クラスタリングを用いたデータ選定による時系列回帰モデル化手法の考察

高橋佑里子† 鈴木 成人†† 山本 拓司†† 福田 裕幸†† 小口 正人†

† お茶の水女子大学 〒 112–8610 東京都文京区大塚 2 丁目 1 − 1
†† 株式会社富士通研究所 〒 211–8588 神奈川県川崎市中原区上小田中 4 丁目 1 − 1
E-mail: †{yuriko-t,oguchi}@ogl.is.ocha.ac.jp, ††{shigeto.suzuki,takuji,fukuda.hiro}@fujitsu.com

あらまし 近年のクラウドサービスにおいて，サーバを仮想化することで使用率を向上させ，サーバ数を削減する取り組みが行われている．この取り組みでは，サーバが自身の CPU 資源を超えた CPU を割り当てられるオーバーコミット状態に陥ることで，仮想サーバの性能が低下する可能性があるため，制御対象のすべての仮想サーバの CPU 使用率を予測し制御を行う必要がある．本研究では，仮想サーバの CPU 使用率の汎用的な深層学習予測モデルの生成に向けて，時系列データの回帰モデル化手法についての考察を行う．再学習に使用するデータを，クラスタリングを用いて選定することで，なるべく少ない再学習による既存モデルの回帰精度向上の施策を検討した．

キーワード 時系列データ，機械学習，深層学習，TFLearn，fine tuning，クラスタリング，回帰

1 はじめに

近年のクラウドサービスにおいて物理サーバ (Physical Machine: PM) の CPU 使用率は低く，そのパフォーマンスを十分に発揮できない状態が続いている．これを改善すべく，事業者では，サーバを仮想化することで使用率を向上させ，PM 数を削減する取り組みが行われている．この取り組みでは，PM が自身の CPU 資源を超えた CPU を割り当てられるオーバコミット状態に陥ることで，仮想サーバ (Virtual Machine: VM) の性能が低下する可能性がある．そのため，図 1 のように制御対象のすべての VM の CPU 使用率を予測し，値が上昇する前に VM を別の PM へマイグレーションする等の制御を行う必要がある [1]．しかし，現状の CPU 使用率の予測モデルは汎用性が低く，特定の VM に対しては高い精度で予測できる一方，その他の VM に対する予測精度は低くなるという課題がある [2]．

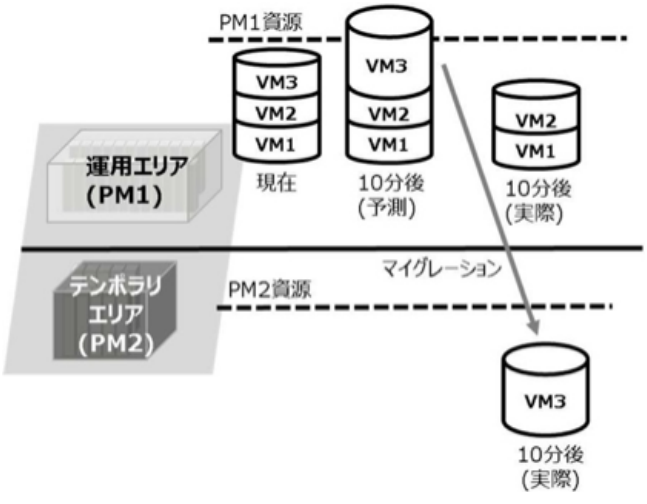

図 1 VM 制御のイメージ

本研究では，VM の CPU 使用率の汎用的な深層学習予測モデルの生成に向けて，時系列データの回帰モデル化手法の考察を行う．深層学習モデルは学習時間が長いため，なるべく少ない再学習での精度向上が求められる．そこで，データを適切に選定することで，少ないデータでの再学習で既存モデルの回帰精度向上の施策を検討した．方法を模索した結果，時系列データを学習に必要な長さごとに抽出し，それらをクラスタリングした結果を元にデータを選定することで，通常よりも少ない再学習でのモデル回帰精度向上が可能であることを確認した．

