DEIM Forum 2022 F41-3

Contextual Bandit を用いた賃貸物件検索システム

安江　優人† 　鈴木　優†

† 岐阜大学工学部電気電子・情報工学科　〒 501–1193 岐阜県岐阜市柳戸 1–1
E-mail: †{x3033160@edu.gifu-u.ac.jp, ysuzuki@gifu-u.ac.jp}

あらまし　賃貸物件検索サイトでは一般に，初期検索時に適切な条件を入力しなかったときユーザにとって不要な物件が表示されてしまうため，必要な物件を見つけることが困難である．これを解決するために，適合性フィードバックによる賃貸物件検索を考えた．既存の適合性フィードバックにおいては，検索結果選択に Rocchio の式が用いられるものが多い．しかし，検索クエリが表示物件と類似したものであるとき，複数回フィードバックを行ったとしても検索クエリがあまり変化せず，類似した物件しか表示されない問題がある．これを解決するために，バンディットアルゴリズムの一つである Contextual Bandit を用いた検索システムを構築した．探索フェーズでは，ユーザが適合とした物件とは類似していない物件を検索結果として表示するため，ユーザの物件選択の可能性を広げる．活用フェーズでは，探索フェーズで得られた結果を用いてユーザに適合する物件を推定し表示する．活用フェーズと探索フェーズを繰り返し，類似した物件以外の物件を表示することにより，検索精度が向上すると考えた．評価実験により， Rocchio の式との比較を行い，最大 30 回のフィードバックで 7 件の適合物件を検索できるかどうかを確かめるため，フィードバック回数を比較した．その結果，Rocchio の式では 7 件の物件を提示できなかったが，提案手法では最大で 9.7 回のフィードバック数で 7 件の物件を提示することができた．

キーワード　ユーザ支援，バンディットアルゴリズム，Contextual Bandit，ベイズ推定

1　はじめに

近年，アットホーム [1] や SUUMO[2] など多数の企業が賃貸物件検索サイトを運営し，多くのユーザに活用されている．現在の物件検索システムは，探す地域や最寄駅などを選択し，家賃や間取り，駅までの徒歩の時間などの条件を入力し，該当物件を絞り込むことが主流である．しかし，初期条件で適切な条件が入力できない場合，初期条件の適合物件数が非常に多く，ユーザに不必要な物件も比例して多く表示される．そのため，ユーザ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考えられる．この状況は，特に引っ越し先の地理情報を把握していないユーザに多いと考えられる．本研究は，適切な条件を入力できない場合でも，ユーザ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．本研究では，多くの研究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で，ユーザとの対話を通して希望の物件条件を推定できないかと考えた．適合性フィードバックの主な手法として Rochhio 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，設定条件に適合する物件を絞ることができなかった．

そこで，ユーザの適合物件を探す際にユーザによる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．当たる確率が不明のスロットマシンが複数あり，1 試行で 1 台しかプレイできないとする．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の最大化の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．

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，報酬をユーザが行う適合・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を探す．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定する．そのため，探索を多く行う必要があり，ユーザに適合する物件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，各ユーザの年齢や性別，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，通常のバンディットアルゴリズムより効率的に

1：https://www.athome.co.jp/
2：https://suumo.jp/

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，UCB や Thompson Sampling を用いたアルゴリズム [3] が適用されている．本研究では，精度が一番良い [3] とされる 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ユーザ設定を提示し，状況に応じた物件検索を行うユーザ実験を行った．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，Rocchio の式を適用したシステムと比較し，提案手法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提案手法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 7 件の適合物件を提示できず，提案手法でフィードバック数 9.7 回で 7 件の適合物件を提示した実験だった．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，Rocchio の式より少ないフィードバック数で物件の提示ができることを示せた．しかし，比較手法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該当物件数が多いと考えられる場合に，提案手法を適用したシステムのフィードバック回数の方が比較手法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ユーザ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，本研究だけではなく，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．

