

条件検索におけるコンバージョンと離脱の双方を考慮した条件推薦

長島 弘昂[†] 田島 敬史^{††} 清田 陽司^{†††}

[†] 京都大学大学院情報学研究科 〒606-8501 京都府京都市左京区吉田本町

^{††} 京都大学大学院情報学研究科 〒606-8501 京都府京都市左京区吉田本町

^{†††} 株式会社 LIFULL 〒102-0083 東京都千代田区麹町 1-4-4

E-mail: [†]nagashima@dl.soc.i.kyoto-u.ac.jp, ^{††}tajima@i.kyoto-u.ac.jp, ^{†††}kiyotayoji@lifull.com

あらまし キーワード検索に対する検索キーワード推薦と同様に、物件検索サービス等の条件検索については、検索条件の推薦手法が研究されている。条件推薦においては、ユーザの検索の手間を軽減しつつ、検索したいアイテムへとユーザを到達させる必要がある。一方で、検索したいアイテムに到達させられる検索条件を推薦したところで、実際にその推薦条件が使われなければ、ユーザが検索したいアイテムに到達することは困難になる。理想的には、ユーザが使いたいと思え、かつ、検索したいアイテムに到達できる条件を推薦したいのであるが、このような検索条件は、必ずしも存在するとは限らない。そこで、本研究では、ユーザが使いたいと思える条件の推薦から、検索したいアイテムへとユーザを到達させられる条件の推薦へと段階的に移行させる手法を提案する。具体的には、ログ中の離脱に繋がらない条件の推薦から、ログ中のコンバージョンに繋がる条件の推薦へと、条件推薦手法を移行させる。

キーワード 情報検索, 情報推薦, 条件検索, クエリ推薦, 商品検索, 不動産検索

1 はじめに

近年の情報のデジタル化に伴い、社会に分散していた多種多様な情報は、従来の紙媒体から、電子ファイルやウェブページへとその媒体を変容させつつある。時と場所に制約されずとも、膨大なデータにアクセスができるようになった昨今には、同時に、情報検索サービスも必要不可欠なシステムとなっている。

情報検索の種類としては、主に、検索したいアイテムを表すキーワードを検索窓等に入力して行う検索手法（キーワード検索）と、検索したいアイテムに合致する様々な条件を指定して行う検索手法（条件検索）が存在する。広く知られている情報検索サービスとして、例えば、Google はキーワード検索に基づくサービスであり、LIFULL HOME'S は条件検索に基づくサービスであるといえる。そのほかにも、条件検索のサービスとしては、中古車検索サイトやホテル検索サイト、一部のファッション通販サイト等が該当するといえる。

このような情報検索に対しては、ユーザの検索の手間の軽減を目的とした、検索キーワードや検索条件の推薦手法が研究されている。推薦の目的である、ユーザの検索の手間の軽減は、具体的に、検索キーワードや検索条件を入力する手間の軽減と、検索キーワードや検索条件を考える手間の軽減とに大別される。つまり、ユーザの想定している検索キーワードや検索条件を推薦する行為は、検索キーワードや検索条件を入力するユーザの手間を軽減する行為であり、一方、ユーザの想定していなかった絞り込みに効果的な検索キーワードや検索条件を推薦する行為は、検索キーワードや検索条件を考えるユーザの手間を軽減する行為であるといえる。

条件推薦においては、このようなユーザの検索の手間を軽減しつつ、検索したいアイテムへとユーザを到達させる必要があ



図1 ユーザが使いたいと思えるような条件と検索が成功するような条件の比較

る。しかしながら、ユーザを検索したいアイテムに到達させられるような条件を推薦したところで、実際に、その条件が使われなければ、ユーザが検索したいアイテムに到達することは困難になる。つまり、ユーザが使いたいと思えるような条件と、ユーザを検索したいアイテムに到達させる為の条件（検索が成功するような条件）は、分けて考えなければならない。図1の例では、ユーザの希望に基づけば、ユーザが使いたいと思えるような条件は「家賃 ≤ 50,000」であるが、家賃が条件から僅かに外れるものの総合的に良い物件が他に存在する為、検索が成功するような条件は「家賃 ≤ 50,000」ではない可能性が高い。理想的には、ユーザが使いたいと思え、なおかつ、検索が成功するような条件を推薦したいのであるが、このような検索条件は、必ずしも存在するとは限らない。

従って、検索が成功するまでに必要なユーザの検索の手間を軽減する為には、ユーザが使いたいと思えるような条件と、検索が成功するような条件を、いかにして推薦し分けるかが課題

となる。単純に、ユーザが使いたいと思えるような条件を推薦し続けても、ユーザの検索したいアイテムに到達できるとは限らず、また、検索が成功するような条件を推薦し続けても、その条件がユーザに受け入れられなければ、ユーザが離脱することになり、結果として、検索は成功しない。

よって、提案手法では、検索条件の遷移に基づき、ユーザが使いたいと思えるような条件の推薦から、検索が成功するような条件の推薦へと段階的に移行させることで、ユーザの検索の手間を軽減しつつ検索を成功させる条件推薦を行う。図1の例であれば、まず、「家賃 \leq 50,000」という条件を推薦し、ユーザに結果を納得してもらった上で、より緩い検索条件を推薦する。なお、ユーザが使いたいと思えるような条件の推薦においては、過去の検索ログ中のユーザの離脱行動を活用し、検索が成功するような条件の推薦においては、コンバージョン行動を活用する。換言すれば、離脱に繋がらない検索条件こそ、ユーザが使いたいと思えるような検索条件であり、また、コンバージョンに繋がる検索条件こそ、検索が成功するような検索条件であると考えられる。

本論文の以降の構成としては、まず、2節において、関連研究について述べる。3節で提案手法の詳細を述べたうえで、続く4節では、データセット及び評価方法等に関する詳細を含む実験について論じ、5節で本研究の総括を行う。