2 関連研究

クラウドサービスにおける VM の CPU 使用率の予測や制御に関する研究は，以前から多く行われている．[3] では，実際のデータセンターから取得したデータトレースを使用し，VM のワークロードの特徴付けを行い，隠れマルコフモデルをベースとした手法でワークロードパターンの変動を予測する機能を開発している．[4] では，サービスレベル契約 (SLA) 違反と電力コストの削減を目的として，線形回帰手法に基づいて PM の CPU 使用率の短期的な予測を行っている．PM が過負荷になった場合，一部の VM を他の PM に移行することで，エネルギー消費量と SLA 違反率の大幅な削減に成功している．[5] では，VM の CPU 使用率などののバースト性があり，中には明確な周期性を持たない時系列データに対しても，将来の負荷やピーク負荷，タイミングを正確に予測できるニューラルネットワーククベースのフレームワークの開発を行っている．このフレームワークにより，ARIMA モデルや基本的なニューラルネットワークモデルと比較して予測誤差は最大 3 倍，予測タイミングは 2〜9 倍の改善に成功している．[6] では，エネルギー消費量に基づいて，リアルタイムに VM を統合するフレームワークの提案を行っている．提案されたフレームワークを使用

することで，フレームワークを使用していないデータセンターと比較して最大 80%の改善が示され，PM の高使用率と省エネルギーの実現に成功している．

3 関 連 技 術

3.1 Recurrent Neural Network (RNN)

RNN とは，再帰型構造を持つニューラルネットワークであり，過去の計算情報を保持することができるため，時系列データの学習に用いられている．しかし，長期間の情報を保持できないという欠点がある．そこで，セルの内部に入力ゲート，出力ゲート，忘却ゲートを導入することで，RNN を長期依存を扱うことができるように改良したものが Long Short-Term Memory (LSTM) である．そして，LSTM の忘却ゲートと入力ゲートを単一の更新ゲートとして一つに統合することで，計算量が比較的少なくなるよう改良したものが Gated Recurrent Unit (GRU) である．本研究では，計算量を軽微にする目的で GRU を使用し，GRU を 2 層繋げた深層学習ネットワークを構築した．GRU の構造を図 2 に示す．

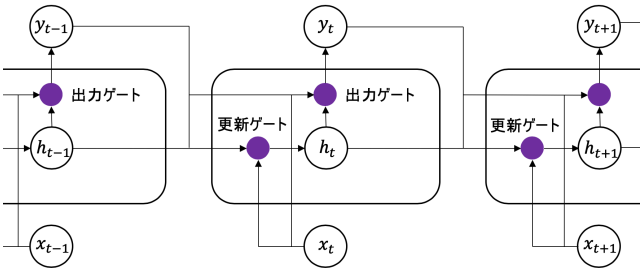

図 2　GRU の構造

3.2 TFLearn

TFLearn [7] とは，Tensorflow [8] の上に構築された透過的かつ互換性のある深層学習ライブラリである．これを使用することで簡潔なコードで深層学習モデルを記述することでき，実験を迅速に行うことが可能になる．本研究では，TFLearn を使用して深層学習を行った．

3.3 fine tuning

fine tuning とは，転移学習の手法の 1 つである．ネットワークの上位層のみを一部を再学習することで，既存のモデルを別のターゲットに応用することが可能となる．

3.4 Root Mean Squared Error (RMSE)

RMSE は，二乗平均平方根誤差と呼ばれる回帰モデルの誤差を評価する指標の 1 つである．この値が小さく 0 に近いほど精度が良いということになる．長さ n の時系列データの正解値を y_t，予測値を $\tilde{y}_t$ とすると，RMSE は以下のような式で計算される．

