

IoT 技術を用いた展示推薦システムと実験

筒井 道紀[†] 水野 陽平[‡] 鬼塚 真[‡]

[†] 大阪大学工学部 〒565-0871 大阪府吹田市山田丘 2-1

E-mail: [†] {tsutsui.michinori,mizuno.yohei,onizuka}@ist.osaka-u.ac.jp

あらまし 本稿では IoT 技術を用いた推薦システムを構築し、実証実験を実施した結果について報告する。本推薦システムでは、iBeacon を利用してユーザーがどの展示にどれくらいの時間滞在したかという履歴情報を収集し、行列分解の技術を用いてユーザーにおススメの展示を推薦するシステムである。本実験によって、推薦の精度の目安となる予測滞在時間の誤差が平均して 2 分以内には抑えられていることが明らかになった。また、IoT 技術を用いた推薦システムには、ユーザー数を増やす必要があることや、推薦の与える影響の大きさの点で課題があることが明らかになった。

キーワード 推薦, 協調フィルタリング, IoT

1. はじめに

Amazon に代表されるネットショッピングでは膨大な数の商品を取り扱っている。消費者の立場から考えると、これらの商品をすべて見比べることは不可能である。一方、商品を提供する側の立場から考えると、商品の売り上げを向上するためには、消費者個人に合わせて商品を推薦することが重要である。そこで Amazon では過去の閲覧履歴や購入履歴などを用いて、個人に合わせた推薦を行うことで、消費者に適切な商品を提示して企業の売り上げの向上を図っている[1][5]。

現実世界の推薦サービスを見てみると、ウェブの世界とは異なり、過去の行動履歴を用いた推薦がなされていないという問題がある。現在、IoT 技術の普及に伴い、センサやアクチュエータを利用した新たなサービスが実現可能になりつつある[13]。IoT とはパソコンなどの通信機器だけでなく、実世界におけるさまざまな「もの」に通信機能を持たせ、インターネットに接続することでサービスを実現する仕組みのことである。実際にはセンサなどを利用して現実世界から情報を取得し、得られた情報を用いてアクチュエータなどで「もの」を制御して実世界にフィードバックを行う。IoT という考え方自体は昔から存在していたものだが、センサ・アクチュエータ技術の進歩によって様々な情報を収集したり、機器を操作できるようになったことや、ビッグデータ技術によって大量の情報を自動で処理できるようになったため、現在注目を集めている。

上記のような背景に基づき、我々は IoT 技術を利用することで、個人の行動履歴などの情報を収集し、現実世界で個人に合わせた推薦を実現するシステムを構築した。また本システムを用いて、大阪にある研究展示施設である The Lab において実証実験を実施した結果について報告する。本システムでは、推薦には協調フィルタリングと呼ばれる、ユーザーの行動履歴をも

とに推薦を行う方法[1-5,7]を用いる。利用者の行動履歴の取得には、iBeacon[9]と呼ばれる技術を用いる。iBeacon を用いて、The Lab を訪れるどのユーザーがどの展示にどの程度の時間滞在したかを行列として表現し、行列分解の技術を適用することで、行列における未知の値を推定する。つまり、利用者が未訪問の展示にどの程度の時間滞在するかを推定することで、より長い時間滞在すると推定された展示を推薦する。

本稿の構成は次のとおりである。2 章で IoT を用いた推薦システムに必要な事前知識について解説する。3 章では、今回構築したシステムについて述べる。4 章で実験の結果や考察について述べた後、5 章で関連研究について述べ、最後に 6 章で本稿のまとめと今後の課題について述べる。

2. 事前知識

本章ではユーザーの行動履歴を収集して推薦を行うシステムに必要な技術について述べる。2.1 節では利用者の行動履歴の収集に利用した iBeacon について説明する。2.2 節では推薦のアルゴリズムについて説明する。2.3 節では実験で用いたアクチュエータを制御する機器について説明する。

2.1 iBeacon

iBeacon とは Bluetooth を利用して、機器を特定するビーコンと呼ばれる技術の一種である。Bluetooth を用いて常に情報を発信し続けており、その情報は iPhone など受け取り処理することで、情報の発信源の機器を認識することができる。情報を発信する機器のことをビーコンと呼ぶこともある。本稿では機器を特定する技術総体を iBeacon と呼び、機器のことをビーコンと呼んで区別することとする。

