

修 士 学 位 論 文

題 目

タクシーの運転支援システム構築に関する研究

指 導 教 員

潮 俊光 教 授

報 告 者

広本 将基

平成 29 年 2 月 1 日

大阪大学基礎工学研究科
システム創成専攻社会システム数理領域
博士前期課程

概要

流しのタクシーが効率よく乗客を乗せるための運転支援システムの開発は、運転手の待遇改善につながる重要な課題である。本論文では、過去の乗車データと現在の流しのタクシーの分布から最適な進行方向を決定する方法を提案する。対象領域をいくつかの分割領域に分割し、各部分領域での需要予測をもとに部分領域ごとの流しのタクシーの変化を混合論理動的システムを使ってモデル化する。そして、モデル予測制御を応用して、各部分領域でのタクシーの最適移動分布を求める。実際に実装したソフトウェアを示す。

目次

概要	i
第 1 章 緒論	1
1.1 研究背景と目的	1
1.2 論文の構成	2
第 2 章 運転支援システムの説明	3
2.1 システム構成	4
2.2 画面上で提示される情報	5
2.2.1 ピン情報	6
2.2.2 ピンに基づいた利己的な走行方向	8
2.2.3 ピンに基づいた推奨される走行方向	8
2.2.4 ニューラルネットワークを用いた需要予測の結果情報	8
2.2.5 交通情報	11
第 3 章 乗降車データに基づいた最適配車問題	12
3.1 タクシー移動モデル	12
3.2 モデル予測制御	17
3.2.1 定式化	17
3.2.2 計算時間	18
3.2.3 実装結果	19
第 4 章 結論	21
謝辞	22
参考文献	23

付録 A	ニューラルネットワークの基礎	25
付録 B	混合論理動的システム	28

第 1 章

緒論

1.1 研究背景と目的

現在のタクシー業界は、高齢化、低賃金、長時間労働という問題を抱えている。流しのタクシーが空車で走行する距離を減らすことで、賃金の向上が達成出来るだけでなく、CO2 排出量の削減にもつながる。現状では、運転手の経験と勘から流し走行をしており、経験の浅い運転手への流し運転の支援は重要な課題である。最近では、すべてのタクシーに GPS が装着されており、乗客を乗せた位置、一定走行時間・距離ごとの位置情報が無線でリアルタイムに会社へ送信されて、管理できるようになった。また、名古屋ではタクシーの自動運転による実証実験が行われている。こうした状況では、ビッグデータを活用して、顧客の発生予測をして、走行方向の支援を行うシステムを考えることは重要である。

一方、ビッグデータを実際に利用するために、プローブ情報の精度に関する比較研究や、自動車の走行データをオンラインでセンシングできるプローブカーを用いた、交通制御や交通状況のモニタリング方法についても研究が行われている¹。タクシーはプローブカーとして重要な役割を果たしているだけでなく、走行履歴から運転手の特性を推定することも可能となってきた²。さらに、携帯電話の位置情報や乗車履歴データから乗客の予測技術も急速に発達してきた³⁻⁵。また、気象条件や交通状況をもとにした機械学習による需要予測に関する研究も行われている⁶。

大量のデータを観測、収集し、解析することが容易になり、需要予測技術も発展してきたことで、サイバーフィジカルシステムの考えに基づいた、タクシーの最適配車の研究が注目されている。Seow 等は、乗客からの呼び出しに対して、乗客の待ち時間が最小になるようにタクシーのグループの間で乗客の割り当てを行う配車方法を提案している⁷。Qu 等は、道路の主要地点をノードとし、それらを繋げたクラフを考え、走行する距離を最小限に抑えながら乗客

を獲得するための走行方法を提案している⁸。Miao 等は、モデル予測制御を用いたパラメータ値が不確実性を持つ場合の最適配車法を提案し、サンフランシスコの市街地を対象に実証実験を行っている⁹⁻¹¹。

本論文では、まず、企業と共同開発したタクシー乗務員の運行をサポートするシステムの説明をする。そして、タクシーの移動モデルを混合論理動的システムでモデル化し、モデル予測制御を応用した合理的な運行方法を提案する。提案モデルでは、個々のタクシーに対して個別の制御入力を行うのではなく、部分領域に分けられた中にあるタクシーに対しては同一の制御入力を行う。また、パラメータに不確実性がないことを仮定している。それらの点が文献⁹⁻¹¹とは異なる点である。

1.2 論文の構成

本論文の構成について述べる。第 2 章では、企業と共同開発したタクシー運転支援システムの説明を行う。第 3 章では、支援システムで開発に取り組んだ、推奨される走行方向の導出を行う。タクシーが営業を行う領域をいくつかの部分領域（セル）に分割し、セル間でのタクシーの移動モデルを混合論理動的システムでモデル化する。そして、そのモデルを用いたモデル予測制御法を提案し、ソフトウェアへの実装結果を示す。最後に、第 4 章では結論と今後の課題について述べる。

第 2 章

運転支援システムの説明

タクシーの営業方法には、待ち営業と流し営業の二種類がある。待ち営業とは、駅前やホテルの入り口などの乗り場で待機して乗客を獲得する営業方法である。この方法の利点は、乗り場で待っていればタクシーは乗客をほぼ確実に獲得できるということ、乗客を探すためにタクシーを空車の状態で走らせる必要がないということである。一方、欠点は、待機所でのタクシーの待ち行列が渋滞の原因の一つになっているということ、待ち営業を行える場所以外での乗客獲得の機会を逃している場合があるということである。流し営業とは、乗客の獲得が期待できる経路を走行して乗客を獲得する営業方法である。この方法の利点は、乗務員の経験と勘があれば待ち営業よりも乗客を獲得できる機会が多いということ、乗客からすれば、わざわざタクシー乗り場に向かわずともタクシーを利用できるということである。一方、欠点は、空車状態での走行時間が長いと燃料費がかかってしまい利益が減ってしまうということ、乗客獲得の確実性がないということである。現在、タクシー事業者は、各乗務員に対して流しでの営業を行うことを推奨している。しかしながら、ほぼ確実に乗客を獲得できるという利点から、待ち営業を行う乗務員が多いのが現状である。

最近では、すべてのタクシーに GPS が搭載されており、乗客を乗せた時刻・位置などが日々データベースに蓄積されている。流し営業の欠点は乗客獲得の確実性がないことだが、乗務員に対してデータベースの情報を活用した有益な情報が提示できる仕組みを構築すれば、流しでの営業を行う乗務員の増加が期待できる。本章では、タクシー営業の生産性の向上を目的として企業との共同開発で作成したソフトウェアのシステム構成と乗務員に提示する情報について説明を行う。

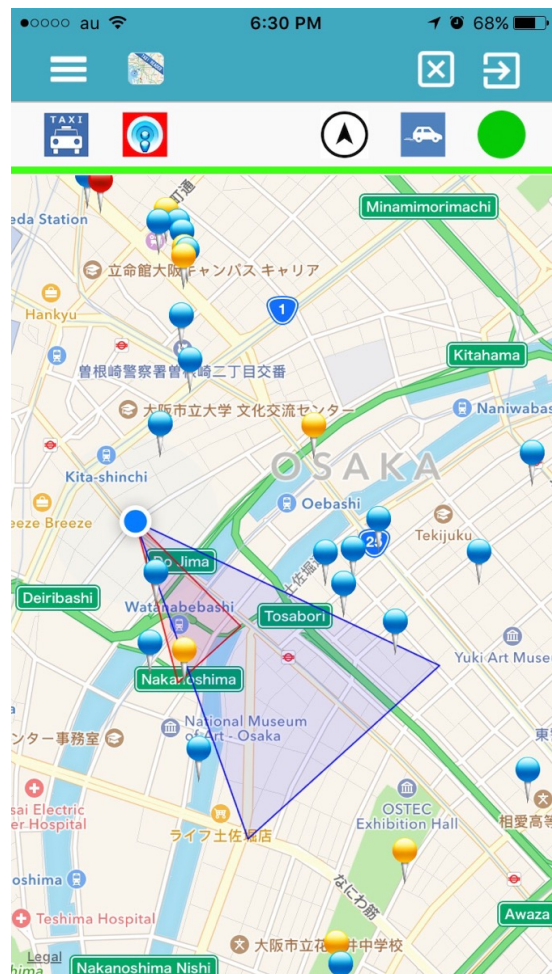


図 2.1: ソフトウェアの実行画面

2.1 システム構成

ソフトウェアはスマートフォン上で実行され、乗務員に提示される。各乗務員が所有しているスマートフォンを利用することで、タクシー事業者はカーナビを購入する経費を抑えられる利点がある。一方、スマートフォンの画面は小さいので、乗務員が使いやすいように搭載する機能や提示する情報を選定する必要がある。