2 関連研究

本節においては、2.1節で、条件検索を対象とした関連研究、2.2節で、キーワード検索を対象とした関連研究を紹介し、最後に、2.3節で、クリック行動等のユーザの様々な行動に基づく推薦等に関する研究を紹介する。

2.1 条件検索におけるユーザの手間の軽減に関する研究

条件検索におけるユーザの検索の手間の軽減に関する研究としては、検索したいアイテムへと早く絞り込む為の条件推薦[1]や、検索条件に該当するアイテムが存在しない場合の条件推薦[2]の他、佐々岡ら[3]による、ランキング学習とそれに基づく条件推薦が存在する。

佐々岡らは、まず、ランキング学習において、過去の検索ログ中の、アイテムのクリック行動に基づき、主要な各検索条件に対して、アイテムのランキング学習を行う。そのうえで、それらの結果を、検索条件間の類似度を用い統合することによって、任意の検索条件に対して、アイテムのランキングを生成する。その後、生成したランキングにおいて上位に現れるアイテムが解に含まれるような検索条件を推薦する。

佐々岡らは、条件推薦に過去のアイテムのクリック行動のみを活用しており、ユーザのその他の行動は活用していない。それに対し、ユーザの離脱行動とコンバージョン行動を活用している我々の手法は、クリック行動よりも前に起こる非離脱行動と、クリック行動よりも後に起こるコンバージョン行動の両面から、ユーザの検索を支援することができるといえる。ユーザの様々な行動に基づく推薦等については、近年、益々研究が行

われており、詳しくは2.3節で論じる。

また、佐々岡らは、再検索回数も考慮していない為、所与の検索条件が同一であれば、再検索回数によらず、同一の検索条件が推薦される。一方、我々は再検索回数を考慮している為、ユーザの検索がどの程度まで進んだかによって、推薦条件を変えることができる。

2.2 キーワード検索におけるユーザの手間の軽減に関する研究

キーワード検索におけるユーザの検索の手間の軽減に関する研究としては、キーワード（クエリ）推薦に関する研究が広く知られている。

Meiら[4]は、複数の検索意図を有する検索クエリが与えられた場合における、クエリ推薦手法を提案している。Meiらのクエリ推薦は、ある二つのクエリから同一のURLが頻繁に選択されていた場合、これらのクエリは互いに推薦クエリとして適切であるという考えに基づいており、クエリ推薦に当たっては、クエリを表すノード群とURLを表すノード群から成る二部グラフを構築する。あるクエリを表すノードとあるURLを表すノード間のエッジには、そのクエリの検索結果からそのURLが選択された回数が重みとして与えられる。よって、構築した二部グラフにおいて、あるクエリノードから重みの大きいエッジをたどり、そのURLノードから重みの大きい異なるエッジをたどった先にあるクエリノードが、推薦すべきクエリとなる。

しかしながら、Meiらのクエリ推薦手法においては、類似したクエリが複数推薦される可能性がある。この場合、複数の検索意図を有するクエリに対して、単一の検索意図に沿ったクエリのみが推薦されることとなり、そのほかの検索意図を有するユーザについては、有効なクエリ推薦がなされない。

この問題に対して、今井ら[5]は、構築したクエリノード群とURLノード群から成る二部グラフをクラスタリングすることによって、特定の検索意図に偏らないクエリ推薦を行う手法を提案している。

しかし、これらのクエリ推薦手法は、条件検索には適さないと考えられる。特に、数値項目を含むような条件検索においては、あるアイテム選択時の検索条件が多様になることが想定される為である。この場合、あるアイテムに関して、特に重みの大きいエッジ、つまり、特に頻出する検索条件が存在せず、適切な検索条件が推薦されない可能性がある。換言すれば、条件検索においては、過去のアイテムのクリック行動を活用する代わりに、より多様性を抑え得る情報、例えば、過去のユーザの検索行動を活用する手法が有効である可能性がある。なお、我々の手法は、後者の検索行動を活用している為、検索結果を受けて検索条件を最低限緩めるユーザの行動等は比較的捉えやすいといえる。

このような、過去のユーザの検索行動を活用したクエリ推薦手法としては、Caoら[6]の手法や、RNNを用いたSordoniら[7]の手法、ランキングとのマルチタスク学習を採用したAhmadら[8]の手法が存在する。一方で、これら手法については、ユーザの想定している次の検索条件を当てることに焦点が

当てられており、クエリ推薦に際し、コンバージョンまでもが同時に考慮されているとはいえない。ユーザの想定している次の検索条件、すなわち、ユーザが使いたいと思えるような条件と、検索が成功するような条件を、分けて考えなければならぬことについては、1節で論じた通りである。

そのほか、キーワード検索におけるユーザの手間の軽減に関する研究として、クエリを推薦するのではなく、検索結果のリランキング (re-ranking) によって、検索したいアイテムに到達するまでに必要なユーザの手間を軽減する研究も、Azar ら [9] によってなされているが、同様に、条件検索には適さないと考えられる。多くの条件検索に基づくサービスでは、検索結果中のアイテムを、アイテムのある属性の値に従って並び替えることができる。このようなランキングでは、検索結果中のアイテムの順位によって、アイテムの特定の属性の値を大まかに把握することができる。これに対して、Azar らによる提案手法のランキングでは、検索結果中の順位によって、アイテムの特定の属性の値を把握することが困難となり、検索結果として得られるアイテムの確認に手間がかかることが想定される。

2.3 ユーザの様々な行動に基づく推薦等に関する研究

2.1 節では、佐々岡らが、過去のアイテムのクリック行動に基づき、条件推薦を行うことを述べた。しかしながら、ユーザの行動はクリック行動のみではなく、離脱行動、コンバージョン行動等と、多様である。EC サイトであれば、加えて、アイテムのカートイン行動、更には、お気に入り登録行動や評価行動等も、ユーザの行動であるといえる。

