

# X-Transformerを用いた 店舗のクラスタリングに基づく訪問予測

野村ひかる<sup>1</sup> 米川慧<sup>2</sup> 李智<sup>1</sup> 原隆浩<sup>1</sup> 天方大地<sup>1</sup> 前川卓也<sup>1</sup> 黒川茂莉<sup>2</sup>

**概要:**近年、企業の効率的な集客施策のため、ユーザの訪問履歴などの行動履歴からユーザの行動パターンや嗜好を分析し、ユーザの興味がある場所 (Point-Of-Interest, POI) を予測することでユーザの将来の訪問予測を行う POI 推薦の研究が盛んに行われている。一般的に、店舗に対する訪問予測の精度を上げるためには、各店舗に対する多くのユーザの訪問情報が大量に必要となる。しかし、店舗によって訪問者数は様々であり、訪問者数が少ない店舗が非常に多い。また、訪問予測の対象となる店舗は大量に存在しているため、計算コストが非常に大きい。そのため、訪問予測を単純に行うことは困難である。

そこで本研究では、ユーザの訪問履歴をテキスト、店舗をラベルとみなし、テキスト分類における計算コストの問題に対応した X-Transformer のラベルクラスタリングの手法を用いて、店舗のクラスタリングに基づく訪問予測を行う手法を提案した。提案手法では、同じ店舗に訪れる異なるユーザは類似した店舗に興味を持つと仮定し、ユーザの分散表現をユーザの実店舗への訪問履歴から作成する。さらに、店舗に訪れたユーザの分散表現の平均を店舗の分散表現とし、それを用いて店舗のクラスタリングを行った。クラスタリングを行う際は、各クラスタに含まれる店舗の数が均等になるよう行うことで、クラスタ内の店舗数の偏りを防いだ。計算コストの問題に対応するため、ユーザに対してクラスタ予測を行い、ユーザとの類似度が上位のクラスタに含まれる店舗のみを訪問予測の対象として、訪問予測を行った。

提案手法の性能を評価するために、実世界上で Wi-Fi アクセスポイントで収集された実データを用いて実験を行った。店舗のクラスタリングを用いて予測対象を削減して行う提案手法は、単純な手法より高い予測精度を達成したことから、提案手法の有効性を確かめた。

## 1. 序論

近年、企業の集客施策として、ユーザに対する Web 上での広告表示やダイレクトメールの送信などが行われている。これらの施策はそれまでその企業を利用したことがないユーザに企業のことを知ってもらうきっかけとなる。しかし、集客施策を無作為に行った場合、企業や商品に興味のないユーザにも広告が表示されてしまい、邪魔であるといった悪印象を抱かれてしまう。そのため、効果的に集客率を向上させることは簡単でない。このような背景から、ユーザの行動履歴に基づいて企業の内容や商品に興味を持つと思われる潜在顧客を適切に抽出し、潜在顧客のみをターゲットとした集客施策に注目が集まっている。

潜在顧客の抽出のためのアプローチとして、ユーザが興味のある場所 (Point-Of-Interest, POI) を予測し、将来の訪問予測を行う研究が盛んに行われている。訪問予測の結果を利用して、ユーザの興味のある企業や商品を予測で

きる。

一般に訪問予測を行うには、店舗に対してユーザの訪問履歴が大量に必要となる。しかし、訪問者数の少ない小規模の店舗では、訪問予測に十分な量のユーザの訪問履歴を得られない場合が多い。さらに、予測対象となる店舗は大量に存在するため、すべての店舗に対して予測を行うことは計算コストが非常に大きい。そのため、訪問予測を単純に行うことは困難である。このような情報の量による問題に対して、テキストにラベルを割り当てるテキスト分類の分野では、ラベルのクラスタリングによって予測対象を絞り計算コストを抑え、さらに、情報が不足しているラベルに対する問題を緩和しているものがある。この手法について、第2章で詳しく述べる。

本研究では、店舗のクラスタリングを行い、予測対象となる店舗数を絞ることで、計算コストを削減し、また、小規模の店舗の情報が補われるという仮説の下、店舗のクラスタリングに基づいて訪問予測を行う手法を提案する。提案手法では、訪問予測の問題をテキスト分類の問題と対応させるため、ユーザの訪問履歴をテキスト、店舗をラベル