2.2 Matrix Factorization

代表的な推薦手法の一つとして協調フィルタリングが挙げられる。協調フィルタリングではユーザーの行動履歴から得られるユーザーとアイテムの関係情報を用いて推薦を行う。協調フィルタリングを行うための代表的な手法として SVD が挙げられる [1]。SVD は、ユーザーとアイテムの関係を表した行列を入力として、その行列を分解することでユーザーやアイテムの特徴を抽出する技術であり、推薦に応用することができる。SVD は特徴を抽出する有効な方法だが、スパースな行列に対してうまく適用できないという問題が指摘されている [1]。この問題を解決する方法として、Matrix factorization の技術が提案されている [1,2]。SVD では行列の欠損値を含めて行列分解を行っていたため、スパースな行列に対して精度が上がらない問題があったが、Matrix factorization では欠損値を除いて行列分解を行うことで、ユーザーやアイテムの特徴を抽出する。

具体的には Matrix factorization では以下のような式を最小化することで予測モデルの学習を行う。

$$\min_{q^*, p^*, b^*} \sum_{(u,i) \in \kappa} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2) \quad (2.1)$$

ここで u はユーザー、 i はアイテム、 κ は u と i の組の中で r_{ui} が存在するものの集合、 μ は滞在時間全体の平均、 q_i はアイテムの特徴を表すベクトル、 p_u はユーザーの特徴を表すベクトル、 b_i と b_u はそれぞれ、アイテム、ユーザーの影響による偏差、 λ は過学習を抑える度合いを決める定数である。 q_i と p_u の次元は特徴の数を表し、 q_i と p_u の各要素はユーザーやアイテムの特徴に対応する。

ユーザー、アイテムの影響による偏差はユーザーやアイテムの評価の傾向を考慮するために導入されている。例えば、ユーザーが他の人に比べて批判的な人だった場合、評価値は低くなる傾向がある。これがユーザーの影響による偏差である。また、あるアイテムに対する評価が高い場合は、その評価に引きずられて高めの評価をしてしまう傾向がある。これがアイテムの影響による偏差である。

得られた予測モデルを用いて、予測評価値 $\hat{r}_{ui}$ を求めるには次の式を用いる。

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (2.2)$$

式(2.2)より評価値を求めるには q_i と p_u と b_i と b_u を式(2.1)を最小化することで求めればよい。式(2.1)の第 1 項ではすでに存在している評価値と予測評価値の誤差をとって 2 乗している。これを最小化するということは、既に存在する評価値と予測評価値との誤差を最小化するということである。つまり、与えられたデータにできるだけ合うようなベクトルを求めている。第 2 項は過学習を防ぐための項である。

式(2.1)を最小化するためには以下の式を繰り返し適

用する。

$$q_i \leftarrow q_i + \gamma(e_{ui}p_u - \lambda q_i) \quad (2.3)$$

$$p_u \leftarrow p_u + \gamma(e_{ui}q_i - \lambda p_u) \quad (2.4)$$

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \quad (2.5)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i) \quad (2.6)$$

ここで e_{ui} は正解の評価値と予測評価値の誤差、 γ は定数である。式(2.3)～(2.6)の右辺の q_i と p_u と b_u と b_i は 1 回前の繰り返しで求めたものを用いる。

2.3 IRKit と Hue

IRKit シンプルリモコン（以下 IRKit）とは WiFi 経由で制御可能な赤外線リモコンデバイスである。専用アプリを用いて家電製品などのリモコンの赤外線信号を登録すれば、さまざまなもののリモコン操作を IRKit で代用できるようになる。IRKit 自体に HTTP サーバが搭載されているため、iPhone などの端末から JSON 形式の赤外線情報を HTTP POST リクエストにのせて IRKit へ送ることで、IRKit がリクエストを解釈して、適切な赤外線信号を発信することができる。

Hue とは、WiFi を経由して操作することのできる照明のことである。1 つの電球でさまざまな色を表現できるだけでなく、明るさを調節したり点滅させたりすることもできる。Hue を操作するには、ブリッジと呼ばれる、照明と端末のかけ橋のような役割を果たすものを用いる。ブリッジから Hue に信号を送ることで Hue の色や明るさを制御する。Hue も IRKit と同じく HTTP 通信で、ブリッジに、JSON 形式の情報を送ることで操作することができる。

3. 展示推薦システム

本章では、我々が開発した展示推薦システムについて解説する。本推薦システムでは、ユーザーが展示に滞在する推定時間が長いと推定される展示を推薦する。つまり式(2.1)のアイテムとして展示を用い、評価値として滞在時間を用いる。各ユーザーがどの展示にどれくらいの時間滞在したかを展示推薦システムに入力すると、各ユーザーに訪れたことのない展示の滞在時間を予測し、予測滞在時間の長い展示を推薦する。本推薦システムの処理の流れは次のようになる。最初にユーザーをシステムに登録する受付処理を行う (3.1 節)。その後、ユーザーに端末を持って展示会場を移動してもらい、移動中のユーザーの位置情報を取得する (3.2 節)。収集したユーザーの移動履歴を入力として推薦モデルの構築および推薦処理を行い (3.3 節)、最後に、ユーザーからの要求に合わせて推薦結果を端末に提示する (3.4 節)。各工程について以降の節で詳述する。