図 2.1 は実際に作成したソフトウェアの実行画面である。乗務員がこのソフトウェアを利用する手順は以下のとおりである。まず、ソフトウェアを起動して画面上で提供される情報を参考にして流し営業を行う。もし、乗客を乗せた場合は画面右上にある緑色のサークルのアイコンをタップする。その時、乗客を乗せた時刻と位置情報はデータサーバーに送信され、サーク

表 2.1: データサーバーに保存される乗降車データ

乗車時刻	乗車緯度	乗車経度	降車時刻	降車緯度	降車経度
2016-11-06 02:18:08	34.6612	135.5029	2016-11-06 02:30:21	34.6494	135.5194
2016-11-06 02:40:40	34.6678	135.5091	2016-11-06 02:47:48	34.6709	135.5390
2016-11-06 03:03:59	34.6693	135.5063	2016-11-06 03:11:49	34.6701	135.5062

ルのアイコンは緑色から赤色に変化する．ソフトウェア上でのタクシーの状態も空車から実車に変化する．緑色のサークルのアイコンはタクシーが空車状態であることを表し，赤色のサークルのアイコンは実車状態であることを表す．乗客を下ろした場合は，もう一度サークルのアイコンをタップしてタクシーの状態を実車から空車に戻す．その時，乗客を降ろした時刻と位置情報はデータサーバーに送信され，サークルのアイコンは赤色から緑色に変化する．

データサーバーには乗降車を行った位置と時刻の情報以外にも，一定時間が経過した場合や，一定距離を走行した場合の位置と時刻の情報も送信される．表 2.1 は実際にデータサーバーに保存される乗降車データである．データ量削減のために空車のときの位置情報や，実車のときの走行経路の情報は保存されない．このソフトウェアは乗務員の運転支援を行うだけでなく，タクシーの走行履歴を取得しデータサーバーに送信する役割もある．

2.2 画面上で提示される情報

ソフトウェア上では，タクシー事業者が定めた時間幅が 1 単位時間に設定されており，1 単位時間ごとにソフトウェアの画面が更新される．画面上で提供される情報は以下の 5 つである．

1 つ目は，過去に乗車実績のあった箇所を表すピン（図 2.1 を参考）である．2 つ目は，ピンに基づいた利己的な走行方向を示す赤色の三角形のオブジェクト（図 2.1 を参考）である．3 つ目は，ピンに基づいた推奨される走行方向を示す青色の三角形のオブジェクト（図 2.1 を参考）である．4 つ目は，古典的なニューラルネットワークにより予測した需要の中で値が大きい箇所を表す情報である．5 つ目は，営業領域内の電車の遅延や事故のような交通情報である．以下では，実際に乗務員に表示できるようにした情報について詳しく説明する．

2.2.1 ピン情報

図 2.1 の画面内にあるピンは過去に乗車実績のあった箇所を示す。実装では、現在の日付から 2,3,4 週間前の日の、現在の時刻から 2 分前から 4 分後までの間に乗車実績のあった箇所をピンで表示するようにした。ピンの色は 3 種類あり、青、黄、赤の順に乗車時間が 10 分未満、10 分以上 20 分未満、20 分以上であった乗降車記録を示している。ピンが密集している箇所は乗客を獲得できる頻度が高いと言える。

図 2.2 と図 2.3 はそれぞれ、表示されたピンの周辺と非周辺での乗車実績を 1 ヶ月スケール（2016 年 12 月 19 日から 2017 年 1 月 18 日）と 1 日スケール（2017 年 1 月 17 日）で示したグラフである。縦軸が乗客獲得数を表し、横軸が時間を表す。赤い折れ線は、ピンから離れたところで乗客を獲得した数を表す。青い折れ線は、ピンの近くで乗客を獲得した数を表す。乗客を獲得した実績に対するピンの周辺と非周辺の識別は以下のように行った。営業領域を一边 200m の正方形のセルで分割した。ピンが 1 本でも立っているセルがあれば、そのセル自身と、そのセルに隣接するセルはピンの周辺と定義した。もし、ピンの周辺で乗客を獲得すれば、ピンの近くで乗客を獲得した数を一つ増やした。そうでなければ、ピンから離れたところで乗客を獲得した数を一つ増やした。

1 日スケールのグラフを見ると、朝の 3 時から 6 時までの間の乗客獲得数は他の時間帯と比べて少ないことがわかる。1 ヶ月スケールのグラフを見ると、正月三が日の乗客獲得数は他の日に比べて少ないことがわかる。さらに、どちらのグラフにも共通して言えることがひとつある。それは、ピンの近くよりもピンから離れたところで乗客を獲得している数が多いということである。

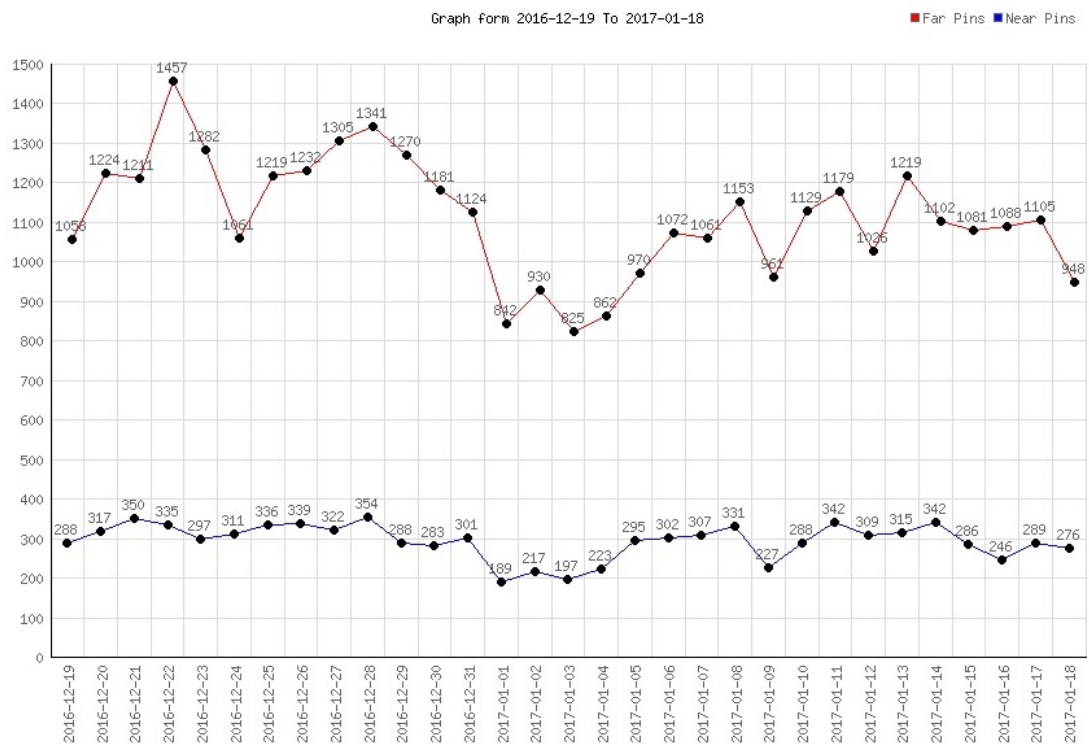


図 2.2: ピンの周辺と非周辺での乗車実績 (1 ヶ月)