コンバージョン行動、あるいは、評価行動といった、ユーザの種類の行動に基づき、ユーザの好みを推定する協調フィルタリング [10], [11] をはじめとした推薦 (single-behavior recommendation) 等とは対照的に、ユーザの様々な行動に基づく推薦 (multi-behavior recommendation) 等については、近年、益々研究が行われている。その初期の研究としては、主に、行列分解 (matrix factorization) を拡張した CMF (Collective Matrix Factorization) [12] が挙げられる。CMF においては、アイテムに対するユーザの様々な行動に対応する複数の行列を、同時に因数分解する。また、多目的最適化手法 [13] や、他にも、GCN (Graph Convolutional Network) ベースの例であれば、MBGCN (Multi-Behavior Graph Convolutional Network) [14] が存在し、MBGCN においては、アイテム間の類似度を求める為、ユーザの様々な行動の重要度を自動的に学習するような、ユーザとアイテムの伝播方法が設計されている。

一方で、これらのモデルは、ユーザとアイテム間の関係に主眼が置かれている為に、ユーザに対して直接アイテムを推薦する応用等は考えられるが、そのアイテムに到達できるような検索キーワードや検索条件を推薦する応用は容易ではない。アイテム推薦と、クエリ推薦及び条件推薦は、検索したいアイテムに到達するまでに必要なユーザの検索の手間を軽減する、異なるアプローチであるから、クエリ推薦及び条件推薦における、ユーザの様々な行動の考慮も、重要な研究課題であるといえる。

ユーザの様々な行動に基づく推薦の他、ユーザの様々な行動に

基づくランキング学習に関する研究も行われている。Hu ら [15] は、EC サイトを対象に、SSMDP (Search Session Markov Decision Process) に基づき、強化学習によってユーザの一連の検索における期待累積報酬を最大化するようなランキングポリシーを学習する手法を提案している。SSMDP の定義においては、ユーザのコンバージョン行動、離脱行動、そして、再検索行動が組み込まれている。

しかしながら、ユーザの様々な行動に基づく推薦と同様、このモデルは、検索キーワードや検索条件を推薦する為のものではない。2.2 節でも論じた通り、ランキング学習と、クエリ推薦及び条件推薦は、検索したいアイテムに到達するまでに必要なユーザの検索の手間を軽減する、異なるアプローチであるから、クエリ推薦及び条件推薦における、ユーザの様々な行動の考慮も、依然、重要な研究課題であるといえる。

特に、ユーザの検索を支援する方法として、アイテム推薦と条件推薦間、そして、ランキング学習と条件推薦間では、網羅性における差異が大きいといえる。例えば、アイテム推薦によって到達したアイテムが、ユーザにとって好ましいものであった場合、ユーザは、その他にも同様に好ましいアイテムが存在する可能性を考えるであろう。しかしながら、アイテム推薦では、たとえ、その他に好ましいアイテムが存在しなかったとしても、それを示すことが困難となる。一方、ユーザが、条件推薦によって好ましいアイテムに到達した場合、その他にも同様に好ましいアイテムが存在するかは、推薦された検索条件を介し確認することができる。ランキング学習と条件推薦についても同様である。そして、この網羅性は特に、住まい探しや仕事探しといった、重要なライフイベントや意思決定において、重視される可能性がある。

3 提案手法

本研究では、条件検索において、ユーザの検索の手間を軽減すると同時に、ユーザを検索したいアイテムに到達させる為、ユーザが使いたいと思えるような条件の推薦から、検索が成功するような条件の推薦へと段階的に移行させる条件推薦手法を提案する。

提案手法は、ユーザが使いたいと思えるような条件の推薦、検索が成功するような条件の推薦、二種類の条件推薦の段階的な移行に分けられ、それぞれについて、3.1 節、3.2 節、3.3 節で論じる。

3.1 ユーザが使いたいと思えるような条件の推薦

本節では、検索条件の遷移に基づいた、ユーザが使いたいと思えるような条件の推薦手法について述べる。条件推薦においては、過去の検索ログ中のユーザの離脱行動を活用し、最もサービスを離脱しにくいと考えられる検索条件の推薦を行う。換言すれば、離脱に繋がらない検索条件こそ、ユーザが使いたいと思えるような検索条件であると考えられる。

従って、まずはユーザの離脱行動を定義する必要がある。本研究ではその定義に検索行動を用いる。今、あるユーザ u が、

検索条件 q_0, q_1, \dots, q_n で、 $n+1$ 回の一連の検索を行い、その後検索を行わなかった場合、離脱行動 e_i は $i \in \{0, \dots, n\}$ で次のように定義される。

$$e_i = \begin{cases} 1 - c_i & (i = n) \\ 0 & (i \in \{0, \dots, n-1\}) \end{cases} \quad (1)$$

ここで、 c_i はユーザ u のコンバージョン行動を表しており、式 1 の範囲では、 q_n の後に u がコンバージョンに至れば 1、至らなければ 0 の値をとる。コンバージョン行動 c_i の詳細については、3.2 節の式 6 で述べる。

最もサービスを離脱しにくいと考えられる検索条件の推薦においては、条件推薦時のユーザの検索条件を用いる。すなわち、今、あるユーザが、検索条件 q_0, q_1, \dots, q_i で、 $i+1$ 回の一連の検索を行ったとき、その直後に推薦すべき、最もサービスを離脱しにくいと考えられる検索条件 $q_{noexit_{i+1}}$ は、条件推薦手法 $f_{noexit}(q)$ によって、 $f_{noexit}(q_i)$ として得られる。

$$q_{noexit_{i+1}} = f_{noexit}(q_i) \quad (2)$$

ここで、式 1 に立ち返れば、最もサービスを離脱しにくいと考えられる検索条件における「離脱」とは、「検索条件を推薦された後、推薦条件によって再検索することなく、サービスを離脱してしまう」ことを指しているのであり、「推薦条件によって再検索した後、検索結果を見て、サービスを離脱してしまう」ことを指している訳ではないことに注意が必要である。