<sup>1</sup> 大阪大学

<sup>2</sup> KDDI 総合研究所

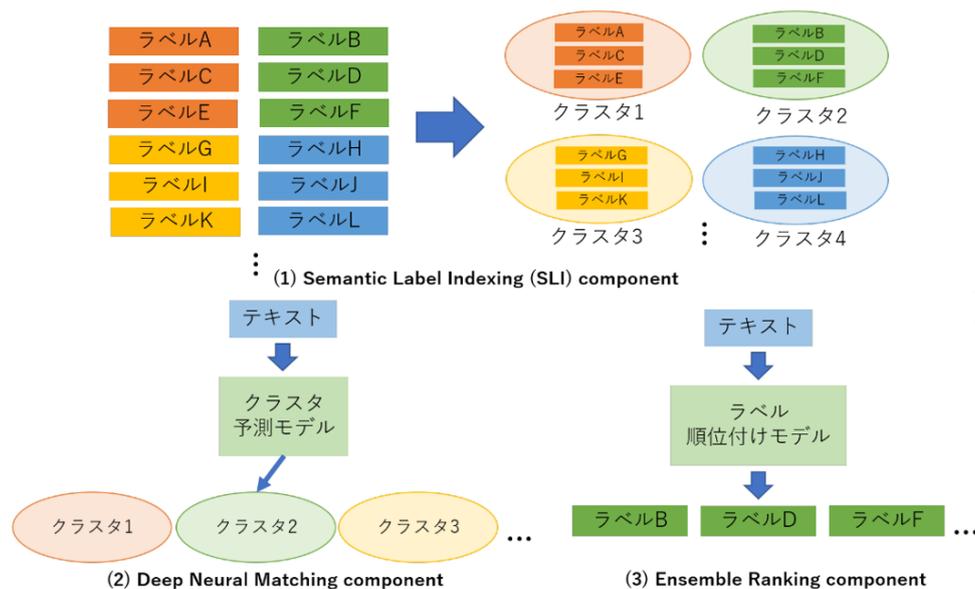


図 1 X-Transformer のフレームワーク

とみなし、ユーザに店舗を割り当てることで訪問予測を行う。ユーザの嗜好や行動の傾向などの特徴を適切に表現するため、駅などの店舗以外の場所も含めたユーザの訪問履歴を用いて分散表現を獲得し、店舗の分散表現をその店舗を訪問したユーザの分散表現の平均とする。各店舗の情報にユーザの訪問履歴を用いることで、訪問履歴が少ない小規模の店舗に対しても訪問予測を行える。また、ユーザの訪問予測は、クラスタに対する順位付けにおける上位のクラスタに含まれる店舗のみを対象として行い、訪問予測の対象となる店舗数を絞ることで、訪問予測にかかる計算コストの課題を解決した。

提案手法の性能を評価するために、実世界上で Wi-Fi を用いて収集された実データを用いて実験を行った。提案手法は単純な手法より高い訪問予測精度を達成し、店舗のクラスタリングを用いた訪問予測の有効性を確認した。以下、第2章では関連研究を紹介し、本研究との関連を述べる。第3章では本研究で提案する訪問予測手法について説明する。第4章では提案モデルの評価及びその結果の考察を行う。最後に第5章で本研究のまとめと今後の課題について述べる。

## 2. X-Transformer

本章では、提案手法のアイデアの元となる X-Transformer[1] について紹介する。

入力テキストが与えられたときに、大量のラベル候補の中から入力テキストに適する複数のラベルを割り当てるテキスト分類問題を考える。このようなテキスト分類問題は Extreme multi-label text classification (XMC) 問題と呼ばれ、XMC 問題は機械学習や自然言語処理 (NLP) の分野に

おいて最も重要かつ基本的なトピックの1つとなっている。最近では、BERT[2] や、その後続モデルである XLNet[3]、および RoBERTa[4] のような事前学習型の Transformer モデルが質問応答などの様々な NLP タスクで最先端の性能を発揮している。しかし、XMC 問題に対しては、出力空間が非常に大きくラベルの情報が少ないという問題から、Transformer モデルを上手く適用することは難解な課題であった。例えばデータが大量にあるため、メモリの使用量が莫大になることでモデルの容量が制限されることや、Wiki-500K データセット\*1における 100 個以上の学習用テキストがあるラベルは全ラベル数の約 2% しかないなど、学習用の情報が少ないラベルが過半数以上を占めているという問題がある。

このような問題点を解決し、XMC 問題に対して Deep Transformer モデルの fine-tuning を成功させたものが、X-Transformer である。X-Transformer は、競合手法である AttentionXML[5] の実験で用いられた 4 つの XMC ベンチマークデータセットにおいて、最高の性能を発揮した。本研究では、大量のラベルに対するテキスト量の分布の問題と大量の店舗に対する訪問者数の分布が似ていると考え、訪問予測手法に X-Transformer におけるラベルクラスタリングを用いたテキスト分類手法の考えを適用した。以下では、X-Transformer の構造について説明する。

X-Transformer は、Semantic Label Indexing (SLI) component, Deep Neural Matching component, および Ensemble Ranking component の 3 つの部分から構成される。X-Transformer のフレームワークを図 1 に示す。まず SLI で、ラベルのクラスタリングを実行する。最初にラベルの