2 関 連 研 究

本村ら [4] 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．家賃の上限と下限を 3 万円 ~5 万円と設定したとする．ユーザが 3 万円 ~5 万円の間の家賃で，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 のようにユーザ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，その中でユーザ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，各項目の受容度を受容度関数としてユーザに入力させる．受容度関数は，単に一次関数のような単調増加や単調減少しないため，曲線で入力を行う仕様になっている．また，各項目の受容度曲線を乗算または加算することにより，どの項目を重視するか分かり，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ユーザに物件の提案を行う．

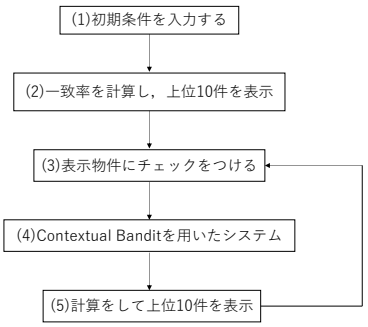

図 1　アプリケーションの流れ

中野ら [5] は，オンライン機械学習を用いた賃貸物件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，ユーザの推薦に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ユーザに提示する．ユーザ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，希望する物件条件を推定し適合物件数を絞り込み，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，ユーザ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，ユーザ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．よって，最寄り駅をどこにするか決定しているユーザが使用するシステムだと考えられる．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，ユーザにとって意外な物件を提示できる．しかし，確率的に表示物件を決めるため，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ユーザ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，3.2 節や 4.4 節，4.5 節で述べる．

3 提 案 手 法

3.1 節では実装したアプリケーションの流れについて述べる．その後，3.2 節では，Thompson Sampling を用いた Contex-

tual Bandit について詳細に述べる．3.3 節では，システムのアルゴリズムについて述べる．

3.1 検索 Web アプリケーション

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．アプリケーションの表示画面は初期検索のクエリを入力する画面と物件を表示し，表示物件に適合・不適合のフィードバックをしてもらう画面の 2 種類である．

アプリケーションの流れは図 1 になる．

（1） ユーザが初期検索のクエリを入力し，送信する．

（2） ユーザ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，上位 10 件の物件を表示する．

（3） 表示物件がユーザ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） それまでの表示物件のフィードバックの有無と住所や家賃，築年などの物件情報を元に Contextual Bandit を適用したシステムで計算を行う．

（5） 計算の結果，次にユーザに適合と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，ユーザの適合物件を絞る．

以降は，主に (4) の Contextual bandit について述べる．

3.2 Contextual Bandit

本研究で適用する報酬が 2 値である Thompson 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 $a_1, a_2, a_3, ..., a_n$，i を $1, 2, 3, ..., n$ とする．t 回目に i 番目のアーム a_i を提示したとすると，ユーザやアーム 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が各賃貸物件，コンテキストが各賃貸物件の物件情報とモデル化できる．

（1） ユーザにアーム a_i を提示する．

（2） ユーザの特徴や (1) のアーム a_i の特徴を表すコンテキスト x_{t,a_i} とユーザの適合・不適合，t 回目までの試行で受け取っている結果を用いて，ユーザ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） 事後分布からニュートン法により MAP 推定量 $\hat{\theta_t}^{MAP}$ を求める．

（4） 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） (4) の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 [6] を参考に詳しく述べる．

3.2.1 アームの提示

t 回目にユーザにアーム a_i を提示し，適合・不適合のフィードバックを行う．本研究ではアームが物件であるため，ユーザは提示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．アプリケーションでは，各物件ごとのチェックボックスに適合であればチェックを入れ，不適合であればチェックを入れないフィードバックをユーザは行う．

3.2.2 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，各アームの報酬を r_{t,a_i} とすると，次にユーザに適合・不適合とフィードバックされる各物件の報酬の確率はそれぞれ以下の式になる．報酬とは，ユーザから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．

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

ユーザに適合であるアームを求めたいので，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．

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

よって，パラメータ θ_t を求めることでユーザ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 と事後分布を説明する．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より導出される確率分布である．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，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分布の事後確率最大化推定 (MAP 推定) を行う．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．

