

機械学習を用いた直列型待ち行列システムの性能評価

工藤 友泰[†]二井 克[‡]奥田 隆史[†]愛知県立大学 情報科学部 情報科学科[†] 株式会社ヒミカ[‡]

1 はじめに

待ち行列システムをネットワーク状につなげたものは待ち行列ネットワークと呼ばれる。なかでも待ち行列が直列に接続された形態はコーヒーショップ、スーパー、医療現場などで客が諸々のサービスを受けるサービスシステムで多く見受けられる。例えばあるコーヒーショップでは注文・会計サービスを受けるために客が列を作る。客がコーヒーを注文すればその場で会計を済ませてすぐに提供をし客は退去する。注文がコーヒー以外の場合、客は会計を済ませたのちに商品提供サービスを受けるために、別窓口に並ぶ。

上記のサービスシステムは図1に示すような二つの待ち行列システムを直列に接続したモデルとして表現できる。これを直列型待ち行列システムと呼ぶ[1]。したがって直列に繋がったサービスシステムの設計には直列型待ち行列システムの性能評価が重要である。しかしながら直列型待ち行列システムを数学的に性能評価することは困難である[1][2][3]。そのため直列型待ち行列システムの性能評価には離散シミュレーションが用いられてきたが、多大な計算時間を要する。

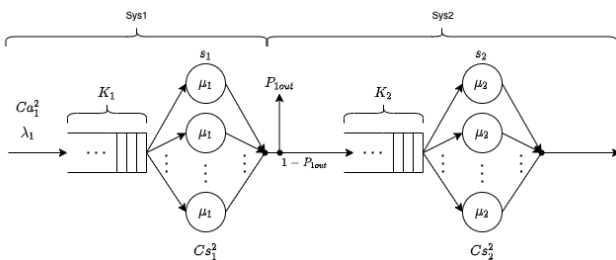


図1: 直列型待ち行列システム

一般待ち行列システムの離散シミュレーションにおける計算時間問題を回避するために、我々の研究グループではこれまで機械学習により一般型待ち行列システム(ケンドールの記号でGI/G/sシステムおよびGI/G/s/sシステム, GI/G/s/Kシステム)の性能評価をおこなうことを提案してきた[4][5]。しかし、直列型待ち行列システムについては性能評価をおこなっていない。

本研究では一般到着間隔分布および一般処理時間分布に従う直列型待ち行列システムの性能評価を機械学習によりおこなうことを提案するとともに、学習効率の高い教師データを検討する。以下、第2節では評価対象の直列型待ち行列システム、第3節では機械学習による性能評価手法、第4節では数値例について述べる。最後に、第5節ではまとめと今後の課題について述べる。

2 評価対象の直列型待ち行列システム

本研究では図1に示す直列型待ち行列システムをGI/G1/s1→G2/s2と表現する。本システムでは、客は到着率λ1、平方変動係数Ca1²の一般到着間隔分布で、最初にSys1へ到着する。到着した客は到着順にサービ

ス率μ1、平方変動係数Cs1²の一般処理時間分布で処理される。Sys1で処理が終わると、客の一部は途中退去確率P1outでシステムから退去し、残りの客は1-P1outの確率でSys2へ到着する。Sys2へ到着した客は到着順にサービス率μ2、平方変動係数Cs2²の一般処理時間分布で処理される。Sys2で処理が終わると、全ての客はシステムから退去する。

なお、Sys1のサーバー性能は同一とし、そのサーバー数はs1とする。同様に、Sys2のサーバー性能は全て同一とし、そのサーバー数はs2とする。本稿ではモデルを単純化するために途中退去確率P1out=0とする。したがってSys1でサービスを受けた客は全てSys2へ到着するものとする。またSys1およびSys2のシステム容量は無限とする。