\*1 <http://manikvarma.org/downloads/XC/XMLRepository.html>

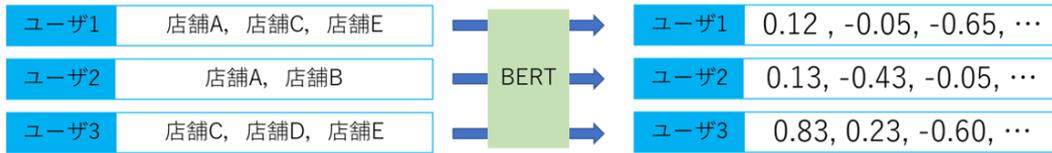


図 2 ユーザの分散表現の獲得

埋め込みとして、ラベルに関する紹介文のような短いテキストの埋め込みを与える。ラベルによっては十分なテキスト情報がないものもあるが、その場合はそのラベルが与えられているテキストの埋め込みを与える。このようなラベルの埋め込みは Positive Instance Feature Aggregation (PIFA) と呼ばれ、AttentionXML や、SLICE[6], Parabel[7], および Bonsai[8] のような最先端の XMC 手法で用いられている。ラベルの埋め込みを作成した後、他の最先端の XMC 手法と同様にラベルを階層的にクラスタリングし、平衡二分探索木を作成する。

次に Deep Neural Matching component は、SLI で作られた各クラスタに対して Transformer モデルを微調整し、テキストのラベルクラスタへのより良いマッピングを実現する。具体的には、Transformer を用いて入力されたインスタンスの特徴を求め、他の XMC 問題の研究 [9] でよりよい順位付け性能を示した二乗ヒンジ損失を用いた線形分類器で計算し、以下の式で表される損失関数が小さい値になるクラスタに割り当てる。

$$\frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K \max(0, 1 - \tilde{M}_{ik} g(\mathbf{x}, k; \mathbf{W}, \theta))^2$$

$$s.t. g(\mathbf{x}, k; \mathbf{W}, \theta) = \mathbf{w}_k^T \phi_{\text{transformer}}(\mathbf{x})$$

ここで、 $\tilde{M}_{ik} = 2M_{ik} - 1 \in \{-1, 1\}$  で、 $\tilde{M}_{ik}$  はインスタンスとクラスタの割り当て行列である。また、 $\mathbf{W} = [w_1, \dots, w_K]^T \in R^{K \times d}$  は線形分類器、そして、 $\phi_{\text{transformer}}(\mathbf{x}) \in R^d$  は Transformer からの埋め込みである。モデルの fine-tuning には、3つの最先端の Deep Transformer モデルである BERT, RoBERTa, および XLNet が用いられる。

最後に Ensemble Ranking component は、予測されたクラスタ内のラベルをランク付けするためにクラスタと Transformer の出力に対して訓練がなされ、さらなる性能向上のために各クラスタから得られた予測スコアをまとめ、インスタンスとラベルの関連性を求める。ここではすべてのラベルに対してテキストとの関連性を求めるのではなく、予測上位のクラスタ内のラベルに対してのみ関連性を求めることで、計算コストを削減している。

### 3. 提案手法

#### 3.1 提案手法の概要

まず最初に前処理として、ユーザの訪問履歴からユー

ザの分散表現を獲得し、そして店舗の分散表現とクラスタの分散表現を獲得する。その後、ユーザの分散表現とクラスタの分散表現を用いてユーザとクラスタの関係を NeuMF[10] にて学習する。

あるユーザの訪問予測を行う際は、学習済のモデルにそのユーザの分散表現を入力し、各ユーザに対してクラスタの順位付けを行う。そして、上位のクラスタ内に含まれる店舗のみを訪問予測対象とし、そのユーザに対して店舗の順位付けを行い、上位の店舗を訪問予測の結果とする。店舗の順位付けの際には、ユーザの分散表現と店舗の分散表現の類似度を考慮する。分散表現の類似度には、コサイン類似度を用いる。2つのベクトル  $\vec{x}$  と  $\vec{y}$  のコサイン類似度の式は以下のように定義される。

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\vec{x}}{|\vec{x}|} \cdot \frac{\vec{y}}{|\vec{y}|} = \frac{\sum_{j=1}^m x_j y_j}{\sqrt{\sum_{j=1}^m x_j^2} \cdot \sqrt{\sum_{j=1}^m y_j^2}}$$

#### 3.2 分散表現獲得

ユーザの分散表現を獲得するために、まずはユーザの過去に訪れた場所を単語として扱う。4章の評価実験で示すように、店舗名を単語として扱う場合と、店舗のカテゴリ名を単語として扱う場合の2つの異なるケースを考える。そして、ユーザの過去の訪問場所の系列である訪問履歴をテキストとみなす。このテキストを日本語で事前学習済の BERT を用いて 768 次元のベクトルに変換し、ユーザの分散表現を獲得する。この流れを図 2 に示す。

ユーザの分散表現を獲得後、各店舗ごとの訪問ユーザの分散表現の平均をとり、各店舗の分散表現とする。そしてそれらを用いて店舗のクラスタリングを行う。各クラスタに所属する店舗の分散表現の平均をとり、それらをクラスタの分散表現とする。