以上を踏まえ、最もサービスを離脱しにくいと考えられる検索条件の推薦の為の学習では、離脱行動がなかった時点の前後の検索条件を用いる。すなわち、今、あるユーザについて $e_i = 0$ であれば、 q_i に対して q_{i+1} がサービスを離脱させない上で有効な検索条件であったとみなす。このような非離脱前後の検索条件対 $q_u(j) \rightarrow q_u(j+1)$ のデータを全ユーザについて蓄積した上で、 q_i の直後に最も多く現れた検索条件を、 q_i に対する $q_{noexit_{i+1}}$ として推薦する。ここで、 $q_u(j)$ は、ログ中のユーザ u の、 $0 \leq j$ 番目の検索条件を表す。以降、 $e_u(j)$, $c_u(j)$, n_u についても同様にこの表記を用いる。また、式 3 を含め、本研究の条件推薦では、最良の検索条件一つのみを推薦しているが、複数推薦するように拡張することも容易である。

$$f_{noexit}(q_i) = \arg \max_q |\{q_u(j) \rightarrow q_u(j+1) | \exists u, j. (e_u(j) = 0 \wedge q_u(j) = q_i \wedge q_u(j+1) = q)\}| \quad (3)$$

従って、本条件推薦手法にハイパーパラメータは存在せず、パラメータチューニングも不要である。

一方で、式 3 の条件推薦手法では、ある検索直後の非離脱のみを考慮しており、それよりも後の推薦条件を選択した後の非離脱については考慮がなされていない。実際には、ユーザの一連の検索に対して離脱を防ぐことが、ユーザとサービス双方にとって重要である。そこで、本節では、ユーザが使いたいと思えるような条件の推薦の範囲から若干外れるものの、式 3 の条件推薦手法に改良を加えた条件推薦手法を提案する。

ある検索直後の非離脱のみならず、それよりも後の非離脱に

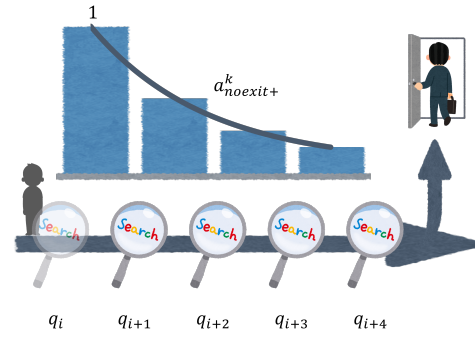


図 2 q_i から $k \in \{1, \dots, m-1\}$ 回検索した後の非離脱に、減衰させた重みを与えた場合のイメージ

についても考慮する、ということは、間接的にコンバージョンを考慮することになり得る。コンバージョンに至らなかったユーザの平均検索回数に比べ、コンバージョンに至ったユーザの平均検索回数は多くなることが想定される為である。この場合、より長く非離脱となるような検索条件とは、よりコンバージョンに至るような検索条件であり、式 3 による推薦条件と比して、ユーザが使いたいと思えるような条件でなくなる可能性が考えられる。一方で、依然、ユーザが使いたいと思えるような条件ではあり、なおかつ、検索が成功するような条件でもある、という点で、二種類の条件推薦の段階的な移行によるコンバージョンと離脱の双方の考慮とはまた異なった、コンバージョンと離脱の双方の考慮のアプローチであるという見方ができる。

ある検索以降の非離脱を考慮する方法としては、例えば、学習時、その後の検索回数による加重を行う方法が考えられる。今、あるユーザについて $e_i = 0, e_{i+m} = 1$ であれば、 q_i に対して q_{i+1} が、今後 m 回の検索にわたってサービスを離脱させない上で有効な検索条件であったとみなし、 m のスコアを与える、という方法である。

しかしながら、これは、本来学習したかった $q_i \rightarrow q_{i+1}$ における非離脱と、 q_i から $k \in \{1, \dots, m-1\}$ 回検索した後の非離脱に、等しい重みを与えている状況として解釈できる。今は、 q_i について、非離脱に繋がるような検索条件を学習したいのであるから、まずは、 q_i の直後に離脱されないことを重点的に考慮すべきである。

そこで、一連の検索に対する非離脱の考慮においては、本来学習したかった $q_i \rightarrow q_{i+1}$ における非離脱と、 q_i から $k \in \{1, \dots, m-1\}$ 回検索した後の非離脱に、それぞれ、1, $a_{noexit+}^k$ ($a_{noexit+} \in [0, 1]$) のスコアを与え、それらを加算したスコアをそのユーザの $q_i \rightarrow q_{i+1}$ のスコアとすることで、 q_i の直後の非離脱を重視しつつ、それよりも後の非離脱についても考慮を行う (図 2)。このような非離脱前後の検索条件対のデータを全ユーザについて蓄積した上で、 q_i の直後に現れた最もスコア $s_{noexit+}(q_i, q)$ の高い検索条件 $q = q_{noexit_{i+1}}$ を、 q_i に対して推薦する。

$$f_{noexit+}(q_i) = \arg \max_q s_{noexit+}(q_i, q) \quad (4)$$

$$s_{noexit+}(q_i, q) = \sum_u \sum_{j \in \{j | e_u(j)=0 \wedge q_u(j)=q_i \wedge q_u(j+1)=q\}} \sum_{k=0}^{n_u-j-1} a_{noexit+}^k \quad (5)$$

ここで、 $a_{noexit+} \in [0, 1]$ はスコアの減衰率を表すハイパーパラメータであり、 $a_{noexit+} = 0$ のときに最も減衰し $q_{noexit+i+1} = q_{noexit+i+1}$ となる。従って、この $a_{noexit+}$ を 0 から 1 に調節することで、今最もサービスを離脱しにくいと考えられる検索条件の推薦から、以降最も長くサービスを離脱しにくいと考えられる検索条件の推薦までが、可能となる。