3.1 受付処理

受付場所を設け、受付フォームにユーザーの属性情

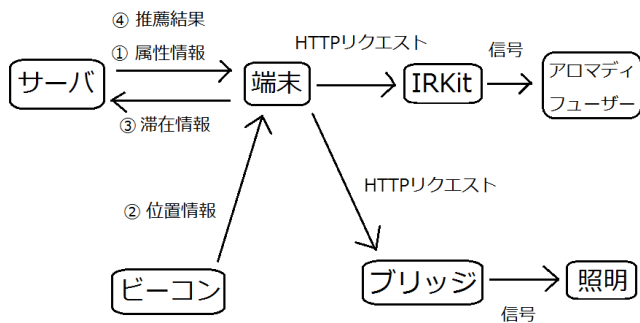


図 1：システムの構成

報（年齢、性別、好きな果物）と貸し出す端末の番号を入力する。ユーザーの属性情報を用いて、ユーザーに対して提供される照明の色とアロマの香りを決定する。ユーザー登録の際にユーザー番号を発行して、ユーザー番号と対応する端末の対応付けを行う。受付処理において、サーバに記録されるデータは、端末の番号、その端末を現在持っているユーザーのユーザー番号、属性情報である。端末ではこのファイルを読み込むことでユーザー番号、属性情報を取得する（図1の①）。

3.2 位置情報の取得

ユーザーがどの展示に近づいているかを区別するため、実験に先立ってビーコンを実験会場の各展示に配置し、実験の際にユーザーに端末を持ち歩いてもらう。ビーコンの識別子として iBeacon で規定されている major を用いる。端末ではビーコンから major に関する情報を受け取り、どの展示の近くに端末があるかを判断する（図1の②）。

ユーザーの現在地は最も強い電波を発するビーコンを用いて決定する。ビーコンの発信する情報の1つに rssi がある。これはビーコンとユーザーが所持している端末との距離の目安となる値である。端末が複数のビーコンから電波を受け取ると、受け取ったすべての電波の rssi の値を比較し、最も大きい rssi の値を発信していたビーコンに近づいたと判断する。

また、特定のビーコンにユーザーが近づくと、照明を点灯させたりアロマディフューザーを起動したりすることで、ユーザーの属性に応じて適宜、周辺環境を変化させる機能を実現した。さらに、端末とビーコンが離れるとき（rssi の値が一定値以下になったとき）に、接近してきた時刻、端末の番号、端末の ID、ビーコンの major、年齢、性別、ビーコンの近くにいた滞在時間（単位は秒）、を記録し、csv ファイルとしてサーバにアップロードする（図1の③）。また、特定のビーコンではこのとき、アロマディフューザーを止めたり、照明を消したりする。

オススメを更新

12時30分の推薦結果 赤：現在位置 緑：オススメ

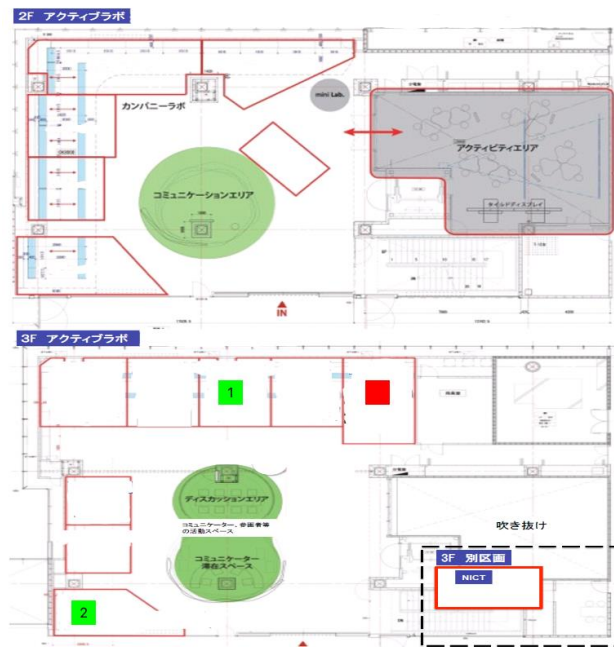


図 2：推薦画面

3.3 推薦処理

サーバでは端末から送信された滞在情報を統合して、入力データを作成し、推薦プログラムに入力する。推薦プログラムは1分ごとに定期的に実行し、推薦結果を更新する。推薦結果をあらかじめ作成しておくことで、推薦結果をユーザーの端末に表示するまでの計算時間を短縮する。

