

# 目標消費カロリー制約を考慮したジョギングルート推薦アルゴリズムの評価

2016SC035 加藤廣紀 2016SC100 山本航生

指導教員：河野浩之

## 1 はじめに

現在、日本人の死因のうち最も大きな要因を占めるのは生活習慣病<sup>\*1</sup>で、生活習慣病の予防方法の1つとして消費カロリーと摂取カロリーの管理を行う方法がある。摂取カロリーの管理を行うためには買い物を行う際に、表記されているカロリーを参考にすれば摂取カロリーの管理を行うのは比較的簡単である。対症的に、消費カロリーの計算方法はあまり知られていない。本研究では、Mets<sup>\*1</sup>を用いてジョギングルート推薦を行う。ランナーが入力する目標消費カロリー制約を満たす最短のジョギングルートを紹介する。研究環境を名古屋市内に設定し、経路スポットは名古屋市内に1000個以上あるという点、地理的座標を取得しやすい点から、寺院を選定した。

## 2 ルート推薦の関連研究

この章では本研究の関連研究について紹介する。2.1節では、TwitterとGoogle Places APIを用いたルート推薦システムについて、2.2節では目標値を考慮したルート推薦システムについて紹介する。

### 2.1 マイクロブログを利用した観光ルート推薦における移動効率の改善 [4]

中川らはTwitterとGoogle Places APIを用いた観光ルート推薦を行っている。投稿文からユーザが実際に観光スポットに訪れていると推測されているものを収集し、Google Places APIと組み合わせて観光スポットの抽出をしている。また、投稿された時間から観光スポットの開園時間を予測し、ただ観光スポットに訪れるだけではなく、移動効率をユーザの満足度を指数として考慮したルート推薦システムを構築している。

### 2.2 消費カロリーを暗黙的に管理する散歩ナビゲーションシステムの提案 [5]

奥村らは現在の消費カロリーをリアルタイムに計算し、目標カロリーに対する残余カロリーを算出する。提案手法は、次の目的地を反復的に推薦する往路モードと、自宅までの帰宅ルートを推薦する復路モードに大別される。往路モードでは、各交差点をノードとした道路ネットワークを構築し、残余カロリー以内で到達可能な目的地を繰り返し推薦する。復路モードでは、残余カロリーをなるべく満たす帰宅ルートを推薦することで、目標カロリーを消費できる散歩ルートの推薦を可能としている。

<sup>\*1</sup> e-ヘルスネット(厚生労働省), <https://www.e-healthnet.mhlw.go.jp/>, 参照 Jan 14, 2020.

## 3 目標消費カロリー制約を考慮したジョギングルート推薦の提案

この章では、目標消費カロリー制約を考慮したジョギングルート推薦の提案について示す。3.1節では、Google Maps APIについて、3.2節では、カロリー計算の方法について、3.3節では、ルート推薦の手法について説明する。

### 3.1 Google Maps APIについて

Google Maps APIについて説明する。Google Maps APIとは、Google Mapsを無料でWebサイトやアプリに掲載することができる仕組みである。Google Maps APIは18種類あり、用途に合わせて使用することができる。我々が使用するGoogle Maps APIを以下に紹介する。

1. Google Maps Distance Matrix API [3]: 複数の始点・終点の組み合わせルートの距離、時間を取得
2. Google Maps Directions API [2]: 最適化された経路の検索を自動で行うことが可能

### 3.2 Mets 計算の方法

本研究では、ジョギングによって消費するカロリーの計算を行う。そのカロリーの計算に厚生労働省が定めたMetsを用いる。Metsとは<sup>\*2</sup>、安静時の運動強度を1としたときに、その運動で何倍のエネルギーを消費できたかで運動強度を示した単位である。1回の運動で消費されるエネルギー量は、体格・運動強度・運動時間で決まる。本研究で用いられるジョギングの運動強度はおおよそ7.0Metsである。消費したい目標カロリーはMetsや時間や体重を用いることで算出する。算出方法を式(1)に表す。

$$\text{消費カロリー} = 1.05 \times \text{Mets} \times \text{時間} \times \text{体重} \quad (1)$$