#### 3.3 クラスタリング手法

本節では、分散表現を与えた後の店舗のクラスタリング手法について述べる。代表的なクラスタリング手法の1つである K-Means[11] を用いてクラスタリングを行うと、各クラスタに含まれるデータの数は考慮されないため、大量のデータが含まれているクラスタやデータが1つしか含まれていないクラスタが作成され、クラスタ内のデータ数の偏りが発生してしまう。本研究では X-Transformer と同様に Balanced K-Means[12] を用いて、クラスタ内のデータ

数の偏りが発生しないようにした。

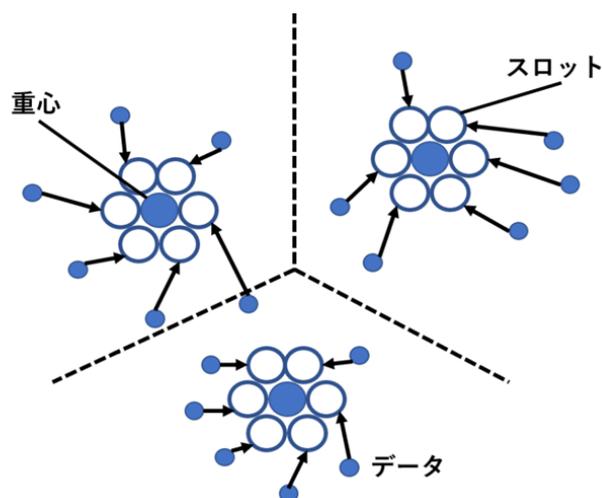


図3 balanced k-means の流れ

Balanced K-Means は K-Means ベースのクラスタリング手法であり、K-Means とは異なり各クラスタに含まれるデータの数がほぼ均等になるようにクラスタリングを行う。均等にクラスタリングを行うためにデータポイントと同じ数だけのスロットを用意し、そのスロットを各クラスタに数が均等になるように与え、データをこのスロットに割り当てる。スロットへの割り当てを図3に示す。データを  $d$  個、クラスタ数を  $c$  個としたときは  $d$  個のスロットを用意し、 $(d \bmod c)$  個のクラスタに  $\lceil \frac{d}{c} \rceil$  個、 $c - (d \bmod c)$  個のクラスタに  $\lfloor \frac{d}{c} \rfloor$  個のスロットを振り分け、すべてのクラスタのデータ数を均等にする。次にクラスタリングのアルゴリズムを説明する。

まずデータ数と同じ数のスロットを用意し、データとスロットからなる二部グラフを用意する。このとき、各データとスロットを結ぶ辺の重みは、各データの位置とスロットが振り分けられているクラスタの重心の位置の二乗距離である。各クラスタの初期の重心の位置は  $d$  個のデータの中からランダムに割り当てる。次に、これらの重みの総和が最小となるデータとスロットの組み合わせを求めるために、各データとスロットを結ぶ辺の重みを  $d \times d$  行列に入れハンガリー法 [13] を用いる。ハンガリー法については後に詳しいアルゴリズムの説明を行う。重みの総和が最小となる組み合わせを求めた後、それぞれのクラスタに割り当てられたデータの平均をとり、重心の位置を計算する。その後、各データとスロットを結ぶ辺の重みも再計算する。この流れをクラスタの重心の位置が変わらなくなるまで繰り返す。

ハンガリー法とは、割当問題の解法の1つで、行列を用いて最も効率的な割り当てを考える手法である。割当問題とは、全員に仕事を割り当てるときに最も効率が良くなる割り当て方を見つける問題である。まず、行列の各行の各

要素からそれぞれの行の最小値を引く。各列に対しても同様の処理を行う。すべての行と列で最小値を引き終えた後、各行各列から0を1つずつ選ぶことができた場合は、その0の位置が割当案となる。もし選ぶことができない場合は、すべての0を含む最小の行と列の組を選び、それらに含まれていない要素から、その最小の要素を引き、選んだ行と列が交わる部分の要素にその最小の要素を加える。そして再び各行各列から0を1つずつ選ぶことができるかどうかを確かめる。

### 3.4 NeuMF の概要

NeuMF [10] は、線形モデルの Generalized Matrix Factorization (GMF) [10] と非線形モデルの multi-layer perceptron (MLP) [10] の2つを融合させた、ユーザの過去の購買履歴などの行動からアイテムに対するユーザの好みを表す協調フィルタリングのためのモデルである。本研究では、ユーザの過去の訪問履歴からクラスタに対するユーザの好みを学習するために NeuMF を用いた。以下、GMF と MLP を説明し、NeuMF の構造を説明する。

#### 3.4.1 GMF

GMF は、推薦分野で利用されるモデルベース協調フィルタリングの代表的なアルゴリズムの1つである Matrix Factorization (MF) を一般化したものである。MF は、共通の潜在空間に投影した特徴を表すベクトルを用いてユーザやアイテムを表現し、それを入力としてユーザとアイテムの関係の予測スコアを求める。予測スコアはユーザとアイテムのベクトルの内積で表される。このとき、予測スコアは以下の式で表される。

$$\hat{y}_{ui} = a_{out} (\mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_i))$$