3.4 推薦結果の提示

推薦の結果はサーバから端末側に送信し、推薦結果を端末に置いて提示する（図1の④）。端末に置ける実際の推薦アプリの画面を図2に示す。画面には実験会場の地図と「オススメを更新」ボタンをレイアウトしている。地図上には赤い四角でユーザーの現在地が表示される。ユーザーが「オススメを更新」ボタンをクリックすると、推薦処理を実行した時刻と推薦される展示の上位2件が表示される。表示の方法は図2のように地図上におススメの展示の場所が緑の四角で表示される。四角内の数字は推薦の順位である。

4. 実験

本章では、実験の概要と結果、評価について述べる。この実験は、ビーコンを用いた推薦の精度などを調べる目的で行った。4.1 節では実験で使用した計算機環

表 1：年齢と性別による実験参加者の人数（人）

	20 歳未 満	20 代	30 代	40 代	50 代	60 代以 上	年 齢 不 明
男性	4	15	8	14	11	2	0
女性	1	3	7	13	2	0	0
性別 不明	0	0	1	1	0	0	2

境を述べる。4.2 節では実験の概要について述べる。4.3 節で実験における各種設定について説明し、4.4 節では推薦の精度に関して考察する。4.5 節では推薦結果について述べ、4.6 節では推薦の与える影響について考察する。

4.1 計算機環境

推薦と受付処理を行うサーバとして Mac OS X を用いた。Mac OS X の CPU は Intel Core i5 であり、クロック数は 2.8GHz、メモリは 8GB である。推薦プログラムは JAVA1.8.0_51 を用いて実装されている。推薦結果を提示する端末として iPad mini を用いた。iPad mini の OS は iOS 9.1 である。ビーコンからの情報を受け取り、推薦結果を提示するプログラムは swift を用いて実装されており、開発環境として Xcode 7.1.1 を用いた。

4.2 実験概要

iPad mini をユーザーに貸し出し、ビーコンから受け取った情報により、ユーザーがどの展示にどれくらいの時間滞在したかを分析し、ユーザーにお勧めの展示を推薦する。さらに、ユーザーから集めた属性情報からユーザーの好みに合いそうな照明の色やアロマの香りを体験してもらう。

実験は 11 月 21、22、27、28、29 日に行った。実験はナレッジキャピタルの運営する The Lab と呼ばれる、大学や企業の先端技術が展示されている場所で行った[8]。展示数は 14 件である。

実験参加者は合計 77 人であった。そのうち、54 人が男性、26 人が女性、4 人が性別不明だった。また、参加者の年齢は 20 歳未満が 5 人、20 代が 18 人、30 代が 16 人、40 代が 28 人、50 代が 13 人、60 代以上が 2 人、年齢不明が 2 人だった。年齢と性別による実験参加者の人数を表にすると表 1 のようになる。人数が合わないのはダミーのユーザーについてもカウントしていたり、受付時に失敗して受付フォームへの入力を 2 回してもらったりしたため、参加者の人数よりもデータにあるユーザーの人数が多くなっているためである。

4.3 実験設定



図 3：ビーコンの設置場所

連続で 1 分間以上滞在していないデータはノイズと考え、入力データには反映されないようにした。同一の展示に 2 回以上訪れた場合は各滞在時間の合計を滞在時間として用いた。

今回は抽出する特徴の数（アイテムの特徴を表すベクトル q_i とユーザーの特徴を表すベクトル p_u の次元数）は 10 とした。これは 14 個の展示およびユーザーをそれぞれ 10 個の特徴で表すことを意味する。10 という数字は展示の数より少ない数で、きりのいい数字であることから選ばれた。式(2.1)の λ の値は 0.08 を用いた。最小化するための更新式の γ の値は 0.0065 を用いた。更新式を適用する回数は最大で 60 回までとした。これらのパラメータについては[10]に従って設定した。

今回の実験では、ビーコンは Kontakt 社のものを使用した[11]。電波の発信頻度は iBeacon の推奨設定に従い 100ms につき 1 回発信するよう設定した。major の値は 1～26 でありこの値で近づいた場所を識別した。major の値が 1～14 のビーコンは図 3 にあるように展示の真上に配置した。真上に設置したのは、電波の反射や人の体による電波の減衰の影響を少しでも抑えるためである。ただし、展示 4 と展示 13 については展示が設置されている台の上に配置した。これは展示の真上にビーコンを設置するちょうどいいものがなかったためである。展示 4 については展示スペースが広かったため、2 つのビーコンを設置した。ビーコンの major の値は共通の値を設定した。3 つのビーコンは照明にセットし、照明に近づくと明かりが付き、照明から離れると明かりが消えるようにした。受付にもビーコンを配置した。受付に近づくとそこに置いてあるアロマディフューザーが起動し、離れると止まるようにした。