### 3.3 ルート推薦の手法

Pythonのプログラムを用いて、Google Maps APIと納骨堂info<sup>\*3</sup>から取得した名古屋市内に存在する寺院のデータを処理する。カロリー消費を目的としたジョギングを行う人(以下ランナー)は情報を入力する。入力された情報から、距離や時間などルート探索に必要な計算を行う。その後、経由する寺院が得られ、ジョギングルートを作成する。

<sup>\*2</sup> 健康長寿ネット運動強度とエネルギー消費量(公益財団法人長寿科学振興財団), <https://www.tyojyu.or.jp/net/kenkou-tyojyu/undou-kiso/undou-energy.html>, 参照 Jan 14, 2020.

<sup>\*3</sup> お寺一覧 | 納骨堂info(インフォ), <http://nokotsudo.info/list/>, 参照 Jan 14, 2020.

はじめにランナーが消費したい目標カロリー、速度 (メートル ( m ) / 分), 体重 ( kg ) を入力する. その情報から厚生労働省が公表している運動強度の単位 (Mets) から, ジョギングでの運動強度の 7.0Mets を用いて時間の算出を行う. その時間はカロリー計算に用いられる式 (1) から得る. 得られた速度と時間から距離を算出し, おおよそのジョギングする距離とする. また, 名古屋市内に存在する寺院を Excel Geo を用いて地理的座標 (緯度, 経度) に変換する. 変換された寺院の地理的座標などの情報を, CSV ファイルとして保存する. それらの情報を基に Google Maps Distance Matrix API を用いてスタート地点から経由地点の距離の計算, そして決定された経由地点から経由地点, またはゴール地点の距離を計算する. それぞれの段階で経由地点を決定し, ゴール地点までのルートを探査する.

#### 4 目標消費カロリー制約を考慮したジョギングルート推薦のアルゴリズム

この章では, 提案するジョギングルートの流れについて説明する. 4.1 節ではジョギングルート推薦のアルゴリズムについて, 4.2 節では Excel Geo を使用した寺院の地理的座標変換, 取得について, 4.3 節ではカロリー計算式に基づいたジョギング距離, 時間の算出について, 4.4 節ではジョギング距離による経由スポットの探索範囲の決定について, 4.5 節では Google Maps Distance Matrix API を用いた経由スポットの選別について, 4.6 節では経由スポット, ジョギングルートの決定について説明する.

##### 4.1 ジョギングルート推薦のアルゴリズム

第 4 章では図 1 のアルゴリズムについて説明する. ランナーはジョギングルート推薦にあたり, 目標消費カロリー, ジョギングの速さ, 体重, スタートとゴール地点を入力する. このジョギングルート推薦では, スタート地点とゴール地点を同じとしてルート探索を行うため, スタートとゴール地点は同じ場所を入力する. 図 1 の 1 については 4.2 節で, 2 については 4.3 節で, 3 については 4.4 節で, 4 については 4.5 節で説明する.

##### 4.2 Excel Geo を使用した寺院の地理的座標の取得

web サイト納骨堂 info から愛知県名古屋市に存在する寺院の住所を抽出する. Excel ファイルに抽出した寺院の名前と住所を入力し, Excel Geo でファイルの解析を行い, 地理的座標を取得する.

##### 4.3 カロリー計算式に基づいた距離, 時間の算出

ランナーが目標消費カロリー, ジョギングの速さ, 体重を入力する. カロリー消費の計算式 (1) を用いて消費カロリーが目標に達する目安のジョギング時間を算出する. 時間の算出はカロリー計算式を変形し, 得られた時間は分単位に変換する. 時間は分単位に変換する際, 秒単位部分は全て切り上げる. これは確実にカロリーを消費するためで