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

上記の式を θ_t の関数として最大化し推定を行う．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．

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

3.2.3 MAP 推定量 $\hat{\theta_t}^{MAP}$ を求める

(2) 式より，θ_t についての 1 次微分，2 次微分は以下になる．ただし，I_d は単位行列とする．

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

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

1 次微分を求めたので，$G(\theta_t) = 0$ を満たす MAP 推定量

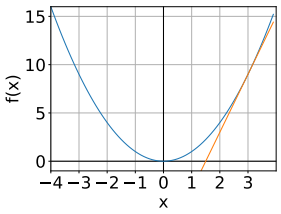

図 2　$f(x)$ と接線 1

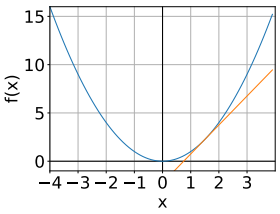

図 3　$f(x)$ と接線 2

$\hat{\theta_t}^{MAP}$ 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．

ここで，ニュートン法について述べる．図 2,3 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 をニュートン法により求める．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 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．$f'(x)$ は接線の傾きであり，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，以下の式が成り立つ．

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

図 3 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 (3) と同様に計算をすると以下になる．

$$f'(1.5) = \frac{f(1.5) - 0}{1.5 - 0.75} \Leftrightarrow 0.75 = 1.5 - \frac{f(1.5)}{f'(1.5)} \tag{4}$$

式 (4) より次は $x = 0.75$ で接線を引いて同様に計算を行う．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．式 (3),(4) を一般化すると式 (5) になる．

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{5}$$

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 (5) を当てはめると以下になる．

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

3.2.4　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 (2) 式の 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．

$$-\log p(\theta_t|\{r_{s,a_i}\}_{s=1}^t) \approx \frac{1}{2}(\theta_t - \hat{\theta_t}^{MAP})^T H(\hat{\theta_t}^{MAP})(\theta_t - \hat{\theta_t}^{MAP}) + \frac{(\hat{\theta_t}^{MAP})^T \hat{\theta_t}^{MAP}}{2\sigma^2} + \sum_{s=1}^t \log(1 + e^{(\hat{\theta_t}^{MAP})^T x_{s,a_i}}) - \sum_{s:r_{s,a_i}=1} (\hat{\theta_t}^{MAP})^T x_{s,a_i} + const \tag{6}$$

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 (7) になる．

$$-\log \frac{1}{(\sqrt{2\pi})^n \sqrt{|\Sigma|}} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) = \frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu) + const \tag{7}$$

Algorithm 1 Contextual bandit with logistic

Input: σ

for $t = 1, 2, 3...N$ **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

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

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

end for

(6) 式の一項目と (7) 式を比べるとほぼ同じ形をしている．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 $MultiNorm(\hat{\theta_t}^{MAP}, (H(\hat{\theta_t}^{MAP}))^{-1})$ とすることで近似でき，事後分布を求めることができる．

3.2.5　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，(1) 式より最適なアームを求めることができる．そして，アームをユーザに提示する (1) に戻る．(1)〜(5) までの流れを繰り返すことによりユーザに適合とされるアームを推定する．

初めにバンディットアルゴリズムでは探索フェーズと活用フェーズを用いていると述べたが，ここまでの Contextual Bandit の流れでは具体的にどちらを利用するかは述べていない．Thompsom Sampling を用いる場合，求める事後確率によって探索フェーズか活用フェーズかが決まるが，明示的に 2 つの切り替えは行われない．イメージとして，確率分布の分散が小さければ活用フェーズを表し，分散が大きければ探索フェーズを表すと考えられる．