4.4 推薦精度の評価方法

今回用いた推薦プログラムの推薦の精度について

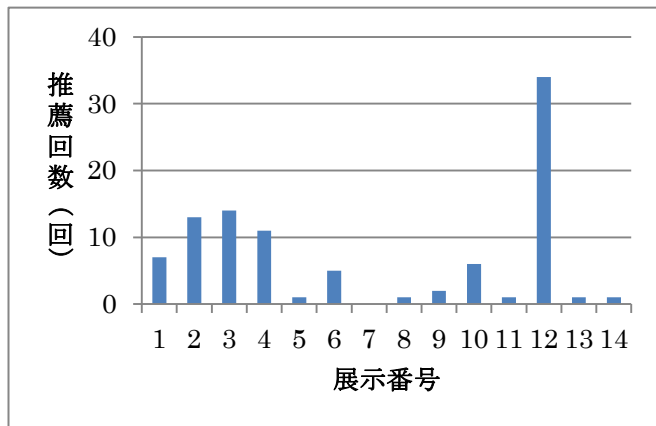


図 4：各展示が実際に 1 位に推薦された回数

考える。推薦の精度を調べるために交差検証を行った。なお、検証に用いた入力データは、滞在時間が 1 分未満のデータも含めたものを使っている。手順は次のとおりである。まず、入力データを 5 つに分割する。次に、分割したデータの 1 つを選び、それを正解データとする。そして、残りの 4 つのデータを用いて学習を行い、予測滞在時間を推定する。その後、予測した滞在時間と正解データにある正しい滞在時間を比較する。次に分割したデータのうち最初に正解データとして選ばなかったデータを正解データとして同様の操作を行う。すべてのデータについて同様に繰り返し、誤差を求め、5 回の平均誤差を計算する。その誤差が小さいほど推薦が正確であったことがわかる。

今回は、推薦の精度の指標として MAE と RMSE を用いた。MAE は誤差の絶対値の平均をとったもので、次のような式であらわされる。

$$MAE = \frac{1}{N} \sum_{(u,l) \in K} |r_{ui} - \hat{r}_{ui}| \quad (4.1)$$

RMSE は誤差の 2 乗の平均の平方根をとったもので、次の式であらわされる。

$$RMSE = \sqrt{\frac{1}{N} \sum_{(u,l) \in K} (r_{ui} - \hat{r}_{ui})^2} \quad (4.2)$$

ここで N は正解データの数である。

交差検証を行った結果、今回実験に用いた推薦プログラムの MAE は 1.651、RMSE は 2.941 となった。

MAE の値から、平均すると予測滞在時間の誤差は 1 分半強に抑えられていると言える。これは滞在時間を予測していると考えれば、問題ない程度の精度が出ていると言える。

4.5 推薦結果の評価

まず初めに、ユーザーに推薦された展示の統計情報を報告する。各展示に関して実験中にユーザーに提示

された推薦結果として 1 位になった回数をグラフにすると図 4 のようになる。横軸の数字は展示番号である。ここでは同じユーザーが複数回「おススメを更新」のボタンを押した場合は、1 人のユーザーに対して複数個の 1 位の展示があることとして数えた。ただし、推薦結果が変わっていなかった場合は 1 回分として数えた。これは短時間の間に何度も推薦結果を更新したユーザーがいたため、結果が変わっていないものもカウントすると、あるときにたまたま推薦されていただけでも、何度も推薦されていることになる展示ができてしまうからである。図 4 を見ると、推薦される展示が大きく偏っていることがわかる。特に 12 番の展示は 2 番目に多く推薦されている 3 番の展示の倍以上の回数推薦されている。これは 12 番の展示には極端に長く滞在したユーザーがいたためだと考えられる。全展示の平均滞在時間を調べてみると約 3 分であった。それに対して、12 番の展示には 20 分近くも滞在したユーザーが存在した。他の比較的推薦されている 2 番、3 番の展示についても同様に 20 分以上滞在しているユーザーが存在していることが確認された。特に 12 番の展示が推薦されているのは 12 番の展示に長く滞在したユーザーが実験の序盤から存在したため、実験序盤で推薦回数をかせいだからだと考えられる。実際に 12 番の展示は実験の初日に 12 回も推薦されている。