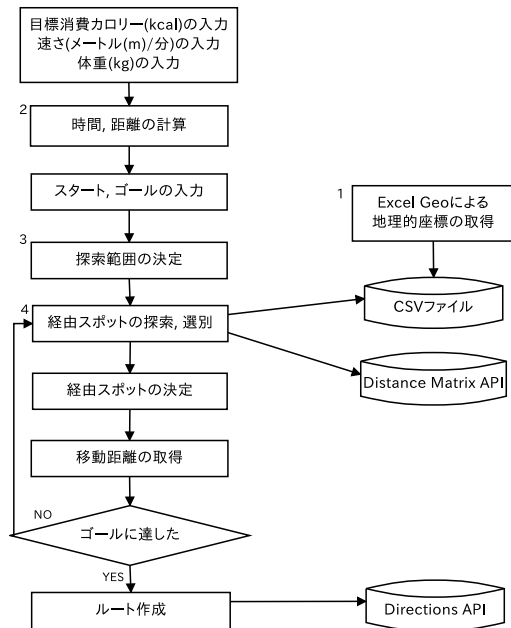


図 1 ジョギングルート推薦のアルゴリズム

ある. 入力された速さと分単位の時間から, ジョギングの距離を算出する. 本研究ではスタート・ゴール地点や経由スポットの位置の関係から短い時間の場合, 目標カロリーと消費カロリーに大きな差が生じる可能性があるため, 20分以上からジョギングルート推薦を行う.

##### 4.4 ジョギング距離による経由スポットの探索範囲の決定

Google Maps Distance Matrix API を用いて経由スポットの探索を行う. 探索範囲はジョギング時間によって変わる. 以下に時間による場合分けを示す.

- 81 分以上の時: 総距離の 4 分の 1 の距離の半径を探索範囲の基準とする. イメージとしては 20 分以上を 4 回となる. しかし 2 目から 3 目は 4 分の 1 以上かつ, ゴールまでが残りの距離以上を範囲とする.
- 61 分以上 80 分以下の場合: 速さに時間を 80 分とみなした乗算で距離を算出する. 1 目と 2 目の経由スポットは 80 分での算出距離の 4 分の 1 の距離の半径を探索範囲の基準とする. また 3 目の経由スポット, そしてゴールまでは, 残りの距離以上を範囲とする.
- 41 分以上 60 分以下の場合: 速さに時間を 60 分とみなした乗算で距離を算出する. 1 目の経由スポットは 60 分での算出距離の 3 分の 1 の距離の半径を探索範囲の基準とする. また 2 目の経由スポット, そしてゴールまでは残りの距離以上を範囲とする.
- 21 分以上 40 分以下の場合: 1 目の経由スポットの探索は, 総距離の 3 分の 1 の距離の半径を探索範囲の基準とする. また 2 目の経由スポット, そしてゴールまでは, 残りの距離以上を範囲とする.

どの場合においても 4 分の 1, 3 分の 1 をした結果の少数部分は切り上げて範囲にする。41 分以上の長時間のジョギングの場合, 20 分後くらいに経由スポットに到着, または通過になるようにする。

#### 4.5 Google Maps Matrix API を用いた経由スポットの選別

Google Maps Distance Matrix API を用いて経由スポットの選別を行う [3]。Google Maps Distance Matrix API のリクエスト用の URL を作成する。スタートとゴールの地点, そして CSV ファイルを読み込んだ寺院の地理的座標を指定し, JSON 形式で受け取った結果を Python で処理をする。図 2 に選別するプログラムの一部 [1] を示す。

```

if x['status']=='OK':
    for key in x['rows']:
        for key2 in key['elements']:
            merter = key2['distance']['value']
            if merter >= distance:
                #1/4 より大きい距離
            if mer1 >= merter:
                #総距離よりも小さい距離
                mer1 = merter
                #条件の中で最小を選ぶ

```

図 2 スポットの選別をするプログラムの一部

図 2 は, 81 分以上のジョギング時間における 1 つ目の経由スポットの選別である。半径を超えた最小の距離のスポットを 1 つ目の経由スポットとし, それぞれの時間においても 4.4 節で示したように選別を行う。