上述の平方変動係数Ca1², Cs1², Cs2²(以降3つをまとめてC²とする)について説明をする。平方変動係数とは到着間隔および処理時間の分布を決定する値である。本研究では文献[6]に従い、C²=0のときは一定分布、C²=1/k(0<1/k<1)のときはk次のアーラン分布、C²=1のときは指数分布、C²>1のときは2次の超指数分布に従うものとして近似する。

3 機械学習による直列型待ち行列システムの評価方法

直列型待ち行列システムの性能評価指標を、平均システム内時間とする[7]。平均システム内時間とは、客が待ち行列システムに到着してからサービスを受けて待ち行列システムを出ていくまでに平均でかかる時間を表す。本稿では直列型待ち行列システムの平均システム内時間をW1+2と表現する。

平均システム内時間W1+2を求めるために本研究では、機械学習を利用する。本研究で利用する機械学習の手法は教師あり学習のニューラルネットワーク(以降NN)とする。NNの利用理由は、(1)NNは近年のディープラーニングの発展により学習ライブラリ、開発環境が整っていること、(2)NNが情報処理速度が速いからである。NNは教師データを与えて学習をする。学習には少々時間がかかるが学習を終えると入力に対して即座に結果を出力をすることができること[11]、である。代表的な学習ライブラリ、開発環境としてKeras[8]やPyTorch[9]、Chainer[10]などがある。

本研究で利用する評価方法は次の4Stepと誤差検証からなる。

Step1 教師データを得る。

Step2 NNを構築する。

Step3 教師データを利用して構築したNNを学習させる。

Step4 学習したNNに評価対象となるパラメータを入力して直列型待ち行列システムの性能評価をおこなう。

Step1の教師データはGI/G1/s1→G2/s2システムを規定するパラメータとそのパラメータを用いて計算した疑似厳密解のW1+2から成る。疑似厳密解W1+2の計算には離散シミュレーションを用いる。離散シミュレーションによる計算で得た疑似厳密解をW1+2とする。

Step2で構築するNNの構造を図2に示す。NNは1つの入力層、3つの中間層、1つの出力層の計5層から成り、各層のニューロン数は順に8, 32, 16, 8, 1とする。入力層の8つのニューロンはGI/G1/s1→G2/s2システムを規定するパラメータに対応する。中間層はハイ

A Performance Evaluation of Tandem Queueing Systems by Using Machine Learning

[†]Tomoyasu KUDOU, Takashi OKUDA,

[‡]Suguru NII

[†]Department of Information Science and Technology, Faculty of Information Science and Technology, Aichi Prefectural University

[‡]HIMIKA, Inc.

パラメータであるため本研究では試行錯誤の末に暫定的に3層とし、ニューロン数は順に32個、16個、8個とする。最後に出力層の1つのニューロンは疑似厳密解 W_{1+2} に対応する。

Step3では**Step1**で用意した教師データを利用して**Step2**で構築したNNに学習させる。またNNの学習は M 回おこなうものとする。

Step4では学習が完了したNNに評価対象の $GI/G_1/s_1 \rightarrow G_2/s_2$ システムのパラメータを入力する。入力したパラメータによってNNから出力される値を評価対象の $GI/G_1/s_1 \rightarrow G_2/s_2$ システムのNNによる評価結果 $W_{m_{1+2}}$ とする。

誤差検証について説明する。本研究では学習効率の高い教師データを検討するために複数のNNを用意しそれぞれ異なる教師データで学習をさせるため、学習後のNNの性能評価が必要である。学習後NNの性能評価は、評価結果 $W_{m_{1+2}}$ と疑似厳密解 $W_{s_{1+2}}$ の相対誤差を比較し誤差検証をおこなう。