また、実験の序盤で何度か推薦結果を確認してみると、たいていの人に同じ場所が推薦されてしまっていた。これは、データの数が少ないために起こったことだと考えられる。データの数が増えない状態で、ある展示に長く滞在したユーザーがいると、その展示の持つ影響力が大きくなり、推薦されやすくなってしまったのだと考えられる。データの数が増えると長く滞在したユーザーの影響力が弱まるかどうかを確かめるために推薦結果を日付ごとに分けて、推薦回数の分散を計算してみると、1 日目から順番に 9.57、2.96、4.27、2.45、8.67 となり、1 日目と 5 日目の偏りが大きいことがわかる。このことから、データの数が増えても必ずしも推薦結果が偏らなくなるわけではないことがわかる。この結果から考えられる可能性は 4 つある。1 つ目はデータの数が増え、長く滞在したユーザーの影響力を弱めるにはまだまだ足りていない、という可能性がある。この場合、対処法としてはデータの数をもっと増やすしかない。2 つ目の可能性は長く滞在したユーザーの影響力が大きすぎてデータを増やしても十分に弱めきれないというものである。この場合、対処法としては滞在時間の最大値をあらかじめ決めておいて、その時間以上滞在したときの滞在時間はすべて最大値にあわせることなどが考えられる。3 つ目はデータの

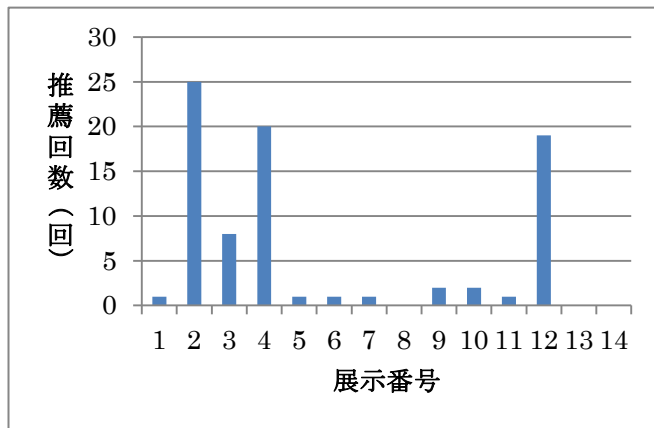


図 5：全履歴データを用いて全ユーザーに対して推薦を実行した結果から、各展示が 1 位に推薦された回数（滞在時間 1 分未満のデータは含まない）

数を増やしても長く滞在したユーザーの影響力は弱まらないという可能性である。この場合も 2 つ目の可能性と同様に、滞在時間を最大値で足切りすることで極端に長く滞在したユーザーの影響力を小さくすることができる。4 つ目の可能性は推薦はすでに適切に行われている、というものである。この場合は何の問題もない。

実験の最初の段階ではデータの量が少ないため適切な推薦が行えない。よって、ユーザーに提示された推薦結果にはノイズとなるデータが含まれている。そこで次に、ある程度データがたまってきた状態でどのように推薦が行われるかを確認する。そのために最終的に集まったデータをすべて用いて推薦を行い、その結果を確認した。最終的に集まったデータをすべて用いて全ユーザーを対象に推薦を行い、1 位になった回数を数えてみると図 5 のようになった。これを見ると、やはり長めに滞在された展示の影響力がまだ大きいことがわかる。しかし、図 5 を見ただけではわからないが、2 位、3 位の展示まで含めてみるとデータがたまってくる前と比較すれば、ユーザーごとに推薦される展示は変わってきている。実験の序盤のデータがたまっていない状態で推薦結果を確認したときには、本当に推薦結果はすべてのユーザーに対してほぼ同じで 12 番、2 番、3 番の展示が推薦されることが多かった。最終的に集まったデータからの推薦では、例えば 7 番の展示は図 5 からほとんど推薦されていないように見えるが、2 位、3 位になった回数も含めると 22 回推薦されている。これはデータの数が増えたことによって、長めに滞在された展示の与える影響が弱まってきていることを示していると考えられる。よって以前に述べた 3 つの可能性のうち 3 つ目の可能性は否定できると