#### 4.6 経由スポット, ジョギングルート決定

61 分以上のジョギングならば 3 つ目の経由スポットからゴール地点まで, 60 分以下のジョギングならば 2 つ目の経由スポットからゴール地点までのルートを決める。どちらの場合においても残り 1 つの経由スポットからゴール地点となる。決定方法としては, 残りの距離を算出し, その残りの距離以上の最小の距離をルートに決定する。

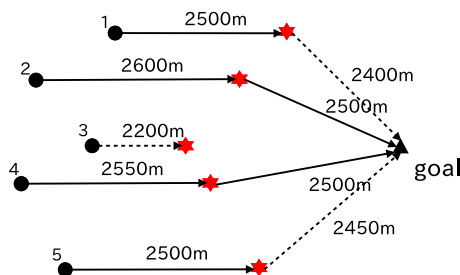


図 3 最終経由スポット決定のイメージ

図 3 の説明をする。これは 81 分以上のジョギング, 半

径が 2500m の場合である。最終経由地に決定されない半径の制約外にあるスポットは 3 のように点線で表している。残った 4 つのルートは, 制約内なので次にゴールまでの距離, そして 1 つ前に求めた経由スポット間の距離を求める。その 2 つの合計距離が残りの距離以上であり, その中で最小距離のルートをジョギングルートとする。図 3 の場合, 残りの距離を 5000m と仮定し, 4 のルートが決定の対象となる。決定したルートは経由スポットが辞書型として格納され, 寺院の名称, 住所, 緯度, 経度そして距離が表示される。その後, Google Maps Directions API を用いて HTML ファイルにスタート地点, ゴール地点, 経由地点を入力し, ブラウザに開いてルートを表示する。

### 5 ジョギングルート推薦による実験

この章では, ジョギングルートを生成し, 実験を行う。5.1 節では, ジョギングルートを生成し, 実験の結果を示す。5.2 節では実験結果を基に, 考察を行う。

#### 5.1 ジョギングルートの生成

Python プログラムを実行し, 図 4 のように目標消費カロリー, 速さ, 体重を入力し, その後スタート・ゴール地点を入力する。図 4 の例では, 表 1 の 1 のように目標消費カロリー, 速さ, 体重, を入力し, スタート・ゴール地点は南山大学付近とする。図 4 にその例を示す。

```

目標消費カロリーを入力(kcal): 300
速さを入力(メートル(m)/分): 125
体重を入力(kg): 55.2
0時間
45分
45分
目標カロリーを消費するためのジョギングは5625m以上です。
スタート, ゴールを入力: 愛知県名古屋市長和区八雲町81-19

1つ目の経由地
{'名前': '善光寺', '住所': '名古屋市天白区山2-3 4 7',
'緯度': '35.132442', '経度': '136.965027', '距離(m)': 2504}

2つ目の経由地
{'名前': '香積院', '住所': '名古屋市昭和区川名山町1 1 5',
'緯度': '35.146519', '経度': '136.956604', '距離(m)': 3193}

合計距離は5697mです。

```

図 4 Python プログラムの実行例

図 4 の実行例に示されたように, 取得した経由スポットの情報, スタート・ゴールの地点を Google Maps Directions API[2] を用いて, HTML ファイルに入力する。HTML ファイルをブラウザに開き, ジョギングルートを表示する。生成されたルートを実際にジョギングし, 結果を図 5 示す。速さや時間, 目標消費カロリーは表 1 の 1 を対象に行う。また消費カロリー, ジョギング時間などを Runkeeper を利用し, 計測を行う。目標カロリーを超えた割合を達成率とする。同じようにルート探索をした後, 計測を行う。合計 11 回の実験結果を表 1 に示す。

実験から, 達成率 91% であった。11 回目の計測では, 消費カロリーが目標カロリーを超えなかった。表 2 のカロリー誤差と距離差, 相対誤差について説明する。カロリー誤差は, 目標カロリーと消費カロリーの差である。次に, 距離