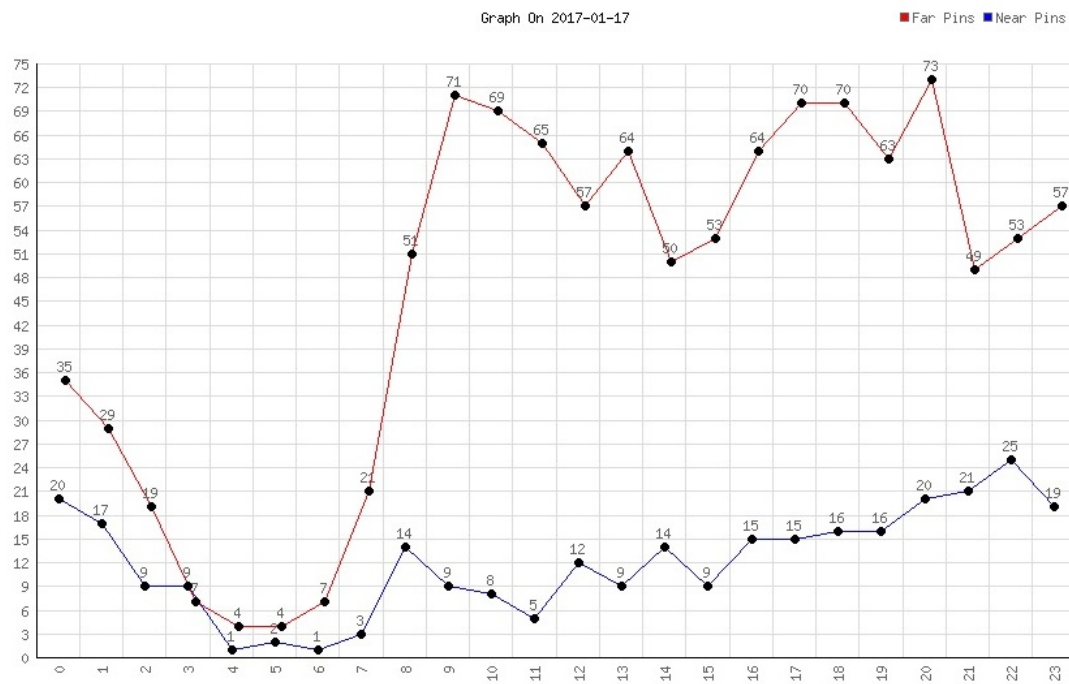


図 2.3: ピンの周辺と非周辺での乗車実績 (1 日)

2.2.2 ピンに基づいた利己的な走行方向

図 2.1 の画面内にある赤い三角形のオブジェクトは周囲で最もピンがある領域を示す方向を表す。ピンの情報が表示してあれば、この情報が無くともピンの多い箇所はマップを見ればわかる。しかし、実際にソフトウェアを利用してみると、マップはある程度拡大表示して利用するため、走行時には周囲のピンの情報を十分に把握することが困難であった。よって、この利己的な走行方向を示す情報は必要である。この走行方向は以下の手順で計算した。まず、営業領域を一辺 200m の正方形のセルで分割し、各セルで表示してあるピンの数を数えた。そして、タクシーがいるセルから 3km 範囲内でピンの数が 3 個以上ある一番近いセルを利己的な走行方向とした。もし、該当するセルが無ければピンの数が 2 個以下である一番近いセルを利己的な走行方向とした。

2.2.3 ピンに基づいた推奨される走行方向

図 2.1 の画面内にある青い三角形のオブジェクトは推奨される走行方向を示すものである。この走行方向を求めるために、まず、タクシーの移動モデルを混合論理動的システムでモデル化した。そして、モデル予測制御を応用して最適な移動方向の分布を求め、その分布の中でもっとも大きい値を持つ方向を推奨される走行方向とした。詳しくは第 3 章で述べる。

2.2.4 ニューラルネットワークを用いた需要予測の結果情報

ピンの情報は、過去の同曜日、同時刻付近の乗車実績を示していた。しかし、気象条件を考慮していない。雨が降ったり、とても暑かったり、寒かったりした場合はタクシーを利用する人が増える傾向がある。そこで、ニューラルネットワークの入力に天候や気温を含ませて、予測した需要の中で値が大きい箇所を表示することを考えた。ニューラルネットワークについては付録を参照すること。

ニューラルネットワークは 3 層とし、以下のような構成にした。入力層は 6 つの情報が入力される。1 つ目は月情報のための 4 個のノードである。1 年を春 (3 月から 5 月)、夏 (6 月から 8 月)、秋 (9 月から 11 月)、冬 (12 月から 2 月) の 4 つに分割して、春であれば春に割り当てられているノードのみに 1 が与えられ、それ以外のノードには 0 が与えられる。2 つ目は時刻情報のためのノードである。24 時間は 1440 分あるので 1440 個のノードを準備する。各時刻で対応するノードのみに 1 が与えられる。3 つ目は祝日であるかどうかのノードである。祝日であれば 1 が与えられ、そうでなければ 0 が与えられる。4 つ目は特別な日であるかどうかの

ノードである．大阪のタクシー業界における特別な日とは 5 日，10 日，20 日，月末のことである．タクシー事業者からの聞き取りで，これらの日は仕事でタクシーを利用する人が多く，それ以外の日と乗客数が異なることがわかったため，ノードに追加した．5 つ目は天候に関するノードである．雨であれば 0 が与えられ，そうでなければ 1 が与えられる．6 つ目は気温に関する 2 つのノードである．その日の最高気温と最低気温を入力ノードに加えた．天候と気温の情報は気象庁のサーバーから XML 形式または JSON 形式のデータで利用可能だが，事前の登録が必要なため，気象庁のサイトの HTML 記述の中からデータ抽出を行った．入力層のノードの総数は 1449 個である．中間層は 1 層，ノード数は 200 個とした．出力層は，タクシーが営業を行う対象領域をいくつかの分割領域に分けたときの，分割領域の個数分だけノードを用意した．出力層では各分割領域の予測した需要数が出力される．

言語は c++，ライブラリは caffe を利用し，2016 年 9 月と 2016 年 10 月の 2 ヶ月間の乗車データを使い学習を行った．セルは一辺 200m の正方形とし，ミニバッチ数は 100，学習方法は確率的勾配降下法とし，最大学習回数を 50000 回とした．ニューラルネットワークに現時刻の 2 分前から 4 分後までの時刻や気象情報をそれぞれ入力して，各ノードの出力結果をそれぞれ足し合わせたものを各セルの需要予測数とした．需要予測数が 1 以上あるセルを需要が多い箇所として，図 2.4 にあるように，赤いサークルで示した．本来需要がない川の上に赤いサークルが表示されている．