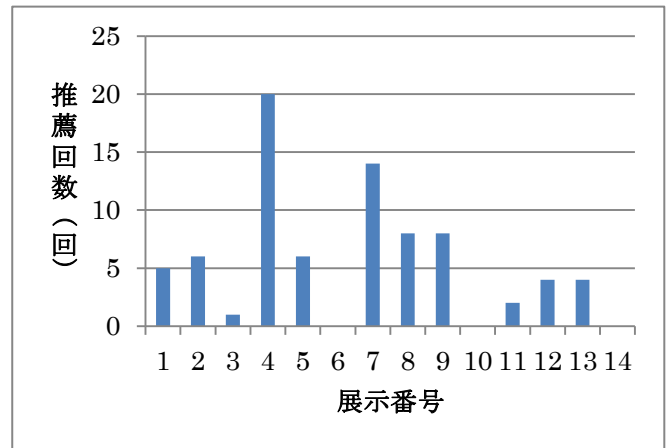


図 6：全履歴データを用いて全ユーザーに対して推薦を実行した結果から、各展示が 1 位に推薦された回数（滞在時間 1 分未満のデータは含まない）

考えられる。しかし、長めに滞在された展示の影響はまだまだ大きい。データの数をもっと増やさなければ正確な推薦はできないと考えられる。

入力データが少ない原因としては、そもそもユーザーの数が少ないということもあるが、理由がもう 1 つ考えられる。その原因とは、連続で 1 分間以上滞在していないデータをノイズとして排除してしまったことである。入力データを見ると、どの展示にも滞在していないユーザーが何人いることがわかった。これは不自然である。さらに、サーバにアップロードされた csv ファイルを確認すると多くのデータの滞在時間が 1 分未満であり、これらの情報をすべて使用していないことがわかった。滞在時間が 1 分未満のデータの量は予想以上に多く、これらのデータも積み重なれば無視できない影響を与える。そこで、これらのデータもすべて含めて入力データを作って推薦をしてみた。その結果で 1 位になった回数をグラフにすると図 7 のようになった。図 6 を見ると、図 5 に比べて推薦結果の偏りが緩和されていることがわかる。図 5 と図 6 を比較すると、ノイズとして排除していたデータを加えたとき、展示 2 や展示 12 の推薦回数は大きく下がっているにも関わらず、展示 4 については変わっていないことがわかる。これは展示 4 には 2 つのビーコンを設置したことが原因だと考えられる。展示 4 に滞在したユーザーの行動履歴を見ると、展示 4 に数秒滞在して離れて、またすぐに滞在して離れるという行動を繰り返していることが多かった。これは展示 4 に設置された 2 つのビーコンの片方に近づいて離れて、すぐにもう一方に近づいて離れる、ということを繰り返していたためだと考えられる。そのため、滞在時間が 1 分以下のデータが多く、それらのデータを加えたことによって展示 4 の滞在時間は長くなる。展示 2 や展示 12 には

そのようなことが起こらない。よって、展示 4 だけがノイズとして排除していたデータを加えても、推薦回数が下がらなかったのだと考えられる。

4.6 推薦の影響力

推薦の与える影響力について考える。1 位に推薦された展示を訪れているユーザー数を計算した。ここでの「展示を訪れた」の定義は「合計で 1 分以上（連続していなくても良い）その展示の近くにいたこと」である。計算の結果、推薦された展示のすべてに訪れた人は 24 人、推薦された展示の一部に訪れた人は 12 人、推薦された展示を訪れなかった人は 20 人、推薦を利用しなかった人は 23 人であった。「推薦された展示の一部に訪れた」とは複数の展示が 1 位としてユーザーに推薦され、その一部の展示を訪れたことを意味する。展示の一部に訪れた人を 0.5 人分としてカウントすると推薦を利用した人の中で推薦された展示を訪れた人の割合は $(24 + 12 \times 0.5) \div (24 + 12 + 20) \cong 53.6\%$ となる。この数値だけを見ると推薦はうまくっているように見えるが、ユーザーの訪れた展示数の平均を計算すると 1 人のユーザーは平均して 6.83 個の展示を訪れていることがわかった。展示の数は 14 個なので約半分の展示を訪れていることになる。これは推薦結果を見ないでランダムに展示を回ったとしても、約 50% の確率で推薦された場所を訪れることを意味する。この結果から推薦の与える影響はあまり大きくなかったと言える。そうなった理由の一つとしては、展示の数が少ないことが考えられる。展示の数が少ないのでわざわざ推薦に頼らなくても、どんな展示があるのか、ざっと見る事ができる。よって、おススメされた展示でも実際に見て興味が持てなければ展示を訪れない可能性がある。これは推薦を利用しなかった人が多い理由でもありそうである。また、推薦を利用しなかった人が多い理由としては、利用しなかったのではなく、利用できなかった人がいることもある。実験中、「おススメを更新」ボタンを押しても推薦ができないというユーザーが何名かいた。これは、おそらくネットワークとの接続が切れてサーバとの接続が切れたことで、推薦結果を見ることができなくなってしまったことが原因だと考えられる。

先ほどと同様に、推薦された展示を訪れた人の割合を日付ごとに求めてみると 1 日目から 5 日目までの結果はそれぞれ、35.4%、26.7%、50.0%、20.0%、55.9% となった。データ量が増えればより適切な推薦ができるはずなので、実験の後半になるにつれて推薦の与える影響力が大きくなるのではないかと考えたが、そのようなことはなかった。

