] 本多淳也, 中村篤祥. バンディット問題の理論とアルゴリズム. 講談社, 2016.
- [6] 本村駿乃介, 高木英行. 受容度を用いた賃貸物件データベース検索に関する研究. 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集第 34 回ファジィシステムシンポジウム, pp. 775–780. 日本知能情報ファジィ学会, 2018.
- [7] 中野猛, 下垣徹, 橋本拓也, 渡邊卓也ほか. Jubatus の機能を利用した二者択一型不動産賃貸物件推薦サービスの開発. *デジタルプラクティス*, Vol. 5, No. 2, pp. 130–138, 2014.
- [8] 原島純, 黒橋禎夫. テキストの表層情報と潜在情報を利用した適合性フィードバック. *自然言語処理*, Vol. 19, No. 3, pp. 121–142, 2012.
- [9] Bianca Dumitrescu, Karen Feng, and Barbara E Engelhardt. Pg-ts: improved thompson sampling for logistic contextual bandits. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4629–4638, 2018.

発表リスト

- [1] 安江優人, 鈴木優『直感的な適合性フィードバックによる物件検索システムの構築』東海関西データベースワークショップ, 2021.
- [2] 安江優人, 鈴木優『Contextual Bandit を用いた賃貸物件検索システム』第 14 回データ工学と情報マネジメントに関するフォーラム, 2022.