3.2 検索が成功するような条件の推薦

本節では、検索条件の遷移に基づいた、検索が成功するような条件の推薦手法について述べる。条件推薦においては、過去の検索ログ中のユーザのコンバージョン行動を活用し、最もコンバージョンに繋がると考えられる検索条件の推薦を行う。換言すれば、コンバージョンに繋がる検索条件こそ、検索が成功するような検索条件であると考えられる。

従って、まずはユーザのコンバージョン行動を定義する必要がある。本研究ではその定義に検索行動を用いる。今、あるユーザが、検索条件 q_0, q_1, \dots, q_n で、 $n+1$ 回の一連の検索を行い、その後検索を行わず、なおかつ、任意の $k \in \{1, \dots, m\}$ (m はコンバージョンの回数) について、検索条件 q_{j_k} による検索と検索条件 q_{j_k+1} による検索との間 ($j_k = n$ ならば、検索条件 q_{j_k} による検索の後) に、コンバージョンに至った場合、コンバージョン行動 c_i は $i \in \{0, \dots, n\}$ で次のように定義される。

$$c_i = \begin{cases} 1 & (i \in \{j_1, \dots, j_m\}) \\ 0 & (i \in \{0, \dots, n\} - \{j_1, \dots, j_m\}) \end{cases} \quad (6)$$

最もコンバージョンに繋がると考えられる検索条件の推薦においては、条件推薦時のユーザの検索条件を用いる。すなわち、今、あるユーザが、検索条件 q_0, q_1, \dots, q_i で、 $i+1$ 回の一連の検索を行ったとき、その直後に推薦すべき、最もコンバージョンに繋がると考えられる検索条件 $q_{cv_{i+1}}$ は、条件推薦手法 $f_{cv}(q)$ によって、 $f_{cv}(q_i)$ として得られる。

$$q_{cv_{i+1}} = f_{cv}(q_i) \quad (7)$$

基本的な方針としては、学習において、コンバージョン行動があった時点の直前の検索条件を用いる。すなわち、今、あるユーザについて $c_{i+1} = 1$ であれば、 q_i に対して q_{i+1} がコンバージョンの上で有効な検索条件であったとみなす。このようなコンバージョン直前の検索条件対 $q_u(j) \rightarrow q_u(j+1)$ ($q_u(j+1)$ の直後にコンバージョン) のデータを全ユーザについて蓄積した上で、 q_i の直後に最も多く現れた検索条件を、 q_i に対する $q_{cv_{i+1}}$ として推薦する。これは、非離脱前後の検索条件対を学習に用いていた、式 3 の学習方法と同様である。

しかしながら、一般的に、コンバージョン行動は非離脱行動と比べて遥かに頻度が低い。コンバージョン直前の検索条件対のみを学習に用いる場合、十分な検索条件に対して学習がな

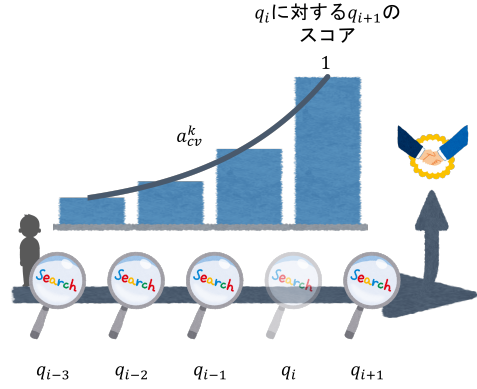


図3 $c_{i+1} = 1$ のとき、 q_i から $k \in \{1, \dots, i\}$ 回前の検索条件対に、減衰させた重みを与えた場合のイメージ

されないことが容易に推測される。よって、その他にも、コンバージョンに繋がると考えられる検索条件を学習させる必要があるといえる。

そこで、本条件推薦手法においては、コンバージョンに至るより前の検索条件対についても、コンバージョンに繋がると考え学習を行う。つまり、あるユーザについて $c_{i+1} = 1$ であれば、 $q_i \rightarrow q_{i+1}$ の他、 $q_{i-1} \rightarrow q_i, \dots, q_0 \rightarrow q_1$ も、コンバージョンの上で有効な検索条件対であったとみなす。

一方で、より少ない再検索回数でコンバージョンに至る検索条件が優先的に推薦されるよう、それら検索条件対に対しては、コンバージョンに至るまでの再検索回数に応じて重みづけを行う。先の例であれば、 $q_i \rightarrow q_{i+1}$ に 1 の重みを付与するのに対して、 $q_{i-1} \rightarrow q_i, \dots, q_0 \rightarrow q_1$ には、それぞれ、 a_{cv}, \dots, a_{cv}^i の重みを付与する (図3)。ここで、 $a_{cv} \in [0, 1]$ は重みの減衰率を表すハイパーパラメータである。

よって、最もコンバージョンに繋がると考えられる検索条件の推薦においては、このようなコンバージョンに至るより前の検索条件対のデータを全ユーザについて蓄積した上で、 q_i の直後に現れた最もスコア $s_{cv}(q_i, q)$ の高い検索条件 $q = q_{cv_{i+1}}$ を、 q_i に対して推薦することになる。

$$f_{cv}(q_i) = \arg \max_q s_{cv}(q_i, q) \quad (8)$$

$$s_{cv}(q_i, q) = \sum_u \sum_{j \in \{j | c_u(j+1)=1\}} \sum_{k \in \{k | q_u(j-k)=q_i \wedge q_u(j-k+1)=q\}} a_{cv}^k \quad (9)$$

3.3 二種類の条件推薦の段階的な移行

提案手法では、3.1 節で定義した、ユーザが使いたいと思えるような条件の推薦から、3.2 節で定義した、検索が成功するような条件の推薦へと段階的に移行させることで、ユーザの検索の手間を軽減しつつ検索を成功させるような条件推薦を行う。なお、ユーザが使いたいと思えるような条件の推薦手法としては、3.1 節の二手法 $f_{noexit}(q_i)$, $f_{noexit+}(q_i)$ の内、後者の、一連の検索に対する非離脱の考慮を行った手法を採用する。