推薦されたときの現在地と、推薦された場所に訪れ

たかどうかの関係について調べてみたが、あまり関係はなかった。現在地から遠い場所が推薦されても展示を見に行った人がいれば、現在地の近くが推薦されていても展示を見に行かなかった人もいた。この原因としては、実験会場がそこまで広くないことが考えられる。現在地から近い展示も遠い展示も、移動時間はほとんど変わらないので、行動に影響が出にくかったのだと考えられる。実験をもっと大規模に行うと意味のある結果が得られることが期待できる。

5. 関連研究

本稿の実験では Matrix Factorization の基本的な部分のみを使ったが、Matrix Factorization には多くの精度を上げるための工夫が提案されている。評価した時間を考慮したり、ユーザーやアイテムの属性情報を利用したり[1, 2]評価される頻度を考慮したりする方法[2]などが提案されている。また Matrix Factorization 以外の推薦アルゴリズムとしては、pLSA[3]や Slope One[4]、制限付きボルツマンマシンを使った推薦[7]などがある。pLSA はどのアイテムを好むかを確率で表す方法である。Slope One では評価値の平均との偏差から評価値を予測する。

協調フィルタリングには、本稿で利用したユーザーベース協調フィルタリングの他にもアイテムベース協調フィルタリングがある。これはアイテム間の関係を見て推薦を行う方法で、Amazon などで行われている[1,5,12]。[5]では推薦に関する基本的なことが述べられていることに加え、入力するデータやプライバシーの問題、サクラ攻撃などに関する考察が述べられている。

位置推定の方法としては WiFi を使った位置推定[6]がある。[6]では WiFi を用いて三角測量のようなことを行っていた。同じ方法が WiFi の代わりに複数のビーコンを使っても実装できそうである。

6. おわりに

本稿では推薦を IoT 技術と組み合わせて利用するという実証実験に取り組んだ。実験で使ったシステムでは iBeacon を用いて位置情報を取得し、滞在時間の長さによってユーザーの嗜好を予測した。推薦には Matrix Factorization と呼ばれる技術を用いた。実験によって、位置情報を正確に把握するのが困難であること、ただ場所を示すだけではユーザーが推薦をそこまで参考にしないので、推薦結果の提示方法を工夫する必要があること、入力データが少ないと推薦結果に偏りが生じてしまうため、入力データの量をもっと増やす必要があることなどがわかった。

今後の研究の方針としては次のようなものが挙げ

られる。まず、より規模を大きくして実験を行うこと。
展示の数が見て回れないくらい多い状態で実験ができると、推薦を参考にする人が増えることが期待できる。
また、今回の実験では属性情報によって照明の色やアロマの香りを変えたが、その色や香りはあらかじめ決め打っていた。これらについても照明やアロマディフューザーの近くの滞在時間をもとに、ある属性の人がどういった色や香りを好むかを解析し、動的に色や香りを決められるようにしていきたい。

参 考 文 献

- [1] Yehuda Koren, Robert Bell and Chris Volinsky “Matrix Factorization Techniques for Recommender Systems”, Proceeding of IEEE Computer Society, 2009
- [2] Yehuda Koren “The BellKor Solution to the Netflix Grand Prize”, Report from the Netflix Prize Winners, 2009,
http://www.netflixprize.com/assets/GrandPrize2009/BPC_BigChaos.pdf
- [3] Abhinandan Das, Mayur Datar, Shyam Rajaram and Ashutosh Garg “Google News Personalization: Scalable Online Collaborative Filtering”, Proceeding of WWW, 2007
- [4] Daniel Lemire and Anna Maclachlan “Slope One Predictors for Online Rating-Based Collaborative Filtering”, Proceeding of SDM, 2005
- [5] 神 嶋 敏 弘 “推薦システムのアルゴリズム(1)”, 人工知能学会誌, vol.22, no.6, 2007
- [6] 暦本 純一, 塩野 崎敦, 末吉 隆彦, 味八 木崇 “PlaceEngine: 実世界集合知に基づく WiFi 位置情報基盤”, インターネットコンファレンス論文集, 2006
- [7] Ruslan Salakhutdinov, Andriy Mnih and Geoffrey Hinton “Restricted Boltzmann Machines for Collaborative Filtering”, Proceeding of ICML, 2007
- [8] <http://kc-i.jp/>
- [9] <https://ja.wikipedia.org/wiki/IBeacon>
- [10] <https://code.google.com/p/mf-predictor/>
- [11] <http://kontakt.io/>
- [12] <http://ameblo.jp/principia-ca/entry-10980281840.html>
- [13] <https://www.change-makers.jp/post/10447>