3.3　アプリケーションへの適用

Contextual Bandit をシステムに適用するために Contextual Bandit の問題設定を本研究に当てはめると，各アームが各賃貸物件，コンテキストが賃貸物件の住所や家賃など物件に関する情報となる．これまで述べた流れを用いて，提案手法のアルゴリズムは [3] [6] を参考に，Algorithm 1 の通りとした．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．今回行った実験では，$\sigma = 0.01$，$i = 20$ とした．

表 1 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．この場合のコンテキストは，表 1 の都道府県より右側のカラムの値となる．

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 に不適合，物件 5 に適合のフィードバックを行ったとする．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 が適合であるので，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．

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利用しているが，住所や間取りなどの文字列と家賃や築年などの数値が混合している．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った．

文字列は Bag of Words によってベクトル化を行った．Bag of Words とは，自然言語処理において自然言語で書かれた文をベクトルに変換する方法である．例として，文字列のみの物件データをそれぞれ以下とする．

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

文を形態素解析によって単語に分け，単語ごとに数値を割り振り辞書を作成する．物件データは単語ごとに分かれているため，形態素解析をする必要がない．物件データを元に単語辞書 d を作成すると以下になる．

$$d = \{ \text{"}q_1^{(1)}\text{"} : 1, \text{"}q_2^{(1)}\text{"} : 2, ..., \text{"}q_n^{(1)}\text{"} : n,$$
$$\text{"}q_1^{(2)}\text{"} : n+1, \text{"}q_2^{(2)}\text{"} : n+2, ..., \text{"}q_n^{(2)}\text{"} : 2n\}$$

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，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，単語ごとの one hot vector は以下になる．

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

表 1　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

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．よってデータごとの Bag of Words は以下になる．

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

数値に関しては，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．一つのカラムの数値データを $x = [x_1, x_2, ...x_n]$ とすると，正規化は以下の式により行った．ただし，$x_k^{'}$ は x_k を正規化した値である．

$$x_k^{'} = \frac{x_k - \min(x)}{\max(x) - \min(x)} \tag{8}$$

正規化を行った数値ベクトルと Bag of Words によりベクトル化を行った文字列のベクトルを結合して一つの物件データのコンテキストのベクトルとし，アルゴリズムの計算に用いた．

4　実　　験

今回の実験は，比較手法として 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かめるために行った．フィードバック回数が少ないということは，ユーザ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ユーザ実験の 2 種類行った．シミュレーション実験は，一人暮らしと 3 人暮らしのユーザを想定して初期検索クエリと物件選択条件を設定した．ユーザの状況を複数準備し，提案手法の方がフィードバック回数が少なくなる状況を確認した．ユーザ実験は，ユーザの状況を提示し，実際のユーザの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ない．しかし，実際の物件検索では必ずしも条件に 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，ユーザの途中の条件変更を可能とし，ユーザが条件変更後に，提案手法が変更後の条件に適合する物件を提示できるか確認した．

4.1 節では実験で使用したデータについて，4.2 節では比較手法について述べる．4.3 節では実験の評価方法について述べる．4.4 節では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，4.5 節では研究室のメンバーに実装したアプリケーションで物件検索を行ってもらった実験について述べる．

4.1　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 [3]を使用した．実験データには，2019 年 1 月 1 日から 2019 年 12 月 31 日の 1 年間に登録された賃貸物件データを利用した．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．

3：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/

実際には物件番号が重複するデータや，ユーザ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．

4.2 比較手法

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 (9) になる [1]．ただし，Q は検索クエリベクトル，$x^+(x^-)$ は適合（不適合）物件データベクトル，$N^+(N^-)$ は適合（不適合）物件データ数，$X^+(X^-)$ は適合（不適合）物件データの集合，α，β，γ はハイパーパラメータとする．

$$Q_{i+1} = \alpha Q_i + \beta \frac{1}{N^+} \Sigma_{x^+ \in X^+} x^+ - \gamma \frac{1}{N^-} \Sigma_{x^- \in X^-} x^- \quad (9)$$