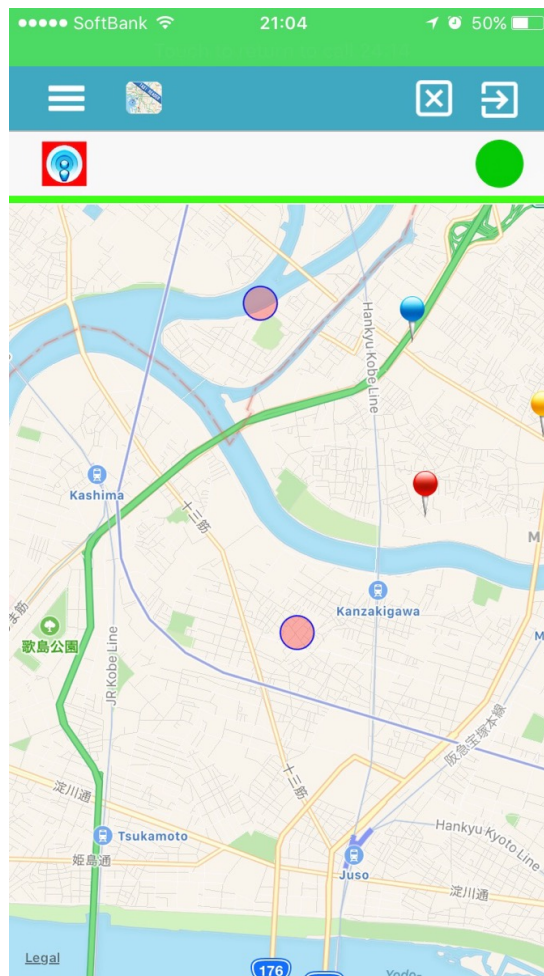


図 2.4: ニューラルネットワークで予測した需要が多い箇所の表示方法

2.2.5 交通情報

電車の大幅な遅延が発生すると、タクシーやバスなどの他の公共交通機関を利用する人が確実に増える。そのため、交通情報も乗務員に提示すべき大事な情報である。インターネット上には、気象情報に限らず交通情報や地震情報のように、時間によって変化するデータが過去のものも含めて公開されている。今回の実装では、「yahoo!路線情報」のサイトの HTML 記述の中からデータ抽出を行い電車の遅延情報を取得し、画面にサイトで表示されている文章を表示することにした。

第 3 章

乗降車データに基づいた最適配車問題

本章では，過去のデータに基づいた推奨される走行方向の計算方法を提案する．タクシーが営業を行う領域をいくつかの部分領域（以下セルと呼ぶ）に分割し，セル間でのタクシーの移動モデルを混合論理動的システムでモデル化する．そして，そのモデルを用いたモデル予測制御法を提案し，ソフトウェアへの実装結果を示す．

3.1 タクシー移動モデル

本節では，タクシー移動のモデル化を行う．対象領域を N 個のセルに分割する．時刻 k のときのセル i ($i = 1, 2, \dots, N$) での空車数を $x_i(k)$ とおき，対象領域内のタクシーの実車数を $r(k)$ とおく．時刻 k でのセル i で空車の実車に変化するタクシー数を $s_i(k)$ とおく．すなわち，時間区間 $[k, k+1)$ の間に，セル i で空車から実車になるタクシー数が $s_i(k)$ である．時刻 k でのセル i で実車が空車に変化するタクシー数を $e_i(k)$ とおく．さらに入力として，時刻 k でのセル i からセル j へ移動する空車数を $u_{i,j}(k)$ とおく．つまり， $x_i(k)$ と $r(k)$ が時刻 k におけるシステムの状態量を表し， $s_i(k)$ と $e_i(k)$ と $u_{i,j}(k)$ が時刻 $k+1$ の状態量を記述するための変動量を表している．このとき，各セルの空車数のダイナミクスは

$$x_i(k+1) = x_i(k) - s_i(k) + e_i(k) + \sum_{j=1}^N \left(u_{j,i}(k) - u_{i,j}(k) \right) \quad (3.1)$$

となる．また，実車数のダイナミクスは

$$r(k+1) = r(k) + \sum_{i=1}^N \left(s_i(k) - e_i(k) \right) \quad (3.2)$$

となる．

ここで，式 (3.1)，(3.2) から

$$\begin{aligned} & r(k+1) + \sum_{i=1}^N x_i(k+1) \\ &= \left(r(k) + \sum_{i=1}^N \left(s_i(k) - e_i(k) \right) \right) + \sum_{i=1}^N \left(x_i(k) - s_i(k) + e_i(k) + \sum_{j=1}^N \left(u_{j,i}(k) - u_{i,j}(k) \right) \right) \\ &= r(k) + \sum_{i=1}^N x_i(k) + \sum_{i=1}^N \sum_{j=1}^N \left(u_{j,i}(k) - u_{i,j}(k) \right) \\ &= r(k) + \sum_{i=1}^N x_i(k) + \sum_{i=1}^N \sum_{j=1}^N \left(u_{i,j}(k) - u_{i,j}(k) \right) \\ &= r(k) + \sum_{i=1}^N x_i(k) \end{aligned}$$

が示せる．すなわち，タクシーの総数は変化しない．タクシーの総数を L とおくと

$$r(k) = L - \sum_{i=1}^N x_i(k) \quad (3.3)$$

の関係が成り立つ．

実車に変化するタクシー数 $s_i(k)$ については以下のように考える．まず，時刻 k での制御入力に従って移動してから実車になりうるとする．このとき，時間区間 $[k, k+1)$ の間で実車から空車になるタクシーもすぐに実車になりうるので，時間区間 $[k, k+1)$ の間にセル i にいる実車になりうるタクシーの台数は $e_i(k) + \sum_{j=1}^N u_{j,i}(k)$ であり，実車になる台数はこの数を超えることはないので，

$$h_i(k) = e_i(k) + \sum_{j=1}^N u_{j,i}(k) - \alpha_i d_i(k) \quad (3.4)$$

とおくと，

$$s_i(k) = \begin{cases} \alpha_i d_i(k) & \text{if } h_i(k) \geq 0 \\ h_i(k) + \alpha_i d_i(k) & \text{otherwise} \end{cases} \quad (3.5)$$

と表される．ただし， α_i はセル i においてタクシーに乗車できる乗客の割合である．つまり，領域内に乗客と空車のタクシーがいる状況でも，乗客を見つけることが出来ず，実車に変化できない場合を考慮したモデルになっている．この定数 α_i は，過去のデータから推定することができる． $d_i(k)$ は時間区間 $[k, k+1)$ の間にセル i で乗車できなかった客数であり，

$$d_i(k+1) = d_i(k) - s_i(k) + p_i(k) \quad (3.6)$$

と表される．ただし， $p_i(k)$ は時間区間 $[k, k+1)$ の間で発生する新たな乗客数で，過去の乗車データから予測される．

空車に変化するタクシー数 $e_i(k)$ は

$$e_i(k) = \beta_i r(k) \quad (3.7)$$

とする．ただし， β_i は実車全体の中でセル i で空車になる割合である．この定数 β_i は，過去のデータから推定することができる．

ここで，入力に関する制約として，各セル i について

$$x_i(k) = \sum_{j=1}^N u_{i,j}(k) \quad (3.8)$$

を与える．このことは，各セルにおいて時刻 k での空車をどこに移動させるかを決定し，それに沿って，空車が移動すると仮定していることになる．空車は制御入力に従って移動してから実車に変化できると仮定する．このように移動する空車数を定めると，式 (3.1)，(3.8) から，各空車数のダイナミクスは

$$x_i(k+1) = e_i(k) - s_i(k) + \sum_{j=1}^N u_{j,i}(k) \quad (3.9)$$

となる．さらに，自動車の移動速度の制約から必ず 0 になる $u_{i,j}(k)$ がある．例えば，時間単位で隣接するセルにしか移動できない場合には，セル i に隣接しないセル ℓ については

$$u_{i,\ell}(k) = 0 \quad \forall k \quad (3.10)$$

とおく．

以上より，タクシー移動モデルは以下ようになる．

$$x_i(k+1) = e_i(k) - s_i(k) + \sum_{j=1}^N u_{j,i}(k) \quad (3.11)$$

$$r(k) = L - \sum_{i=1}^N x_i(k) \quad (3.12)$$

$$e_i(k) = \beta_i r(k) \quad (3.13)$$

$$s_i(k) = \begin{cases} \alpha_i d_i(k) & \text{if } h_i(k) \geq 0 \\ h_i(k) + \alpha_i d_i(k) & \text{otherwise} \end{cases} \quad (3.14)$$

$$d_i(k+1) = d_i(k) - s_i(k) + p_i(k) \quad (3.15)$$

$$h_i(k) = e_i(k) + \sum_{j=1}^N u_{j,i}(k) - \alpha_i d_i(k) \quad (3.16)$$

$$x_i(k) = \sum_{j=1}^N u_{i,j}(k) \quad (3.17)$$

$$u_{i,\ell}(k) = 0 \quad \text{if セル } i \text{ からセル } j \text{ に移動不可能} \quad (3.18)$$