具体的に本手法では、検索が成功するような条件の推薦の重み $w_{cv} \in [0, 1]$ を、検索が成功するような条件の推薦に際し計

算した各検索条件のスコア（式 9）を正規化したものに一律に乘じるとともに、重み $1 - w_{cv}$ を、ユーザが使いたいと思えるような条件の推薦に際し計算した各検索条件のスコア（式 5）を正規化したものに一律に乘じ、その後、各検索条件に対して両者を加算することで、最終的な各検索条件 q のスコア $s_{hybrid}(q_i, q, i + 1)$ を算出する。あとは、そのスコアが最も高かった検索条件 q を、推薦条件としてユーザに推薦する。

$$f_{hybrid}(q_i, i + 1) = \arg \max_q s_{hybrid}(q_i, q, i + 1) \quad (10)$$

$$s_{hybrid}(q_i, q, i + 1) = w_{cv} * \frac{s_{cv}(q_i, q)}{\sum_q s_{cv}(q_i, q)} + (1 - w_{cv}) * \frac{s_{noexit+}(q_i, q)}{\sum_q s_{noexit+}(q_i, q)} \quad (11)$$

ここで、条件推薦それぞれの重みを決定している w_{cv} は、ユーザの検索がコンバージョンに対してどの程度まで進んだかを表す $phase \in [0, 1]$ に基づき算出する。一方、その為には、ユーザがあとどの程度検索すればコンバージョンに至ることが期待されるのか推定しなければならないが、提案手法では、これを、そのユーザの直前の検索条件に基づき行う。例えば、検索条件 q_0, q_1, \dots, q_i で、 $i + 1$ 回の一連の検索を行ったユーザ u に対しては、 $n_{from}(q_i)$ として、 u のコンバージョンまでの残り再検索回数を推定する。ここで、 $n_{from}(q_i)$ は、過去に q_i を経由してコンバージョンに至った全ユーザ中の、全 q_i から全コンバージョンまでの平均再検索回数を返す関数である。以上を踏まえ、 $phase$ は、式 12 の通り求められる。なお、式では、ユーザがこれまでにどの程度検索してきたかが $i + 1$ として得られることを用いている。

$$phase = \frac{i + 1}{i + 1 + n_{from}(q_i)} \quad (12)$$

さて、 $phase$ に基づく w_{cv} の算出方法としては、最も単純には、 $phase$ をそのまま用いる方法が考えられる。つまり、今、ユーザの検索がコンバージョンに対して半分程度進んだのであれば、ユーザが使いたいと思えるような条件の推薦も、検索が成功するような条件の推薦も、同程度重視するのである。この算出方法は簡潔で分かりやすい一方で、 w_{cv} が $phase$ のみによって定まる為、例えば、コンバージョンの直前までは離脱しないことをより重視する、という方針は十分に表現できない。

そこで、提案手法では、 w_{cv} を設定する為の追加のパラメータ $b_{cv} \in (0, 1)$ を導入した上で、このパラメータを組み込んだ、 $(0, 0)$, $(1 - b_{cv}, b_{cv})$, $(1, 1)$ の三点を線形補間した関数によって、 w_{cv} を求める（図 4）。

以上の定義により、 w_{cv} は、推薦対象となっているユーザのこれまでの検索回数 $i + 1$ と直前の検索条件 q_i に依存している他、 b_{cv} にも依存している為、式 10 に対応する提案手法のハイパーパラメータは、 $b_{cv} \in (0, 1)$ 、そして、 $a_{cv} \in [0, 1]$ （式 9）、 $a_{noexit+} \in [0, 1]$ （式 5）である。

これまででは、ユーザの検索がコンバージョンに対してどの程度まで進んだかによって、二種類の条件推薦の重みを決定していた。その一方で、過去、最終的にコンバージョンに至ったこ

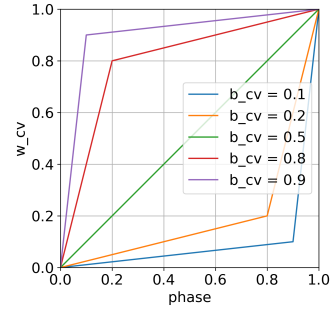


図 4 w_{cv} を $(0, 0)$, $(1 - b_{cv}, b_{cv})$, $(1, 1)$ の三点を線形補間した関数によって求めた場合の、 $phase$ に対する w_{cv} の変化

とのある検索条件と、そうでない検索条件とは、一律に扱っていた。従って、例えば、過去、最終的にコンバージョンに至ったことのある検索条件の中から、最もサービスを離脱しにくいと考えられる検索条件を推薦すること等は、現状、不可能である。

そこで、式 10 の提案手法に対して、コンバージョンに至ったことのある検索条件の重みを設定する為のパラメータ $b_{incv} \in [0.5, 1)$ を導入し、式 13 によって、 $f_{hybrid+}(q_i, i + 1)$ を定義する。 $b_{incv} = 0.5$ のときは、 $f_{hybrid+}(q_i, i + 1) = f_{hybrid}(q_i, i + 1)$ となり、 $b_{incv} > 0.5$ のときに、コンバージョンに至ったことのある検索条件が重視される。

$$f_{hybrid+}(q_i, i + 1) = \arg \max_q \begin{cases} b_{incv} * s_{hybrid}(q_i, q, i + 1) \\ (\exists u, j, k. (q_u(j) = q_i \wedge c_u(j + k) = 1)) \\ (1 - b_{incv}) * s_{hybrid}(q_i, q, i + 1) \\ (otherwise) \end{cases} \quad (13)$$

よって、式 13 に対応する提案手法のハイパーパラメータは、 $b_{cv} \in (0, 1)$ 、 $b_{incv} \in [0.5, 1)$ 、そして、 $a_{cv} \in [0, 1]$ 、 $a_{noexit+} \in [0, 1]$ である。