式 (9) では，検索クエリと物件データをベクトル化して計算を行う．ベクトル化については，3.3 節で述べたようにベクトル化を行った．本稿では，Q_0 をユーザが初期検索時に入力した初期検索クエリとする．初期検索クエリもベクトル化を行う必要がある．文字列のベクトル化は，3.3 節で作成した単語辞書 d を用いて Bag of Words を行い，ベクトル化を行った．数値の正規化は，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・最小値を用いて式 (8) により正規化を行った．正規化を行った数値ベクトルと Bag of Words によりベクトル化を行った文字列のベクトルを結合して初期検索クエリのベクトルとした．

表示物件は，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．

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

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り，検索クエリ，適合物件，不適合物件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物件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．今回の実験では，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．

4.3 評価方法

原島ら [7] 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，ユーザ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 を求め，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 2 の MAP は，$(1 + \frac{2}{4}) \div 2 = 0.75$ となる．

本研究は，ユーザの物件を探す負担を減らすことを目的としている．P@K は主に一度のフィードバックでの精度を評価し，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，評価方法はユーザ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ユーザが物件を探す負担が少ないと評価できる．

具体的な評価方法は，ユーザが最初に適合物件をフィードバックした試行からフィードバック回数を計測した．フィードバックされた物件 10 件のうち，適合物件が 7 件以上を規定物件数として実験終了とし，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 回で実験終了とし，収束しないと定義し評価とした．

表 2　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	d	不適合	0

4.4 シミュレーション実験

4.4.1 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．実験 1~3 は一人暮らし、実験 4~6 は 3 人暮らしの利用者を想定している．それぞれの初期検索クエリは表 4 の通りである．

ここで初期検索クエリの決定方法を述べる．本研究は初期検索で適切な条件が入力できない場合を対象とする．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事前実験を行った．その結果，駅名を入力すると該当物件が絞られると分かった．同じ市や区の物件でも駅名により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も入力しないとした．

一人暮らしの未入力の数値データは，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．

それぞれの実験の物件選択条件は表 3 になる．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 の通りである．

表 3　物件選択条件

実験	駅までの徒歩	駅までのバス	支払う合計金額	築年	部屋の広さ	駐車場有無
1	15 分以内	—	50000~55000 円	1981 年以降	20 平米以上	—
2,3	15 分以内	—	55000 円以内	1981 年以降	20 平米以上	—
4	20 分以内	10 分以内	90000 円以内	1981 年以降	56 平米以上	有 or 無
5	20 分以内	—	69000 円以内	1981 年以降	56 平米以上	有 or 無
6	20 分以内	20 分以内	71000 円以内	1981 年以降	56 平米以上	有 or 無

表 4　各実験の初期検索クエリ

	都道府県	所在地名 1	家賃	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
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
7	岐阜県	岐阜市	—	2	2000	—	—
8	愛知県	—	—	2	2000	—	—

4.4.2 結　　果

比較手法は表示物件が変化しないと検索クエリベクトルも変化しないため，実験は一回行った．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するため，表示物件が同じでもパラメータ θ_t が異なるため，次の表示物件も異なる．そこで，実験は 50 回行い，収束した実験のフィードバック (FB) 回数の平均で評価を行った．各手法の結果は表 5 である．

表 5 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．

比較手法では，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，各物件データベクトルとコサイン類似度を求めた際に，表示物件が変化しなかった．既に検索クエリベクトルが理想のベクトルに近づき，更新を行っても更新前と比べあまり検索クエリベクトルが変化しないからだと考えられる．

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た．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と適合物件の推定ができないと考えられる．

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該当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た．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，収束回数に差が出たと考えられる．また，実験 4 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，すべて不適合だった．Contextual Bandit は適合・不適合の結果両方を得ることで選択肢を絞るため，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たため実験 3 と比べ，該当物件数が多いが，収束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べると，3 人暮らしより一人暮らしのユーザに対して，提案手法がフィードバック回数 15 回以内にユーザに適合物件を提示できると考えられる．