図5 ジョギング結果例

表1 ジョギングルート推薦によるジョギングの結果

	目標カロリー	速さ	消費カロリー	時間
1	300 kcal	125 m/分	324 kcal	43 分 55 秒
2	280 kcal	130 m/分	271 kcal	41 分 16 秒
3	250 kcal	125 m/分	271 kcal	36 分 10 秒
4	200 kcal	130 m/分	247 kcal	26 分 57 秒
:	:	:	:	:
10	150 kcal	133 m/分	169 kcal	22 分 17 秒
11	150 kcal	100 m/分	125 kcal	18 分 00 秒

表2 実験結果による評価

	カロリー誤差	距離差	相対誤差
1	24 kcal	72 m	8.0 %
2	19 kcal	2 m	6.8 %
3	21 kcal	285 m	8.4 %
4	47 kcal	287 m	23.5 %
:	:	:	:
10	19 kcal	92 m	12.7 %
11	25 kcal	41 m	16.7 %

表3 実験結果による平均誤差

平均カロリー誤差	平均距離差	平均相対誤差
30.1 kcal	158.5 m	11.7 %

差はジョギングルートの距離と、時間と速さから算出した距離の差である。最後に相対誤差の算出について以下の式(2)を用いて説明する。

$$\text{相対誤差} = \frac{\text{消費カロリー} - \text{目標カロリー}}{\text{目標カロリー}} \quad (2)$$

式(2)の消費カロリーと目標カロリーの差を目標カロリーで除算する。この数値が小さいほど目標カロリーに対して精度の高いジョギングルート推薦が行われたとする。11通りのルートの平均相対誤差は、11.7%であった。また、

距離差が大きいほど相対誤差が大きくなる傾向となった。

## 5.2 考察

11の計測のように速さが100(m/分)の場合、ジョギングの運動強度に達していないためカロリーが消費できなかったと考えられる。速さが120(m/分)から135(m/分)の時は算出通り、または時間以内にゴールをしたがカロリーは確実に消費できたことから、おおよそ130(m/分)の速さが我々のジョギングルートに向いている。時間を超えている全ての結果について、算出時間から距離を求めているため、実際のルートの距離が大きくなりジョギング時間が超えた可能性も要因と考えられる。本研究では、勾配や風向きを考慮していないため、それらを考慮することでより精度の高いジョギングルートの推薦が行えると考えられる。

## 6 むすび

本研究ではGoogle Maps APIを活用して消費カロリー制約を考慮したジョギングルート推薦アルゴリズムの作成をすることができた。スポットの探索範囲の決定、そして選別によって条件下のスポットを抽出を行い、目標消費カロリーに応じたルートの生成をすることができた。本研究で作成したジョギングルート推薦アルゴリズムは、ランナーの設定した目標消費カロリーを91%達成した。目標消費カロリーと実際に走行して消費したカロリーとの平均誤差は11.7%であった。本研究では推薦されたジョギングルート进行することで、ランナーは消費カロリーの管理を行うことができると考えられる。

## 参考文献

- [1] Directions API(Google Maps Platform) をコールしてルート間の距離と所要時間を Python で取得する, <https://qiita.com/kngsym2018/items/15f19a88ea37c1cd3646>, 参照 Jan 9, 2020.
- [2] Google Maps Directions Service, <https://developers.google.com/maps/documentation/javascript/directions#DraggableDirections>, 参照 Jan 9, 2020.
- [3] Google Maps Distance Matrix API, <https://developers.google.com/maps/documentation/javascript/distancematrix>, 参照 Jan 9, 2020.
- [4] 中川智也, 新妻弘崇, 太田学, “マイクロブログを利用した観光ルート推薦における移動効率の改善,” 第8回データ工学と情報マネジメントに関するフォーラム(DEIM2016), H1-3, pp .1-8, 2016.
- [5] 奥村尚史, 佐々木渉, 高間康史, “消費カロリーを暗黙的に管理する散歩ナビゲーションの提案,” 人工知能学会, インタラクティブ情報アクセスと可視化マイニング研究会(第5回), 2013, [must.c.u-tokyo.ac.jp/sigam/sigam05/sigam0501.pdf](http://must.c.u-tokyo.ac.jp/sigam/sigam05/sigam0501.pdf), 参照 Dec 9, 2019.