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \tilde{y}_t)^2}$$

4 実 験 準 備

4.1 実験に使用するデータ

本実験では，データセットとして Bitbrains IT Services Inc. [9] [10] が公開しているデータセンタにおける VM 時系列データセットの ”CPU usage[%]” の項目を使用した．前処理として，時間間隔 300ms，長さ約 43 分間のデータセットを，10 点ごとの平均値を取る平滑化処理と，最小値を 0，最大値を 1 に揃える正規化処理を行った．その後，実験で使用するデータセットを選択するために，2000 個のデータセットの階層型クラスタリングを行った．クラスタ数を 20 としたときのデンドログラムは図 3 のようになった．縦軸はクラスタ間のユーグリッド距離，括弧内の数字はそのクラスタに分類されたデータの個数である．この結果から，学習元データセット (以下 A とする) と A と似ているデータセット (以下 B とする) を赤色部分右側の山からそれぞれ 103 個，A と似ていないターゲットデータセット (以下 C とする) として赤色部分左側の山から 103 個のデータを選択した．

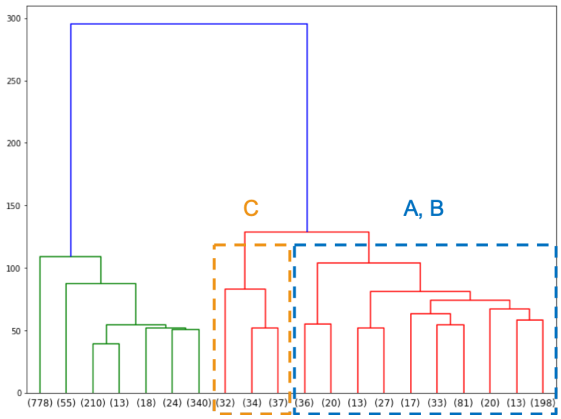

図 3　データセット全体の階層型クラスタリング結果と A,B,C の範囲

4.2 データ粒度最適化

時系列データの学習を行う場合は通常，学習元データ数 (以下，history とする) と正解データ数を設定し，開始点を 1 点ずつずらしながら長さ history の学習元データとそれに対応する直後の長さ正解データ数の正解データのペアを作成し，学習に使用する．そのため，学習には長い波形をそのまま使うのではなく，細かい波形を多数使っているということになる．そこで本研究では，長い時系列データから細かい時系列データを事前に抽出し，それらをクラスタリングした結果を活用することで適切なデータ選定が可能になるのではないかと考えた．

fine tuning 実験を行う前に，データを抽出する最適な長さを見つけるための history 比較実験を行った．代表として選んだ 10 個のデータのそれぞれで，正解データ数を 1 に固定し，前半 7 割を学習データとして history を変えて学習させ，後半 3 割をテストデータとして予測し，それらの結果 10 個の RMSE の平均値を比較した．

まず，history を最小値 6 から 12 ずつ大きく変化させ実験を行った．結果は図 4 のようになった．縦軸が RMSE の平均値，横軸が history である．history が 18〜30 の間で，RMSE の平均値が小さくなっていることが分かる．

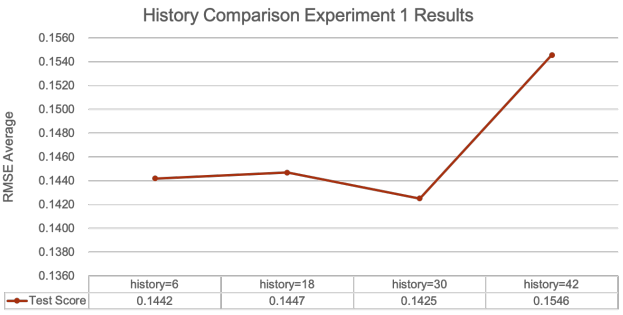

図 4　history 比較実験結果 (12 刻み)

次に，先ほどの実験で RMSE が小さくなった範囲について history を 3 ずつ小さく変化させ実験を行った．結果は図 5 のようになった．history が 24 のとき，RMSE の平均値が最も小さくなることが分かる．

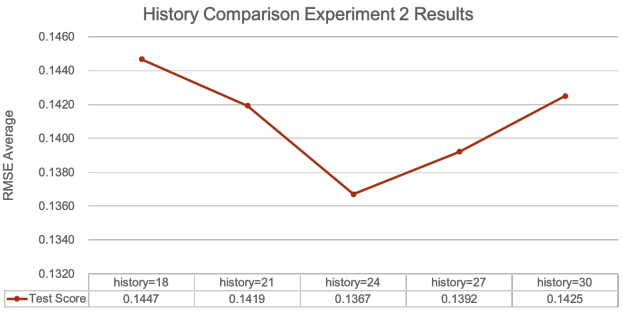

図 5　history 比較実験結果 (3 刻み)