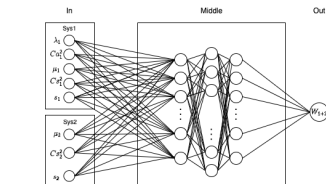


図2: 構築するニューラルネットワークの構造

4 数値例

本研究では表1のパラメータ構成で規定される $GI/G_1/s_1 \rightarrow G_2/s_2$ システムの性能評価をおこなう。また、多様な到着分布および処理分布に対応するために C^2 の取る値は0.1~10.0とする。 μ_1, s_1, μ_2, s_2 の組み合わせは Sys1 および Sys2 におけるサーバの数が単一のときと複数のときの両方を想定し、対応するサービス率は $s_1\mu_1 = s_2\mu_2 = 1$ となる値をとる。したがって到着率 λ_1 はシステム全体が安定となるように0.1~0.9をとる[12]。

表1: $GI/G_1/s_1 \rightarrow G_2/s_2$ システムパラメータ構成

$C_{s1}^2, C_{s2}^2, C_{\mu1}^2$	λ_1	(μ_1, s_1, μ_2, s_2)
0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 6, 8, 10	0.1~0.9	(1, 1, 1, 1) (1, 1, 0.1, 10) (0.1, 10, 1, 1) (0.1, 10, 0.1, 10)

NNに与える教師データの構成は表1のパラメータ構成の中から表2のものを使用する。ここで $X = (0.1, 0.4, 0.8, 2.0)$ としたものを教師データ構成1, $X = (0.2, 0.6, 4.0, 8.0)$ としたものを教師データ構成2, $X = (0.6, 2.0, 6.0, 10.0)$ としたものを教師データ構成3, $X = (0.1, 0.6, 4.0, 10.0)$ としたものを教師データ構成4とする。なお、一度の学習で使用する教師データは4種類のうちの一つであり同時に複数は使用しない。教師データに使用する疑似厳密解 $W_{s_{1+2}}$ は、直列型待ち行列システムを規定するパラメータの組み合わせ一つに対して離散シミュレーションを10万回おこなった平均値とする。離散シミュレーションには Csim20[13] を使用する。また構築したNNは誤差逆伝播法をおこない活性化関数には ReLU 関数を使用する。学習回数は $M = 5000$ 回とし学習ライブラリは Chainer[10] を使用する。

各教師データ構成でNNの学習完了後に表1に示す全てのパラメータをNNに入力し、出力結果 $W_{m_{1+2}}$ を得

表2: 教師データ構成

C_{s1}^2	C_{s2}^2	$C_{\mu1}^2$	λ_1	(μ_1, s_1, μ_2, s_2)
1	X	X	0.1~0.9	(1, 1, 1, 1) (0.1, 10, 0.1, 10) (1, 1, 1, 1) (0.1, 10, 0.1, 10)
X	1	X		(1, 1, 1, 1) (0.1, 10, 0.1, 10)
X	X	1		(1, 1, 1, 1) (0.1, 10, 0.1, 10)
X	X	X		(1, 1, 1, 1) (0.1, 10, 0.1, 10)

る。その後学習効率の高い教師データを検討するためにそれぞれのNNの性能評価をおこなう。まずそれぞれの教師データ構成で学習したNNの出力結果 $W_{m_{1+2}}$ と対応する疑似厳密解 $W_{s_{1+2}}$ との相対誤差を算出する。算出した相対誤差の割合を度数分布表にまとめ横軸に相対誤差の値、縦軸に全体に対する割合としてヒストグラムにしたものを図3~図6に示す。なお、図中の折れ線グラフは累積相対度数である。本研究では小さな相対誤差で評価することができた割合が最も多いもの高い学習効率の教師データとする。