ここで、 $\hat{y}_{ui}$  は予測スコア、 $\mathbf{p}_u$ 、 $\mathbf{q}_i$  はそれぞれユーザ  $u$  とアイテム  $i$  の埋め込みベクトルを表す。また、 $\odot$  はアダマール積、 $a_{out}$ 、 $\mathbf{h}$  はそれぞれ恒等写像、全要素を1とした一様ベクトルを表す。 $a_{out}$  をシグモイド関数や tanh 関数のような別の活性化関数に一般化し、そして、 $\mathbf{h}$  をデータから学習することができる重みの列として設定することで、MF をより表現力のある GMF に一般化することができる。

#### 3.4.2 MLP

MLP は、multi-layer feed-forward neural network であり、GMF のようにユーザとアイテムのベクトルの内積を求めてユーザとアイテムの関係を単純に表すのではなく、層を追加することでより複雑なユーザとアイテムの関係を学習することができる。MLP のモデルは以下のように定義される。

$$\mathbf{z}_1 = \phi_1 (\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}$$

$$\phi_2 (\mathbf{z}_1) = a_2 (\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_2)$$

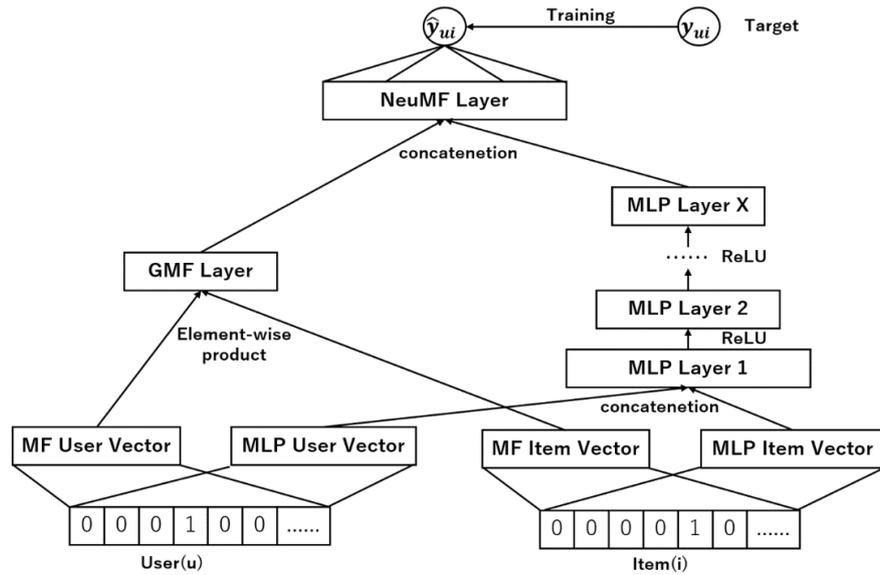


図 4 NeuMF の構造

...

$$\phi_L(\mathbf{z}_{L-1}) = a_L (\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L)$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \phi_L(\mathbf{z}_{L-1}))$$

ここで、 $\mathbf{W}_x$ ,  $\mathbf{b}_x$  そして  $a_x$  はそれぞれ  $x$  番目の層の重み行列、バイアスベクトル、活性化関数である。

ReLU は非飽和で疎なデータに適していることが証明されており [14], また、過学習の可能性も低下させるため、NeuMF では活性化関数として ReLU を利用している。

### 3.4.3 NeuMF

ユーザとアイテムの関係の特徴を表現するために線形カーネルを用いた GMF と、データからユーザとアイテムの関係を学習するために非線形カーネルを用いた MLP を組み合わせることを考える。簡単な方法は、GMF と MLP で同じ埋め込みを利用し、出力を組み合わせることである。しかし、同じ埋め込みを使用すると、この 2 つを組み合わせたモデルの性能が制限される可能性がある。そのため、NeuMF では、GMF と MLP でそれぞれ別の埋め込みを用いて学習し、得られる出力を最後の隠れ層で組み合わせている。この構成により、MLP と GMF が互いに補強しあうため、NeuMF では複雑なユーザとアイテムの関係のより効果的な学習が可能になる。図 4 に NeuMF の構造を示す。モデルは次式のように定式化できる。

$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G$$

$$\phi^{MLP} = a_L \left( \mathbf{W}_L^T \left( a_{L-1} \left( \dots a_2 \left( \mathbf{W}_2^T \left[ \begin{array}{c} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{array} \right] + \mathbf{b}_2 \right) \dots \right) \right) + \mathbf{b}_L \right)$$

$$\hat{y}_{ui} = \sigma \left( \mathbf{h}^T \left[ \begin{array}{c} \phi^{GMF} \\ \phi^{MLP} \end{array} \right] \right)$$