これらの実験結果を受け，本実験の深層学習では，直前の 24 個のデータをもとに 1 個先のデータを予測するネットワークを構築することにした．また，データを抽出する長さは，学習元データ 24 点と正解データ 1 点を繋げた合計 25 点とした．

4.3　データセット細分化

A,B,C に該当する合計 309 個のデータを，開始点を 1 点ずつずらしながら 25 点ごとに抽出し，細分化を行った．細分化後のデータセットの一部を図 6 に示す．同じ値の部分は同じ色で表している．

			0	1	2	3	4	...
cluster_name	data_number	position_number						
A	0	0	0.000056	0.000295	0.549699	0.996331	0.996857	...
		1	0.000295	0.549699	0.996331	0.996857	0.996881	...
		2	0.549699	0.996331	0.996857	0.996881	0.997759	...
		3	0.996331	0.996857	0.996881	0.997759	0.996195	...
		4	0.996857	0.996881	0.997759	0.996195	0.996929	...
...	...	...	...	...	...	...	...	...

図 6　細分化後のデータセットの一部

上記で抽出したデータを，k-means 法クラスタリングにより 10 個のクラスタに分類することで，3 種類のデータセット間の特徴を分析した．結果は図 7 のようになった．縦軸が分類された個数，横軸がクラスタ番号である．A と B は似ているデータセットのため同じような分布になっている一方，A と C は似ていないのデータセットのため異なる分布になっていることが分かる．

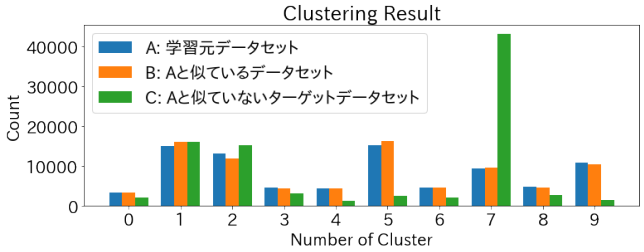

図 7　細分化後 A,B,C の k-means 法クラスタリング結果

4.4　ターゲットデータセットのクラスタリング

fine tuning 実験で使用するデータ選定のために，細分化後の C を k-means 法クラスタリングにより 10，20，30 個のクラスタに分類した．結果はそれぞれ図 8，図 9，図 10 のようになった．縦軸が分類された個数，横軸がクラスタ番号である．

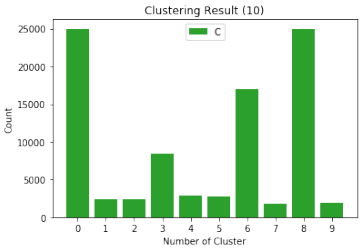

図 8　細分化後 C の k-means 法クラスタリング結果 (クラスタ数 10)

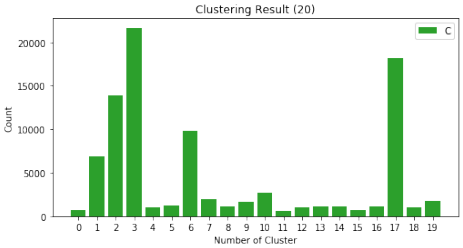

図 9　細分化後 C の k-means 法クラスタリング結果 (クラスタ数 20)

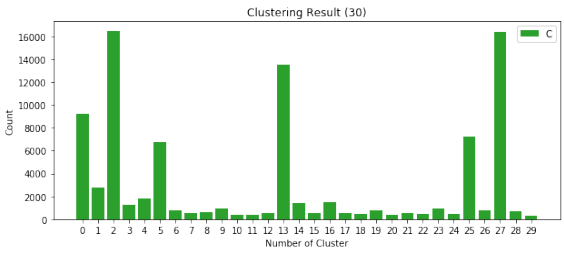

図 10　細分化後 C の k-means 法クラスタリング結果 (クラスタ数 30)