4 実 験

本節では、まず、4.1 節にて、実験で使用するデータセットについて論じ、続く 4.2 節で、評価指標及びベースラインを含む、評価方法について、4.3 節で、実験手順について論じる。最後に、4.4 節にて、実験結果として各手法の性能とハイパーパラメータ等を掲載した上で、それに対する考察を行う。

4.1 データセット

条件検索に基づくサービスのログとして、本研究では、株式会社 LIFULL から提供を受けた、LIFULL HOME 'S のアクセスログを利用する。当該アクセスログには、2020 年 10 月 1 日から 2020 年 10 月 15 日までの期間に、東京都内の物件検索を一度でも行ったことのあるユーザに関する、期間内の全てのアクセスログが記録されている。

アクセスログの内、本研究では、特に、検索条件を含む条件検索のログ、及び、どの物件について問合せたかのログを用い、

実験を行う。検索条件を含む条件検索のログは、各ユーザの一連の検索 $\{q_0, q_1, \dots, q_n\}$ に関する情報を抽出し、対応する離脱 $\{e_0, e_1, \dots, e_n\}$ を特定する為に用い、また、どの物件について問合せたかのログは、各ユーザの一連の検索に対応するコンバージョン $\{c_0, c_1, \dots, c_n\}$ を特定する為に用いる。なお、検索条件の項目としては、駅徒歩分、賃料、築年数、専有面積を用いる。

4.2 評価方法

3.3 節で論じた、二種類の条件推薦の段階的な移行による条件推薦手法は、ユーザの検索回数に応じて推薦の目的が変化する。ユーザの検索回数がまだ少ない内は、サービスを離脱させないことが条件推薦の主な目的になるのに対して、ユーザの検索回数が多くなってきたら、コンバージョンに至らせることが条件推薦の主な目的となる。

一方で、ユーザの検索回数に応じて推薦の目的が変化するのは、最終的には、ユーザの検索の成功こそが、提案手法の目的となる。従って、ユーザのコンバージョンに着目し、評価を行うことが、適当であると考えられる。

なお、オフライン評価では、無論、過去ログ中の一ユーザ u の一検索 q_i に対して、提案手法が $q_{hybrid+i+1}$ を推薦したとして、その推薦条件が、 u をコンバージョンに至らせるかどうかは不明である。そこで、本実験では、推薦条件が平均的にどの程度ユーザをコンバージョンに至らせ得るかを評価する。すなわち、今、過去ログ中の一ユーザ u の一検索 q_i に対して、提案手法が $q_{hybrid+i+1}$ を推薦した場合、その推薦条件が、直前に q_i を入力していた全ユーザの内、どの程度を最終的にコンバージョンに至らせたかを評価する。以降、この割合をコンバージョン率と呼び、 $r(q_i \rightarrow q_{hybrid+i+1})$ として表すことにする。従って、以上の評価により、過去ログ中の全ユーザによる全検索に対して、それぞれ、コンバージョン率が得られる。

本研究では、得られた各検索に対応するコンバージョン率を、全検索について平均した値、平均コンバージョン率を評価指標として算出する。平均コンバージョン率は、一検索条件ではなく一検索を単位としている為、より検索頻度の高い検索条件を用いた検索ほど、重視していることになる。

以上の評価方法を踏まえ、まずベースラインとして思いつく条件推薦手法が、コンバージョン率を最大化するような条件推薦手法 $f_{cvr}(q_i)$ である。つまり、 q_i に対しては、 $r(q_i \rightarrow q)$ を最大化する q を推薦する手法である。

$$f_{cvr}(q_i) = \arg \max_q r(q_i \rightarrow q) \quad (14)$$

他にも、高いコンバージョン率を得る可能性のある条件推薦手法としては、コンバージョンを考慮している、検索が成功するような条件の推薦手法 $f_{cv}(q_i)$ (3.2 節) が考えられる。従って、 $f_{cv}(q_i)$ もベースラインに加え、更には、提案手法を構成する、ユーザが使いたいと思えるような条件の推薦手法 $f_{noexit}(q_i)$ (3.1 節) もベースラインとする。両者ともに、コンバージョンか離脱か的一方のみを考慮しており、コンバージョンと離脱の双方を考慮してはいない。以降、ベースラインの三手法につい

表 1 条件推薦手法の性能比較

手法	平均コンバージョン率 [%]
cvr	1.76
cv	1.91
noexit	1.94
noexit+	2.04
hybrid+	2.09

表 2 提案手法 hybrid+ のハイパーパラメータ

名前	値
a_{cv}^*	0.70
$a_{noexit+}^*$	0.97
b_{cv}^*	0.40
b_{incv}^*	0.97

て、順に、cvr, cv, noexit の略称を用いる。

それに対し、コンバージョンと離脱の双方を考慮している提案手法として、一連の検索に対する非離脱の考慮を行った条件推薦手法 $f_{noexit+}(q_i)$ (3.1 節)、そして、二種類の条件推薦の段階的な移行による条件推薦手法 $f_{hybrid+}(q_i, i+1)$ (3.3 節) を評価する。以降、各手法について、順に、noexit+, hybrid+ の略称を用いる。