ここで、 $\mathbf{p}_u^G$ ,  $\mathbf{p}_u^M$  はそれぞれ GMF と MLP におけるユーザ埋め込みを表す。同様に  $\mathbf{q}_i^G$ ,  $\mathbf{q}_i^M$  はアイテム埋め込み、 $\hat{y}_{ui}$  は NeuMF における予測スコアを表す。本研究では、GMF と MLP どちらに対しても埋め込みも同じユーザの

分散表現とクラスタの分散表現を用いた。また、ロス関数として以下の式で表される平均二乗誤差を用いて、これらを最適化するように学習を進めた。

$$\text{MSE} = \frac{1}{n} \sum_{k=1}^n (\hat{y}_{ui} - y_{ui})^2$$

ここで、 $n$  はデータの数で、 $\hat{y}_{ui}$  は予測スコア、 $y_{ui}$  は実際のスコアである。

## 4. 評価実験

提案手法を用いた訪問予測の有効性を検証するために、実世界で収集された実データを用いて実験を行った。4.1 節で評価に用いたデータセットの詳細や評価指標、比較手法などの評価環境について述べ、4.2 節では実験結果とその結果に対する考察について述べる。

### 4.1 評価環境

以下ではまず、本実験に用いたデータセットについて述べ、評価方法について述べる。

#### 4.1.1 データセット

本実験では、実世界で Wi-Fi を用いて収集された訪問データを用いた。訪問データでは、Wi-Fi アクセスポイントが設置されている場所でユーザがその Wi-Fi ポイントにアクセスしたタイミングで記録がとられ、ユーザの ID、アクセス時間、および場所の位置情報やカテゴリなどが記録される。本実験では、ある場所に設置されている Wi-Fi アクセスポイントにアクセスしたことをその場所に訪問したこととしている。また、ユーザの分散表現を作成する際は、ユーザごとの訪問の記録すべてを参照しているが、予測対象とする場所はスーパーやショッピングモールなどの店舗のみに絞り、駅などの店舗以外の場所は予測対象には

含まない。提案手法による訪問予測の精度を評価するため

表 1 データセットの統計情報

	学習データ	テストデータ
ユーザ数	1,292,617	1,199,200
店舗数	49,597	48,220

本実験では、学習データとして2020年5月1日から2020年5月14日までに収集されたデータのうち、十分な訪問履歴を持つユーザのデータを用いた。具体的には、10回以上の訪問情報を持つユーザのみを用いた。そして、テストデータとして2020年5月15日から2020年5月21日までに収集されたデータを用いた。テストデータでは、学習データに含まれていないユーザと店舗は取り除いた。本実験で用いたデータセットの統計情報を表1に示す。

#### 4.1.2 評価手順

本実験では、学習データのうち2020年5月1日から5月7日までに収集されたデータを利用してユーザと店舗の分散表現を作成し、店舗を1024個のクラスタに分割した。これらの店舗を本実験での予測対象とする。予測対象の店舗で十分な訪問者数を持つ店舗はほとんど存在せず、訪問者数が100未満の店舗は約25000店と過半数を占めていた。また、ユーザの分散表現作成のためのユーザの訪問履歴は、ユーザの訪れた店舗の名前を羅列した店舗名ベースのものとその店舗のカテゴリを羅列したカテゴリベースのもの2種類を用意した。ここで、店舗名には、チェーンストアの場合はチェーン名(例:〇〇ドラッグ)、店名については「店名(例:〇〇駅前店)」が含まれている。

2020年5月8日から5月14日までのユーザの訪問データを利用してユーザとクラスタの関係を学習した。そしてテストデータに含まれるユーザのうち、ランダムに選んだ1500人を対象に訪問予測を行い、提案手法の訪問予測精度の評価を行った。具体的には、まずユーザに対してクラスタを順位付けし、上位100個のクラスタに含まれる店舗のみを訪問予測対象とする。それらの店舗に対して順位付けを行い、Top-kの予測店舗のリストを作成する。

そして、テストデータにおけるユーザが訪問した店舗のリストとTop-kの予測店舗リストの重複した店舗数により、提案手法の性能を計測した。評価指標にはHit Rate@k (HR@k)を用いた。HR@kは、あるテストユーザ*i*に対して予測店舗のリストの上位*k*個の中にユーザの訪問した店舗のリストの店舗と同じ店舗が含まれているかどうかの精度を測る指標である。HR@kの式は以下のように定義される。

$$HR@k = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} \frac{1}{N_{shop}} \sigma(t_i \in S_i(k))$$

ここで、 $N_{shop}$ はユーザが訪問した店舗の総数、 $N_{te}$ はテストユーザの総数である。また、 $t_i$ はテストユーザ*i*が訪

問した店舗の集合を表し、 $S_i(k)$ はテストユーザ*i*に対する予測店舗の上位*k*個の集合を表す。 $\sigma(t_i \in S_i(k))$ は集合同士の重複した要素の個数を返す関数であり、 $S_i(k)$ の中に $t_i$ の要素がどれだけ含まれているかを返す。