ここで，式 (3.14) が条件付きの式になっている．この式は以下のように変形すれば，システム全体は混合論理動的システムになる^{12,13}．

まず，以下の論理変数 $\delta_i(k) \in \{0, 1\}$ を導入する．

$$\delta_i(k) = \begin{cases} 1 & \text{if } h_i(k) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.19)$$

と定義する．このとき，制約条件式 (3.19) は次の不等式制約条件になる¹²．

$$h_i^{\inf}(k)(1 - \delta_i(k)) \leq h_i(k) \leq h_i^{\sup}(k)\delta_i(k) + (\delta_i(k) - 1)\epsilon_i(k) \quad (3.20)$$

ただし， $h_i^{\inf}(k), h_i^{\sup}(k) \in \mathbb{R}$ は $h_i(k)$ の引数を取りうる任意の値に対して $h_i^{\inf}(k) \leq h_i(k) \leq h_i^{\sup}(k)$ であり， $\epsilon_i(k) \in \mathbb{R}_{++}$ は十分に小さな正の実数である．実際に，式 (3.20) は $\delta_i(k) = 1$ のときは

$$0 \leq h_i(k) \leq h_i^{\sup}(k)$$

となり， $\delta_i(k) = 0$ のときは

$$h_i^{\inf}(k) \leq h_i(k) \leq -\epsilon_i(k) (< 0)$$

となるので, $\epsilon_i(k)$ の値を十分に小さくすれば, 任意の精度で制約条件式を不等式制約式に変換可能であることが確認できる. また, 初期時刻を $k = t$ とおくと, $k > t$ の $h_i(k)$ について式 (3.15) から以下の不等式が成り立つ.

$$\begin{aligned}
h_i(k) &= e_i(k) + \sum_{j=1}^N u_{j,i}(k) - \alpha_i d_i(k) \\
&\geq -\alpha_i d_i(k) \\
&= -\alpha_i \left(d_i(k-1) - s_i(k-1) + p_i(k-1) \right) \\
&\geq -\alpha_i \left(d_i(k-1) + p_i(k-1) \right) \\
&\geq -\alpha_i \left(d_i(k-2) + p_i(k-2) + p_i(k-1) \right) \\
&\geq \dots \\
&\geq -\alpha_i \left(d_i(t) + \sum_{c=t}^{k-1} p_i(c) \right) \\
h_i(k) &\leq e_i(k) + \sum_{j=1}^N u_{j,i}(k) \\
&\leq L
\end{aligned}$$

したがって, $h_i(k)$ の上界と下界を以下のように定める.

$$h_i^{\sup}(k) = L \quad (3.21)$$

$$h_i^{\inf}(k) = -\alpha_i \left(d_i(t) + \sum_{c=t}^{k-1} p_i(c) \right) \quad (3.22)$$

論理変数 $\delta_i(k)$ を用いることで式 (3.14) は以下のように変形できる.

$$\begin{aligned}
s_i(k) &= \delta_i(k) \alpha_i d_i(k) + (1 - \delta_i(k)) (h_i(k) + \alpha_i d_i(k)) \\
&= -\delta_i(k) h_i(k) + h_i(k) + \alpha_i d_i(k)
\end{aligned} \quad (3.23)$$

ここで,

$$z_i(k) = \delta_i(k) h_i(k) \quad (3.24)$$

とおくと, 式 (3.23) は

$$s_i(k) = -z_i(k) + h_i(k) + \alpha_i d_i(k) \quad (3.25)$$

となる．式 (3.24) は次の不等式制約条件になる¹³．

$$h_i^{\inf}(k)\delta_i(k) \leq z_i(k) \leq h_i^{\sup}(k)\delta_i(k) \quad (3.26)$$

$$h_i(k) - h_i^{\sup}(k)(1 - \delta_i(k)) \leq z_i(k) \leq h_i(k) - h_i^{\inf}(k)(1 - \delta_i(k)) \quad (3.27)$$

以上より，初期時刻を $k = t$ とおくと，タクシー移動モデルは以下の混合論理的システムで記述される．

$$x_i(k+1) = e_i(k) - s_i(k) + \sum_{j=1}^N u_{j,i}(k) \quad (3.28)$$

$$r(k) = L - \sum_{i=1}^N x_i(k) \quad (3.29)$$

$$e_i(k) = \beta_i r(k) \quad (3.30)$$

$$h_i^{\inf}(k)(1 - \delta_i(k)) \leq h_i(k) \leq h_i^{\sup}(k)\delta_i(k) + (\delta_i(k) - 1)\epsilon_i(k) \quad (3.31)$$

$$s_i(k) = -z_i(k) + h_i(k) + \alpha_i d_i(k) \quad (3.32)$$

$$h_i^{\inf}(k)\delta_i(k) \leq z_i(k) \leq h_i^{\sup}(k)\delta_i(k) \quad (3.33)$$

$$h_i(k) - h_i^{\sup}(k)(1 - \delta_i(k)) \leq z_i(k) \leq h_i(k) - h_i^{\inf}(k)(1 - \delta_i(k)) \quad (3.34)$$

$$h_i^{\sup}(k) = L \quad (3.35)$$

$$h_i^{\inf}(k) = -\alpha_i \left(d_i(t) + \sum_{c=t}^{k-1} p_i(c) \right) \quad (3.36)$$

$$d_i(k+1) = d_i(k) - s_i(k) + p_i(k) \quad (3.37)$$

$$h_i(k) = e_i(k) + \sum_{j=1}^N u_{j,i}(k) - \alpha_i d_i(k) \quad (3.38)$$

$$x_i(k) = \sum_{j=1}^N u_{i,j}(k) \quad (3.39)$$

$$u_{i,\ell}(k) = 0 \quad \text{if セル } i \text{ からセル } j \text{ に移動不可能} \quad (3.40)$$

3.2 モデル予測制御

3.2.1 定式化

節 3.1 で導出したタクシー移動モデルをもとに，時刻 t において以下の有限区間最適制御問題を考える．ただし， T は正の整数である．モデル予測制御とは，1 単位時間（ステップ）以

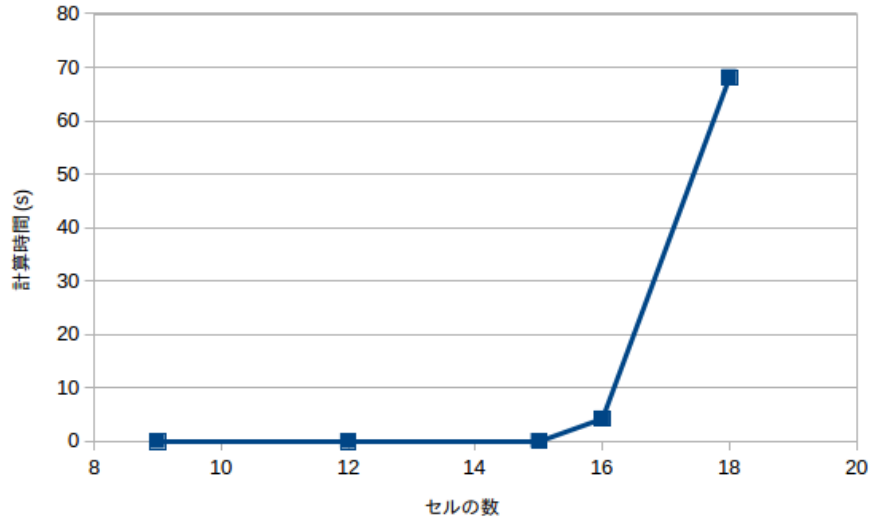


図 3.1: 最適化問題の求解にかかる時間

内で有限区間最適制御問題を解き，最適解の初期入力を次のステップまでの制御入力値として用いるということを各ステップで繰り返し行う制御方法である²⁰．

$$\text{minimize } J_t = \sum_{i=1}^N d_i(t+T) \quad (3.41)$$

制約条件：各 $i = 1, 2, \dots, N$ と $k = t, t+1, \dots, t+T-1$ について

式 (3.28) から式 (3.40) の制約式

$$x_i(t) = \text{given} \quad (3.42)$$

$$d_i(t) = \text{given} \quad (3.43)$$

自明なものも含めると，変数の数は $N^2T + 6NT + 2N + T$ ，制約条件の数は $10NT + 2N$ である．制約条件は線形の式であり，変数に論理変数が含まれるので，求める最適化問題は混合整数計画問題として定式化できる．この問題は NP 困難な問題である．コスト関数 (3.41) は各時刻 t において，時刻 $t+T$ での乗車できない乗客数の総和を最小にすることを意味する．実際にこの問題を解くときには制約式を整理して， $x_i(k)$ ， $u_{i,j}(k)$ ， $d_i(k)$ ， $\delta_i(k)$ の 4 種類のみに変数の数を削減可能である．