5 実　　験

以下の 2 つの実験を行った．すべての実験において，学習および fine tuning を行う際のエポック数はいずれも 100 に統一しており，モデルの回帰精度の評価指標には，RMSE の平均値を用いている．

5.1 実　験　1

実験 1 では，既存予測モデルは学習元データセットと似ているデータセットに対してどの程度の回帰精度を発揮するかを調査した．A で学習したモデルと B で学習したモデルのそれぞれで B を予測し，それらの回帰精度を比較した．

5.2 実　験　2

実験 2 では，既存予測モデルに対して fine tuning を行うことで別のターゲットに応用する適切な方法を検討した．A で事前学習したモデルに C を使用した fine tuning を以下の 3 種類の方法で行い，その後，A で事前学習した fine tuning を行う前のモデル，方法 1，2，3 で fine tuning 行った後のモデル，および C そのもので学習したモデルのそれぞれで C を予測し，それらの回帰精度を比較した．

5.2.1 方　法　1

方法 1 は，細分化後のクラスタリングで多く分類されたクラスタのデータを使用して fine tuning を行うというもので，図 8 のクラスタ数を 10 とした場合の結果から，No.0(C 全体の 28.0%)，No.3(C 全体の 9.4%)，No.6(C 全体の 19.0%)，No.8(C 全体の 27.9%) に分類されたデータを使用した．

5.2.2 方　法　2

方法 2 は，C の各クラスタのデータを一定の割合で使用して fine tuning を行うというもので，クラスタ数 10，20，30 の 3 通りのクラスタ数について，5%，10%，15%，20%，40%，60%，80%の 7 パターン行った．各クラスタから取得したデータが均一に混ざるよう，繋ぎ合わせた後にデータの順番のシャッフルを行い，試行回数 4 回の平均値を取得した．

5.2.3 方　法　3

方法 3 は，細分化前の状態でデータを選び fine tuning を行うというもので，方法 2 と同様に 5%，10%，15%，20%，40%，60%，80%の 7 パターン行った．この方法では，使用するデータを細分化前の状態で選ぶため，方法 2,3 の同一割合のパターンと細分化後のデータ数は完全には一致しない．

6 結果と考察

6.1 実　験　1

結果は表 1 のようになった．この結果から，既存予測モデルは学習元データセットと似ているデータセットに対して，対象のデータセットで学習したモデルとほぼ同等の回帰精度を発揮することができるということが読み取れる．

表 1　実験 1 結果

学習元	予測対象	RMSE の平均値
A	B	0.1399
B	B	0.1381

6.2 実　験　2

それぞれのグラフの縦軸は RMSE の平均値，横軸は fine tuning のパターンである．また，左端が A で学習した場合の結果，右端が C で学習した場合の結果である．

6.2.1 方　法　1

結果は図 11 のようになった．黄色で示した部分が方法 1 の結果である．この結果から，ターゲットデータセットであっても，偏ったデータを使用して fine tuning を行うことでモデルの回帰精度が低下するということが読み取れる．

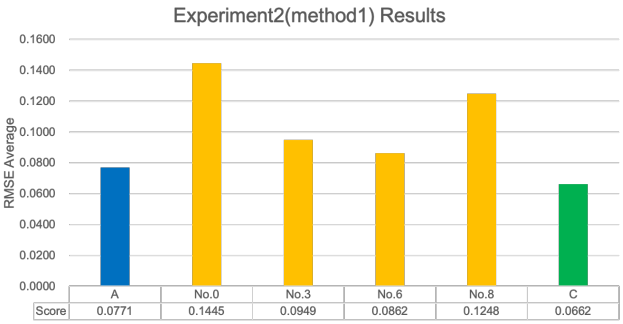

図 11　実験 2(方法 1) 結果

6.2.2 方　法　2,3

結果は図 12 のようになった．両端以外の部分が方法 2,3 の結果であり，方法 2 の括弧内の数字はクラスタリング時のクラスタ数を表している．方法 2 で fine tuinig を行ったモデルの精度は，同じ割合で比較すると方法 3 で fine tuinig を行ったモデルよりも高いことから，ターゲットデータセットを各クラスタから一定の割合で使用して fine tuning を行うことで，細分化を行わない通常のデータ選定時よりも少量の再学習で回帰精度が向上するということが読み取れる．また，方法 2 ではクラスタ数 20，30 と増やすことで，より良いデータ選定ができているということが読み取れる．しかし，方法 2 の結果においても，fine tuning で使用するデータが少ないパターンでは回帰精度は低下しており，40%以上のデータを使用しないと有効とは言えない結果となった．