比較手法として、クラスタリングを用いない手法と、店舗を訪問ユーザが多い順に並べた店舗リストを推薦するTop Popularの2つを用いた。クラスタリングを用いない手法では、提案手法と同様に店舗名ベースおよびカテゴリベースでユーザと店舗の分散表現を獲得し、その後、クラスタリングを行わずに、直接、ユーザと店舗の訪問を予測する2つのケースを用いた。

#### 4.2 実験結果

テストデータに含まれるユーザのうち、ランダムに選んだ1500人に対する訪問予測の結果を図5に示す。図5は、上位*k*個の予測店舗リストの*k*を1から100まで変化させた場合の性能の変化を表している。

この図において、図中の1vs1はクラスタリングを用いない手法を表している。なお、この図では、クラスタリングを用いない2つの手法とランダムに選ぶ手法、そしてTop Popularの4つの結果がほぼ重なっている。この結果から、すべての*k*においてクラスタリングを行う提案手法が比較手法より高い訪問予測精度を達成していることがわかる。これは、各ユーザに対して店舗クラスタ予測を行い訪問予測対象となる店舗の削減を行った際に、よりユーザの好みであると考えられる店舗クラスタが予測上位に来たためであると考えられる。このことから、ユーザの分散表現から作成された店舗のクラスタは、ユーザの好みかそうでないかがうまく現れた店舗の集まりとなっていると考えられ、同じ店舗を訪れるユーザの好みは似たものになっていることが分かる。また、クラスタリングを用いずに全店舗を対象に訪問予測を行う手法が、ランダムに*k*個選び予測結果とする手法と精度があまり変わらないことから、大量の店舗がある場合のユーザの好みは、ユーザの訪問場所を直接予測する単純な手法では予測が難しいことがわかる。したがって、店舗のクラスタリングを行って予測対象を絞って訪問予測を行うことは、予測精度の向上に対して有効であると考えられる。

さらに、予測を行う際、クラスタリングを行わない1vs1の場合はGMF layerによるベクトルの要素ごとの積の計算とその他の部分での内積計算がそれぞれユーザ数*U*×全店舗数*S*の合計*US*回必要になるが、今回クラスタリングを行っているためそれぞれ*U*×クラスタ数*K*( $K \ll S$ )の*UK*回となり、かなり計算量の削減に繋がっている。また、クラスタリングを行った際には上位クラスタ内の店舗とユーザのコサイン類似度の計算を追加する必要があるが、その回数は*U*×上位クラスタ内の店舗数*S'*( $S' \ll S$ )の計*US'*回であり、クラスタリングを行わない場合よりも

計算量はかなり少なくなっている．実際，筆者らの計算環境では，1回の予測に要する時間が提案手法は1vs1の比較手法よりも8倍以上高速であった．

また，提案手法で用いる訪問履歴において店舗名ベースとカテゴリベースのものを組み合わせた実験も行った．具体的には店舗名ベース（カテゴリベース）でクラスタ予測を行った後，上位のクラスタ内の店舗の順位付けにはカテゴリベース（店舗名ベース）のベクトルを用いる．この結