3.2.2 計算時間

図 3.1 はセルの数の増加に伴う，最適問題を解くのにかかる時間の変化を表す図である．開発環境は，OS は Ubuntu 16.04，CPU は Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz，メ

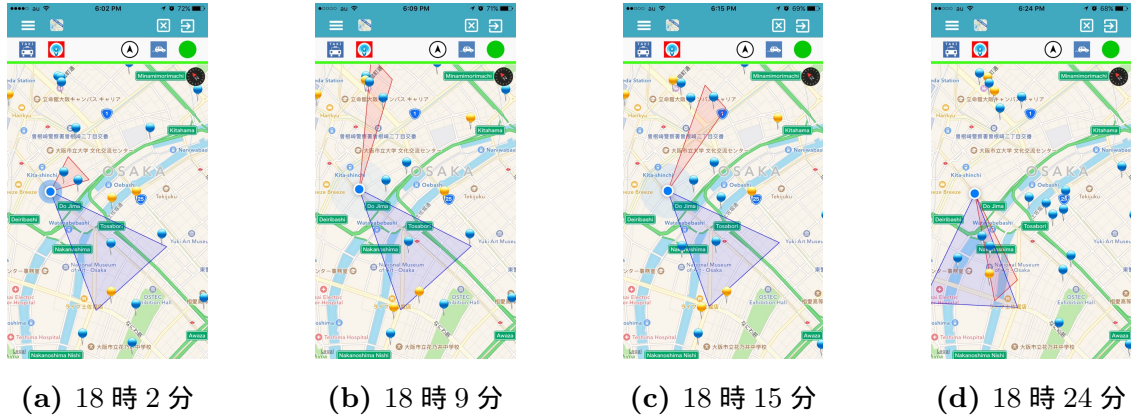


図 3.2: 最適方向の時間的变化 (2016 年 3 月 31 日)

メモリは 4GB である．開発言語は R，ソルバーは lpsolve である．図を見ると，計算時間が指数オーダーで増加していることが確認できる． $N = 20$ のときは，計算に 5 分かかった時点でプログラムの実行を中止した．

3.2.3 実装結果

最後に，ソフトウェア上に実装した結果を示す．タクシーの時速は 60km/h と仮定する．1 単位時間は 2 分とした．タクシーは 1 単位時間で 2km 進むので，一辺の長さ 2km の正方形のセルを使って対象領域を分割した．対象領域は大阪駅から堺市あたりまでとした場合，東西に 6 セル，南北に 4 セルに分割された ($N=24$)．過去の乗降車データからは乗車率 α_i を推定できないため，乗務員の経験則からすべてのセルにおいて乗車率 $\alpha_i = 0.9$ とした．2 単位時間区間 ($T=2$) でモデル予測制御により最適移動分布を計算した．推奨される走行方向は最適移動分布の中でもっとも大きい値を持つ方向とした．図 3.2 に運転手に提示する画面の時間変化を示す．青色の方向が推奨する最適な方向，すなわち，最適化問題を解くことで得られた，移動するタクシー数が最大となる方向である．ピンは，現在の日付から 2,3,4 週間前の日の，現在の時刻から 2 分前から 4 分後までの間に乗車実績のあった箇所を表示している．1 辺の長さ 200m の正方形のセルで分割された対象領域の，3km 範囲内のセルの中で最も乗車数が多いセルを赤色の方向（以下，利己的な方向と呼ぶ）が示している．18 時 2 分から 15 分までは，推奨される方向と利己的な方向が異なっており，18 時 24 分にはほぼ同じ方向を示している．最多乗車データのある領域セルが必ずしも最適な方向であるとは限らないことがわかる．これは，最多乗車データのあるセル周辺に多くの空車タクシーがいる場合には，むしろタクシーを分散させて，対象領域全体として乗客の獲得を図るような最適移動分布を求めているためで

ある .

第 4 章

結論

本論文では、タクシーの走行支援システムを企業と共同開発し、実装を行った結果を示した。過去の乗車実績に基づいてマップ上に立てたピンについては、周囲にピンが立っていないセルでも需要が多く発生していることがわかった。これは、過去の乗車実績をただ表示するだけでなく、需要予測を行いその値に基づいて表示するピンを選ばなければいけないことを示している。推奨される走行方向の計算では、まず、タクシーの移動モデルを混合論理動的システムを用いて表し、有限区間最適制御問題を作成した。そして、モデル予測制御を応用して、混合性数計画問題を解くことで空車の最適移動分布を求めた。推奨される走行方向は最適移動分布の中でもっとも大きい値を持つ方向とした。

実装にあたっては、大阪のタクシー事業者から実データをいただき、実際の空車分布から実証実験を行った。そのとき、利己的な進行方向と、推奨される方向とが必ずしも一致しなかった。これから、推奨される方向を提示して支援を行うことが重要であることがわかる。混合性数計画問題は NP 困難な問題だが、対象領域の設定や分割の仕方を適切に行えば、推奨される方向の提示は実用可能であることが判明した。

また、今回は 1 単位時間に進むことができるセルは隣接するセルのみとした。長さを短くし、隣接するセル以外にも空車が移動できるようにすれば、より精度よく最適移動方向を計算できるが、計算時間が急速に増大する。今後の課題としては、混合整数計画問題の計算の並列化、最適解の近似などによる計算時間の短縮が挙げられる。

現在、車の自動運転に関して多くの研究が行われている。いずれ、タクシーの自動運転が実用化される可能性があるため、タクシーの最適配車問題は今後も様々な研究がなされていくと考える。本研究が、これらの研究の発展に寄与することを期待する。

謝辞

本研究を行うにあたり，懇切丁寧なご指導，ご教授を賜りました大阪大学大学院基礎工学研究科システム創成専攻 潮 俊光 教授に心より感謝の意を表します．

また，さまざまな面において励ましや助言をいただきました潮研究室の皆様に深く感謝いたします．

参考文献

- [1] 山本俊行, K. Liu, 森川高行, “ タクシー配車データのプローブデータとしての活用に関する基礎的分析,” 土木計画学研究・論文集, Vol. 23, No. 4, pp. 863-870, 2006.
- [2] 金月寛彰, 服部宏充, “ プローブカーデータを利用したタクシードライバーの個人特性の分析とモデル化,” 第 29 回人工知能学会全国大会, 演題番号 1N4-4, 2015.
- [3] K. Zhao, S. Tarkoma, S. Liu, and H. Vo, “ Urban HumaMobility Data Mining: An Overview,” in Proc. 2016 IEEE International Conference on Big Data, 2016.
- [4] K. Zhao, D. Khryashchev, J. Freire, C. Silva, and H. Vo, “ Predicting Taxi Demand at High Spatial Resolution: Approaching the Limit of Predictability,” in Proc. 2016 IEEE International Conference on Big Data, 2016.
- [5] L. Matias, J. Gama, M. Ferreira, J. Mendes-Moreira and L. Damas, Predicting Taxi-Passenger Demand using Streaming Data, IEEE Trans. on Intelligent Transportation Systems, Vol. 14, No. 3, pp. 1393-1402, 2013.
- [6] R. Xu, “ Machine Learning for Real-time Demand Forecasting,” PhD thesis, Massachusetts institute of technology, 2015.
- [7] K. T. Seow, N. H. Dang, and D.-H. Lee, “ A Collaborativ Multiagent Taxi-Dispatch System,” IEEE Trans. Automation Science and Engineering, Vol. 7, No. 3, pp. 607-616, 2010.
- [8] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, “ A Cost-Effective Recommender System for Taxi Drivers,” in Proc. 20th International Conference on KDD, pp. 45-54, 2014.
- [9] F. Miao, S. Lin, S. Munir, J. A. Stankovic, H. Huang, D. Zhang, T. He, G. J. Pappas, “ Taxi Dispatch with Real-Time Sensing Data in Metropolitan Areas - A Receding Horizon Control Approach,” in Proc. International Conference on Cyber-Physical Systems, pp. 100-109, 2015.
- [10] F. Miao, S. Han S. Lin, and G. J. Pappas, “ Robust Tax Dispatch under Model