4.3 実験手順

4.1 節、4.2 節を踏まえ、実験は下記手順に従って行う。

(1) アクセスログから、各ユーザの一連の検索と、それに対応する離脱及びコンバージョンに関する情報を取得する。

(2) 全ユーザを 8:2 に分割し、ユーザ単位で、三種類の情報を訓練データとテストデータに分割する。

(3) 訓練データを用いて、5 分割交差検証により、cv, noexit+, hybrid+ のパラメータチューニングを行う。

(4) 獲得したパラメータを固定した上で、訓練データを用いて全手法の学習を行い、テストデータを用いて全手法の評価指標を算出する。

4.4 実験結果

各手法の性能は、表 1 の通りである。提案手法 noexit+ と hybrid+ はともに、ベースライン noexit の性能を上回り、有意水準 5% で有意差ありという結果になった。これは、コンバージョンと離脱の双方を考慮した為であると推測され、特に、noexit+ と noexit の平均コンバージョン率の差は分かりやすい。両手法、ユーザの離脱行動 e_i のみを活用しているのにも関わらず、 e_i に基づき間接的にコンバージョンを考慮した noexit+ の方が、値が高く出ている。また、その noexit+ よりも更に hybrid+ の平均コンバージョン率は高く (有意水準 5% で有意差あり)、コンバージョン行動を併せて活用することで、平均コンバージョン率を更に上げられる可能性を示唆している。

そして、表 1 の性能に対応する、提案手法 hybrid+ のハイパーパラメータの値は、表 2 の通りである。ハイパーパラメータは、Optuna を使い、TPE [16] によって最適化を行った。 $a_{cv}^* \neq 1$ からは、ユーザのコンバージョン行動を起点として、そこからより過去の検索になるほど、重要度が低くなる可能性が読み取

表3 提案手法 hybrid+ のアプレーション研究

除外	平均コンバージョン率 [%]
{}	2.09
{ a_{cv} }	2.08
{ $a_{noexit+}$ }	2.03
{ b_{cv} }	2.07
{ b_{incv} }	2.01

れる。一方、 $a_{noexit+}^* \approx 1$ となっており、推薦した後に最も長くサービスを離脱しにくいと考えられる検索条件の推薦が有効であると分かる。また、 $b_{cv}^* < 0.5$ から、非離脱を重視することが効果的であると考えられる。ここで、 $b_{incv}^* \approx 1$ である点から、コンバージョンに至ったことのある検索条件を抽出した上で、非離脱を重視した条件推薦を行うことが、性能向上に有効であると推測される。

また、hybrid+ の各ハイパーパラメータを初期値に固定して性能を評価することで、各ハイパーパラメータの効果を測定した(表3)。表3の中でも、 $a_{noexit+}$ 、 b_{incv} を除外した場合の性能低下が目立つ点から、ある検索直後の非離脱のみならずそれよりも後の非離脱についても考慮すること、また、最終的にコンバージョンに至ったことのある検索条件を抽出することが、特に効果的であるといえる。

5 結 論

条件推薦では、検索したいアイテムへとユーザを到達させられる条件を推薦したところで、実際に、その条件が使われなければ、ユーザが検索したいアイテムに到達することは困難になる。この課題に対し、本論では、検索条件の遷移に基づき、ユーザが使いたいと思えるような条件の推薦から、検索が成功するような条件の推薦へと段階的に移行させる条件推薦手法を提案した。なお、ユーザが使いたいと思えるような条件の推薦においては、過去の検索ログ中のユーザの離脱行動を活用し、検索が成功するような条件の推薦においては、コンバージョン行動を活用した。この提案手法に対し、LIFULL HOME'S のアクセスログを用い性能を測定したところ、コンバージョンと離脱の双方を考慮してはいない手法と比して高い性能が得られた。この結果は、コンバージョンと離脱の双方を考慮することの有効性を示しており、最終的にはユーザの検索の成功こそが条件推薦の目的であると捉えれば、コンバージョンのみならず、離脱をも考慮した本論の提案手法によって、より多くのユーザの検索の成功が期待されるといえる。

謝 辞

本研究では、株式会社 LIFULL から提供を受けた、LIFULL HOME'S のアクセスログのデータセットを利用した。また、本研究は、JST CREST (JPMJCR16E3)、JSPS 科研費 21H03446 の支援を受けたものである。

文 献

- [1] Senjuti Basu Roy, Haidong Wang, Gautam Das, Ullas Nambiar, and Mukesh Mohania. Minimum-effort driven dynamic faceted search in structured databases. In *Proceedings of the 17th ACM CIKM*, CIKM '08, pp. 13–22, 2008.
- [2] Davide Mottin, Alice Marascu, Senjuti Basu Roy, Gautam Das, Themis Palpanas, and Yannis Velegrakis. Iqr: An interactive query relaxation system for the empty-answer problem. In *Proceedings of the 2014 ACM SIGMOD*, SIGMOD '14, pp. 1095–1098, 2014.
- [3] 佐々岡哲哉, 田島敬史, 清田陽司. 条件検索におけるランキング学習とそれに基づく条件推薦. DEIM Forum 2021, 2021.
- [4] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM CIKM*, CIKM '08, pp. 469–478, 2008.
- [5] 今井良太, 戸田浩之, 関口裕一郎, 望月崇由, 鈴木智也, 今井桂子. Web 検索サービスにおける多義的なクエリ推薦手法. *DBSJ Journal*, Vol. 9, No. 1, pp. 1–6, 2010.
- [6] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD*, KDD '08, pp. 875–883, 2008.
- [7] Alessandro Sordani, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM CIKM*, CIKM '15, pp. 553–562, 2015.
- [8] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *ICLR*, 2018.
- [9] Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. Multiple intents re-ranking. In *Proceedings of the 41st ACM STOC*, STOC '09, pp. 669–678, 2009.
- [10] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th WWW*, WWW '01, pp. 285–295, 2001.
- [11] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 8th IEEE ICDM*, pp. 263–272, 2008.
- [12] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD*, KDD '08, pp. 650–658, 2008.
- [13] Liang Tang, Bo Long, Bee-Chung Chen, and Deepak Agarwal. An empirical study on recommendation with multiple types of feedback. In *Proceedings of the 22nd ACM SIGKDD*, KDD '16, pp. 283–292, 2016.
- [14] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. *Multi-Behavior Recommendation with Graph Convolutional Networks*, pp. 659–668. 2020.
- [15] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD*, KDD '18, pp. 368–377, 2018.
- [16] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th NIPS*, NIPS'11, pp. 2546–2554, 2011.