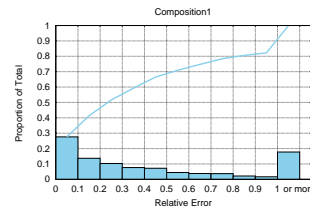


図3: 教師データ構成 1

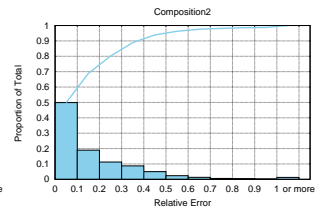


図4: 教師データ構成 2

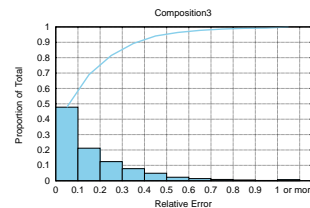


図5: 教師データ構成 3

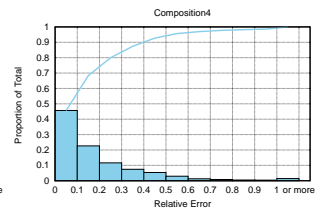


図6: 教師データ構成 4

教師データ構成1を学習したNNの性能評価を示す図3では相対誤差が0~0.1の中で評価できたものが全体の約3割であった。また、相対誤差が非常に大きい1以上の割合が約2割であるため性能が低いといえる。教師データ構成2を学習したNNの性能評価を示す図4では相対誤差が0~0.1の中で評価できたものが全体の約半数であった。相対誤差が1以上のものは少ないことが確認できる。教師データ構成3を学習したNNの性能評価を示す図5は教師データ構成2に似た結果となった。また教師データ構成3は教師データ構成2よりも相対誤差が1以上の割合が小さいことが確認できる。教師データ構成4を学習したNNの性能評価を示す図6では相対誤差が0~0.1の中で評価できたものの割合が教師データ構成2や教師データ構成3と比べるとやや小さいが、全体的には教師データ構成2や教師データ構成3と似た結果となっている。結果として教師データ構成2が小さい誤差で評価できた割合が最も大きいといえる。

5 おわりに

本研究では機械学習を用いて $GI/G_1/s_1 \rightarrow G_2/s_2$ システムの性能評価をおこなった。また、学習効率の高い教師データを検討するために学習後のNNについて性能評価をおこなった。本研究の結果から教師データ構成2が最も学習効率が高い結果となった。したがって教師データ構成2を使用すると $GI/G_1/s_1 \rightarrow G_2/s_2$ システムの性能評価を小さな誤差で性能評価することができることがわかった。今後の課題は誤差をより小さくなるような機械学習の手法やNNの中間層の最適化方法、学習効率が高い教師データなどを検討することが挙げられる。

なお本研究は JSPS 科研費 21K11865 の助成を受けたものである。

参考文献 (下記 URL の最終閲覧日は全て 2021 年 12 月 24 日)

[1] 高橋幸雄他,『混雑と待ち』,朝倉書店, 2001. [2] 宮沢政浩,『待ち行列の数理とその応用』,牧野書店, 2006. [3] 紀一誠,『待ち行列ネットワーク』,朝倉書店, 2003. [4] 奥田隆史他,『ニューラルネットワークによる一般トラフィックモデル GI/Gs の性能評価』, 信学論 B-1, Vol. J76-B1, No. 10, pp. 730733, 1993. [5] 二井克,『機械学習を用いた一般トラフィック型待ち行列システムの性能評価に関する研究』, 令和 2 年度 愛知県立大学大学院情報科学研究科, 博士學位論文, 2021. [6] 秋丸春夫,『情報通信トラフィック 基礎と応用』, オーム社, 1990. [7] 川島幸之助他,『待ち行列理論の基礎と応用』, 共立出版, 2014. [8] Keras, <https://keras.io/> [9] PyTorch, <https://pytorch.org/> [10] Chainer, <https://chainer.org/> [11] 佐々木貴介,『ニューラルネットワークによる語彙解答能力』, 情報処理学会研究報告ゲーム情報学 (GI), Vol.2002, No.69, pp. 45-51, 2002. [12] 小沢利久,『待ち行列ネットワークの安定性』, オペレーションズ・リサーチ, vol. 59, No.11, pp. 649-655, 2014. [13] Mesquite Software, <http://www.mesquite.com>.