- Uncertainties, ” in Proc. 54th IEEE Conference on Decision and Control, pp. 2816-2821, 2015.
- [11] F. Miao, S. Han, S. Lin, Q. Wang, J. Stankovic, A. Hendawi, D. Zhang T. He, and G. Pappas, “ Data-Driven Robust Taxi Dispatch under Demand Uncertainties, ” arXiv:1603.06263, 2016.
 - [12] A. Bemporad and M. Morari, “ Control of Systems Integrating Logic, Dynamics, and Constraints, ” Automatica, Vol. 35, No. 3, pp. 407-427, 1999.
 - [13] 井村順一，東俊一，増淵泉，ハイブリッドシステムの制御，コロナ社，2014.
 - [14] 中野 良平，ニューラル情報処理の基礎数理，数理工学社，2005.
 - [15] C. Bishop，元田他監訳，パターン認識と機械学習（上，下），丸善出版，2008.
 - [16] 神鷹敏弘，深層学習，近代科学社，2015.
 - [17] F. Rosenblatt, “ The Perceptron: a probabilistic model for information storage and organization in the brain, ” Psychological Review, Vol. 65, No. 6, pp. 386-408, 1958.
 - [18] G. Cybenko, “Approximation by superpositions of a sigmoidal function, ” Mathematics of Control, Signals, and Systems, Vol. 2, No. 4, pp. 303-314, 1989.
 - [19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “ Learning representations by back-propagating errors, ” Nature, Vol. 323, No. 9, pp. 533-536, 1986.
 - [20] 大塚敏之他，実時間最適化による制御の実応用，コロナ社，2015.

付録 A

ニューラルネットワークの基礎

ニューラルネットワークは、動物の脳神経系を模倣した情報処理の数理的モデルである¹⁴⁻¹⁶。本節では、入力に対して確定的に出力が決まる階層型ニューラルネットワークの数理モデルについて説明を行う。

階層型ニューラルネットワークは多層パーセプトロンとよばれ、単純パーセプトロンを層状に繋ぎあわせた構造をしている。単純パーセプトロンは Rosenblatt が提案した情報処理モデルである¹⁷。図 A.1 に例を示す。単純パーセプトロンでは、多次元の入力 $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ に対する出力 $y \in \mathbb{R}$ を以下のように計算する。

$$y = a\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (\text{A.1})$$

$$a(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.2})$$

ただし、 $w_i \in \mathbb{R}$ は第 i 番目の入力要素 x_i に対する結合の重みである。 $\theta \in \mathbb{R}$ はしきい値である。関数 $a(\cdot)$ は単純パーセプトロンの活性化関数や出力関数と呼ばれる。ここで、常に 1 の値をとるダミー変数 x_0 を導入して、 $\mathbf{x} = (x_0, \dots, x_n)$ 、 $\mathbf{w} = (w_0, \dots, w_n)$ とすれば、式 (A.1) は

$$y = a\left(\sum_{i=0}^n w_i x_i\right) = a(\mathbf{w}^T \mathbf{x}) \quad (\text{A.3})$$

となる。ただし、 $w_0 = -\theta$ である。

図 A.2 に 4 層からなる多層パーセプトロンの例を示す。この例では、入力層に入力された信号 \mathbf{x} は途中 2 つの中間層を経て出力層に伝播している。各層の単純パーセプトロン（以下ノードと呼ぶ）では、式 (A.1) に従って信号が変換される。ただし、入力層のノードは入力信号をそのまま出力するので、 $y_i^{(1)} = x_i$ である。多層パーセプトロンの第 n 層の第 k ノードが

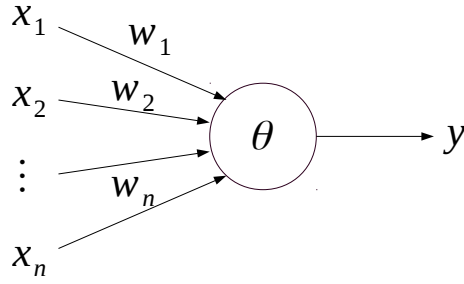


図 A.1: 単純パーセプトロンの例

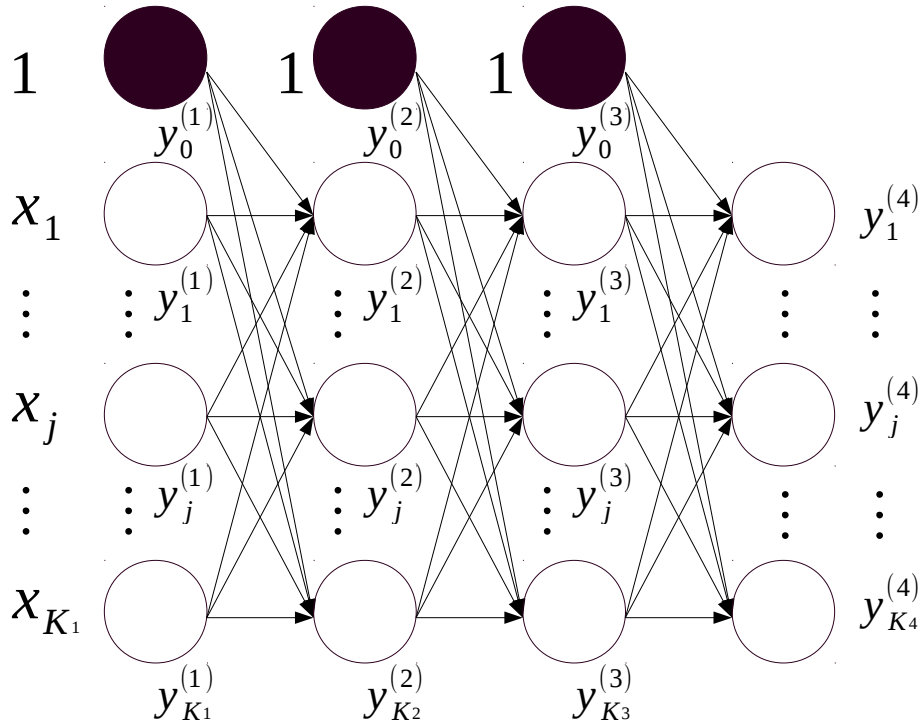


図 A.2: 多層パーセプトロンの例

ら第 $n + 1$ 層の第 j ノードへの結合の重みを $w_{k,j}^{(n+1)}$ とする．ネットワークの中の結合の重みをすべてまとめて $W = \{w_{k,j}^n\}$ と書くと，多層パーセプトロンの出力は $y = g(x, W)$ のように入力ベクトルと結合の重みによって決まる．また，Cybenko は中間層の活性化関数が

$$a(t) = \begin{cases} 1 & \text{as } t \rightarrow \infty \\ 0 & \text{as } t \rightarrow -\infty \end{cases} \quad (\text{A.4})$$