表 5　該当物件数と各実験結果

		比較手法	提案手法	
実験	物件数	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

4.5 ユーザ実験

4.5.1 実 験 設 定

以下の実験設定と図 4 の実験 7,8 のように検索クエリの一部を指定し，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ユーザに自由に決めてもらい実験を行ってもらった．指定した実験設定は「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「就職して、名古屋駅付近で仕事をする」の 2 種類である．それぞれの指定した検索クエリは表 4 の実験 7,8 である．

4.5.2 実 験 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 6 である．

どちらの実験も，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ユーザがいる結果となった．このユーザは，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設定していることが多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かったと考えられる．提案手法と比較手法のフィードバック回数を比べると，最大 16.3 回差があった．

一方，提案手法の方がフィードバック回数が少なかったユーザは，物件選択条件が複雑だった．例えば，所在階が 1or2 階だったり，駐車場付き物件が良いなどの物件選択条件にしたユーザが見られた．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 回となった実験だった．

また，実験 7,8 を比べると，ユーザの収束しなかった試行の総数は同数だが，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，愛知県の物件から希望物件を検索している．岐阜県の全物件数と比べると約 18 倍愛知県の方が多い．

表 6　実験 7,8 の各手法の結果

		比較手法	提案手法	
実験	ユーザ	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

よって，提案手法はより多くの物件から適合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，フィードバックをしてから次の物件表示に 30 秒 ~1 分かかった．これはシミュレーション実験でも見られた．愛知県の物件データは 42,533 個あり，ベクトル化した際に一つのベクトルの次元数が 1,081 次元となった．そのため，3.3 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 1 分ほどの時間がかかった．これは検索システムとしては十分な速度であるとはいえず，改善を行う必要がある．

5 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，ユーザが少ない労力で賃貸物件検索ができる物件検索アプリケーションの提案を行った．賃貸物件を探すユーザが全て詳細な条件を考えているわけでない．また，引っ越しを検討するユーザの中には，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．そのユーザが少ない労力で賃貸物件を探せるシステムとして，Thompson Sampling を用いた Contextual Bandit を適用したシステムを提案した．比較手法として 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，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できないと考えられる．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．ユーザの適合物件が一つでも無いと，物件の推定ができないと分かった．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ユーザに対して 15 回以内のフィードバックでユーザに適合する物件の提案ができると考えられる．

今後の研究課題として，a) 該当物件数によるフィードバック回数の差，b) 別の評価方法の検討，及び c) 事後分布の近似方法について解決しなければならないと考える．

課題 a) では，該当物件数に関係なく，比較手法より少ないフィードバック回数での適合物件の提示を行いたい．ユーザ実験では，該当物件数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 16.3 回少ない結果となった．該当物件数が多いと考えられる場合には，比較手法の方がフィードバック回数が少なくなった．反対に該当物件数が少ないと考えられる場合には，提案手法の方がフィードバック回数が少なくなった．前者の場合でも提案手法のフィードバック回数を少なくする方法を考えなければならない．

課題 b) では，別の評価方法での実験を考えている．今回は，規定個数の適合物件が提示されるまでのフィードバック回数を評価方法とした．しかし，実際の物件検索ではユーザが納得する物件が提示されているかやアプリケーションの操作性などユーザの満足度も重要となる．そのため，システムの考察や改善を行うために別の評価方法として、利用したユーザに対して満足度調査を行うことが考えられる．

課題 c) では，事後分布の近似方法の改善を行いたい．式 (6) と (7) を比較すると，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，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 [8] が事後分布を近似して求めるのではなく，解析的に求める手法を提案している．この実装を行い，提案手法のフィードバック回数や収束しない試行を現在の結果より少なくできると考えられる．

謝辞　本研究の一部は JSPS 科研費 18H03342，19H04221，19H04218 の助成を受けたものです．

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] 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.

[5] 中野猛, 下垣徹, 橋本拓也, 渡邉卓也ほか. 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.

[6] 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.

[7] 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.

[8] Bianca Dumitrascu, 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.