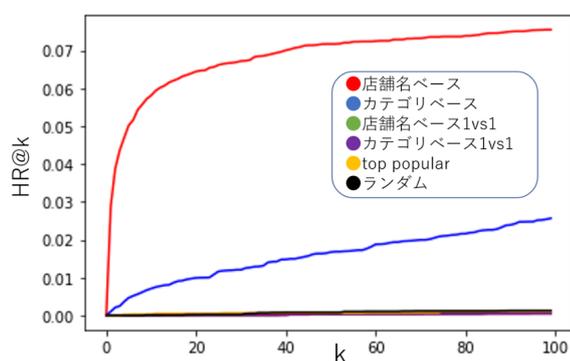


図5 提案手法とクラスタリングを用いない手法の比較

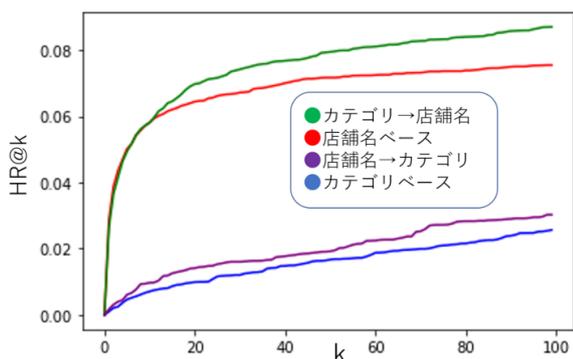


図6 店舗名，カテゴリベースの手法とそれらを組み合わせた手法の比較

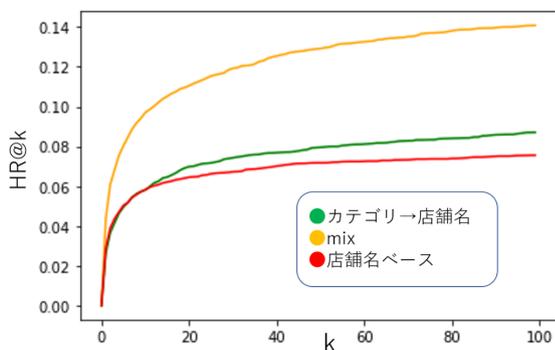


図7 上位2個の手法とそれらを混ぜた手法の比較

果を図6に示す．店舗名ベースのみ，カテゴリベースのみの結果では店舗名ベースがよりよい精度を達成し，カテゴ

リベースでクラスタ予測までを行いその後店舗名ベースで店舗の順位付けを行ったものが一番良い精度を達成していることから，ユーザの店舗の好みはカテゴリレベルで固まっているためクラスタリングを行う際にはカテゴリベースのほうが店舗名ベースよりもよりユーザの好みの店舗が固まると考えられる．また，ユーザと店舗の関係は大まかなカテゴリベースよりも細かい店舗名ベースのほうがうまく捉えられると考えられる．

さらに，先ほどの実験で上位2個の精度を達成した手法を組み合わせる実験も行った．具体的には，Top-kの予測を行う際に店舗名ベースの予測結果とカテゴリベースでクラスタ予測までを行いその後店舗名ベースで店舗の順位付けを行った場合の予測結果それぞれの上位k個を取り出し，それらを合わせたもので再度順位付けを行う．その結果を図7に表す．この手法は図のmixに対応しており，一番高い精度を達成している．

以上の3つの実験より，店舗名ベースとカテゴリベースを上手く組み合わせることが精度向上のカギになっていると考えられる．

## 5. 結論

本研究では，ユーザの訪問履歴を用いた店舗の分散表現を基に店舗のクラスタリングを行い，ユーザの訪問予測を行う手法を提案した．提案手法では，まず実店舗へのユーザの訪問履歴を基にユーザの分散表現を作成した．そして，ある店舗に訪れたユーザの分散表現の平均を求めて店舗の分散表現を作成し，それを用いて店舗のクラスタリングを行った．ユーザと店舗の分散表現に基づいた店舗のクラスタリングにより，規模の小さい店舗の情報を補うことができる．また，クラスタを作成する際に各クラスタ内に含まれる店舗の数を均等にする事で数の偏りを解消し，予測対象を上手く絞ることができる．そして，クラスタを作成した後，ユーザとクラスタの分散表現を用いてユーザとクラスタの関係をNeuMFで学習した．この学習したモデルを用いてユーザに対してクラスタの順位付けを行い，上位のクラスタ内の店舗を対象に訪問予測を行った．

提案手法の有効性を検証するために，実データを用いて提案手法による訪問予測を行った．その結果，ほかの全ての比較手法よりも提案手法は高い訪問予測精度を示すことを確認した．

今後の課題としては，店舗名ベースとカテゴリベースの予測を上手く組み合わせ予測精度を向上させる手法を検討する．

## 謝辞

本研究の一部は，JST CREST (JPMJCR21F2) の研究助成によるものである．

## 参考文献

- [1] Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y. and Dhillon, I. S.: Taming pretrained transformers for extreme multi-label text classification, *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*, pp. 3163–3171 (2020).
- [2] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [3] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V.: Xlnet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237* (2019).
- [4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach, *arXiv preprint arXiv:1907.11692* (2019).
- [5] You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H. and Zhu, S.: Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification, *arXiv preprint arXiv:1811.01727* (2018).
- [6] Jain, H., Balasubramanian, V., Chunduri, B. and Varma, M.: Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches, *Proc. ACM Int'l Conf. on Web Search and Data Mining*, pp. 528–536 (2019).
- [7] Prabhu, Y., Kag, A., Harsola, S., Agrawal, R. and Varma, M.: Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising, *Proc. World Wide Web Conference*, pp. 993–1002 (2018).
- [8] Khandagale, S., Xiao, H. and Babbar, R.: Bonsai: Diverse and shallow trees for extreme multi-label classification, *Machine Learning*, Vol. 109, No. 11, pp. 2099–2119 (2020).
- [9] Yen, I. E., Huang, X., Dai, W., Ravikumar, P., Dhillon, I. and Xing, E.: Ppdspare: A parallel primal-dual sparse method for extreme classification, *In Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 545–553 (2017).
- [10] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.-S.: Neural collaborative filtering, *In Proc. Int'l Conf. on World Wide Web*, pp. 173–182 (2017).
- [11] MacQueen, J. et al.: Some methods for classification and analysis of multivariate observations, *In Proc. Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, No. 14, Oakland, CA, USA, pp. 281–297 (1967).
- [12] Malinen, M. I. and Fränti, P.: Balanced k-means for clustering, *Joint IAPR Int'l Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 32–41 (2014).
- [13] Kuhn, H. W.: The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, Vol. 2, No. 1-2, pp. 83–97 (1955).
- [14] Glorot, X., Bordes, A. and Bengio, Y.: Deep sparse rectifier neural networks, *In Proc. Int'l Conf. on Artificial Intelligence and Statistics*, pp. 315–323 (2011).