の性質を持つ非線形の連続な単調増加関数であり，出力層の活性化関数が線形関数であるとき，中間層が 1 層の多層パーセプトロンによって任意の連続関数が関数 g によって近似可能であることを示している¹⁸．

最後に、多層パーセプトロンの重みの学習方法のひとつである誤差逆伝播学習について説明する¹⁹。誤差逆伝播学習法では、ノードの活性化関数として微分可能な関数を用いる。ここでは、式 (A.6) のシグモイド関数を用いた場合の説明をする。

$$a(t) = \text{sig}(t) = \frac{1}{1 + e^{-t}} \quad (\text{A.5})$$

多層パーセプトロンの層の数は N とし、学習用のデータとして入力 x と正解の出力 r を与える。学習のための評価基準として、出力層での誤差評価、例えば二乗誤差を用いることにする。

$$E(W) = \frac{1}{2} \sum_{j=1}^{K_N} (r_j - y_j^{(N)})^2 \quad (\text{A.6})$$

活性化関数は微分可能なものを仮定しているので、誤差評価 $E(W)$ への結合の重み $w_{k,j}^n$ の寄与を $E(W)$ の偏微分係数として計算することができる。つまり、勾配降下法により、出力と正解のデータとの誤差が小さくなるように、すべての結合の重みを更新することができる。例えば、確率的勾配降下法による結合の重みの更新は以下ようになる。

$$\begin{aligned} w_{k,j}^{(n+1)} &\leftarrow w_{k,j}^{(n+1)} - \eta \frac{\partial E(W)}{\partial w_{k,j}^{(n+1)}} \\ &= w_{k,j}^{(n+1)} - \eta \delta_j^{(n+1)} y_k^{(n)} \end{aligned} \quad (\text{A.7})$$

ただし、 η は学習率である。式 (A.7) の中にある $\delta_j^{(n+1)}$ は以下の式で求められる。

$$\delta_j^{(N)} = -(r_j - y_j^{(N)}) y_j^{(N)} (1 - y_j^{(N)}) \quad (\text{A.8})$$

$$\delta_j^{(n)} = \left\{ \sum_{k=1}^{K_{n+1}} \delta_j^{(n+1)} w_{k,j}^{(n+1)} \right\} y_j^{(n)} (1 - y_j^{(n)}) \quad (\text{A.9})$$

$\delta_j^{(n+1)}$ の計算過程を見ると、出力層での誤差を前の層に伝播させていく形になっている。このことが誤差逆伝播学習の名前の由来である。

付録 B

混合論理動的システム

混合論理動的システム (以下 MLD システムと呼ぶ) モデルは

$$\begin{cases} x(k+1) = Ax(k) + B_1u(k) + B_2z(k) + B_3\delta(k) \\ Cx(k) + D_1u(k) + D_2z(k) + D_3\delta(k) \leq D_4 \end{cases} \quad (\text{B.1})$$

で与えられる¹³ . ここで $x(k) \in \mathbb{R}^n$ は状態, $u(k) \in \mathbb{R}^m$ は入力, $z(k) \in \mathbb{R}^{l_1}$ と $\delta(k) \in \{0, 1\}^{l_2}$ は補助変数である . また, $A \in \mathbb{R}^{n \times n}$, $B_1 \in \mathbb{R}^{n \times m}$, $B_2 \in \mathbb{R}^{n \times l_1}$, $B_3 \in \mathbb{R}^{n \times l_2}$, $C \in \mathbb{R}^{q \times n}$, $D_1 \in \mathbb{R}^{q \times m}$, $D_2 \in \mathbb{R}^{q \times l_1}$, $D_3 \in \mathbb{R}^{q \times l_2}$, $D_4 \in \mathbb{R}^q$ は定数行列である . 補助変数 δ は, このモデルの離散状態を表している .

バイナリ変数と論理積, 論理和, 否定などの論理演算を含む命題論理は, バイナリ変数と四則演算からなる線形不等式で表現できる . 例えば, 命題 i の真偽を表す変数を論理変数と呼び, $X_i \in \{0, 1\}$ で表す . そして, X_i を命題「 $\delta_i = 1$ である」と対応付け, $X_i = [\delta_i = 1]$ と表現することにする . すると, 各論理演算について以下の補題が成り立つ .

補題 B.1

1. $[\delta_1 = 1] \vee [\delta_2 = 1] (= X_1 \vee X_2)$ と $\delta_1 + \delta_2 \geq 1$ は等価である .
2. $[\delta_1 = 1] \wedge [\delta_2 = 1] (= X_1 \wedge X_2)$ と $\delta_1 = 1, \delta_2 = 1$ は等価である .
3. $[\delta_1 = 1] \rightarrow [\delta_2 = 1] (= X_1 \rightarrow X_2)$ と $\delta_1 - \delta_2 \leq 0$ は等価である .
4. $[\delta_1 = 1] \leftrightarrow [\delta_2 = 1] (= X_1 \leftrightarrow X_2)$ と $\delta_1 - \delta_2 = 0$ は等価である .
5. $[\delta_1 = 1] \oplus [\delta_2 = 1] (= X_1 \oplus X_2)$ と $\delta_1 + \delta_2 = 1$ は等価である .

ただし, $\vee, \wedge, \rightarrow, \leftrightarrow, \oplus$ は, それぞれ, 論理和, 論理積, 否定, 含意, 等価, 排他的論理和である .

最後に本論文で利用した, 連続値変数を含む場合の論理条件の不等式表現について補題を示

す .

補題 B.2 $\delta \in \{0, 1\}$ をインデックス変数 , $x \in \mathbb{R}^n$ を連続値変数とする . このとき , 関数 $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ に対して , 有界集合 $\mathbb{X} \subset \mathbb{R}^n$ 上で次の関係が成り立つ .

1. $[\delta_1 = 1] \leftrightarrow [h(x) \geq 0]$ は , つぎの線形不等式によって任意の精度で近似できる .

$$h_{\inf}(1 - \delta) \leq h(x) \leq h_{\sup}(\delta - 1)\epsilon \quad (\text{B.2})$$

ただし , $h_{\inf}, h_{\sup} \in \mathbb{R}$ はすべての $x \in \mathbb{X}$ に対して $h_{\inf} \leq h(x) \leq h_{\sup}$ を満たすものであり , $\epsilon \in \mathbb{R}_{++}$ は任意に選ばれた十分小さい定数である .

2. $z = \delta g(x)$ はつぎの不等式と等価である .

$$g_{\inf}\delta \leq z \leq g_{\sup}\delta \quad (\text{B.3})$$

$$g(x) - g_{\sup}(1 - \delta) \leq z \leq g(x) - g_{\inf}(1 - \delta) \quad (\text{B.4})$$

ただし , $g_{\inf}, g_{\sup} \in \mathbb{R}^m$ はすべての $x \in \mathbb{X}$ に対して $g_{\inf} \leq g(x) \leq g_{\sup}$ を満たすベクトルである .

研究業績

国際会議

- Masaki Hiromoto and Toshimitsu Ushio, “Learning an Optimal Control Policy for a Markov Decision Process Under Linear Temporal Logic Specifications,” in Proceedings of 2015 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (in 2015 IEEE Symposium Series on Computational Intelligence), Cape Town, South Africa, pp. 548-555, Dec. 2015.
- Toshimitsu Ushio, Masaki Hiromoto, Akiyoshi Okamoto, and Tomoaki Akiyama, “WIP Abstract: A Mixed Logical Dynamical System Model for Taxi Cruising Support System,” In Proceedings of 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems Vienna, Austria April 2016.

国内会議

- 広本 将基, 潮 俊光, 「LTL 制約の下での MDP に対するスーパーバイザの強化学習」, 2016 年電子情報通信学会総合大会, p. 166, 2016.
- 広本 将基, 潮 俊光, 岡本 明義, 秋山 友昭, 「モデル予測制御によるタクシーの最適配車問題の定式化」, 電子情報通信学会技術報告書, MSS2016-76, SS2016-55, pp. 113-116, 2017.