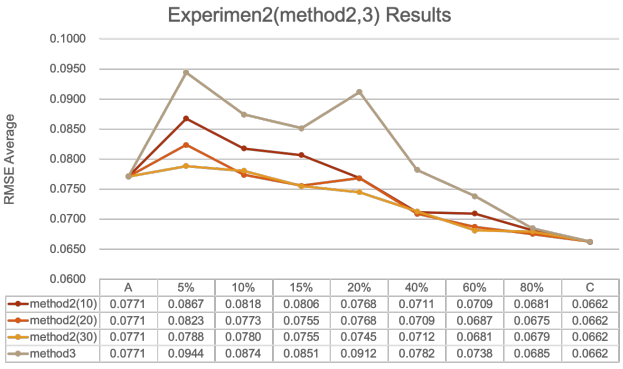

図 12　実験 2(方法 2,3) 結果

7 まとめと今後の予定

VM の CPU 使用率の汎用的な深層学習予測モデルの生成に向けて，なるべく少ないターゲットデータセットの再学習で既存時系列予測モデルの回帰精度を向上させる方法を検討した．実験の結果，似ているターゲットに対しては既存予測モデルをそのまま使用することができること，似ていないターゲットに対しては，データセットを学習に必要な長さごとに抽出し細分化することで，再学習で使用するデータを適切に選定することができ，通常のデータ選定時よりも少量の再学習で回帰精度が向上することを確認した．しかし，現段階ではクラスタリングによるデータ選定が有効であるかどうかについては明言できない結果となった．

今後は，fine tuning を行う範囲を変えたり，新たなデータの選定方法を考えたりしながら，既存予測モデルの汎用化に向けてさらに実験を進めていきたい．回帰の次の段階として，予測の精度向上に向けた取り組みも進めていきたいと考えている．

謝　　辞

本研究の一部はお茶の水女子大学と富士通研究所との共同研究契約に基づくものである．

文　　献

[1] 児玉宏喜, 鈴木成人, 福田裕幸, 吉田英司: マイグレーションを利用したデータセンタの高効率運用手法の提案とオーバコミット時における VM の性能評価, 情報処理学会論文誌 (2018)
[2] 鈴木成人, 児玉宏喜, 遠藤浩史, 福田裕幸: JIT モデリングによるサーバ負荷予測手法の検討と評価, 電子情報通信学会ソサイエティ大会 (2018)
[3] Khan, A., Yan, X., Tao, S., Anerousis, N. (2012, April). Workload characterization and prediction in the cloud: A multiple time series approach. In 2012 IEEE Network Operations and Management Symposium (pp. 1287-1294). IEEE.
[4] Farahnakian, F., Liljeberg, P., Plosila, J. (2013, September). LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In 2013 39th Euromicro Conference on Software Engineering and Advanced Applications (pp. 357-364). IEEE.
[5] Xue, J., Yan, F., Birke, R., Chen, L. Y., Scherer, T., Smirni, E. (2015, November). PRACTISE: Robust prediction of data center time series. In 2015 11th International Conference on Network and Service Management (CNSM) (pp. 126-134). IEEE.
[6] Ismaeel, S., Miri, A. (2019, January). Real-time energy-conserving VM-provisioning framework for cloud-data centers. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0765-0771). IEEE.
[7] TFLearn: http://tflearn.org
[8] TensorFlow: https://www.tensorflow.org
[9] Siqi Shen, Vincent van Beek, Alexandru Iosup: Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters, CCGrid (2015)
[10] http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains