6S-02

帰納論理プログラミングを用いたサーバー障害要因の可読な推定規則の抽出

島田 拓磨 † 秦野 亮 † 西山 裕之 †

† 東京理科大学理工学研究科

1 序論

近年、サーバー環境はますます大規模化・複雑化しており、障害箇所を特定する作業のコストが大きな問題となっている。現在、多くのサーバーが障害検知について ping などによる死活監視などで対策を行っているが、この方法では障害要因の特定について管理者が自力でログ分析を行う必要があり、負荷が大きい。また障害発生後障害の原因や状況についてまとめたレポートを作成することも管理者の大きな負担となっている。

本研究ではサーバーリソースについてのログを帰納論理プログラミング (Inductive Logic Programming、以下 ILP) を用いて解析、ルールを生成を行い、それを用いて障害の検出と障害原因の推定を行うことを目的とする。また、通常管理者が作成する障害レポート作成の補助も期待できる。

2 関連研究

サーバー障害について障害の検知や要因推定について数多く研究されている。Wang ら [1] 時系列データからクラウドコンピューティングにおける Web アプリケーションの自動障害診断手法を提案した。しかし、この研究の主題は高精度な自動障害検知であり、障害の内容の分類や要因の特定までは行っていない。Xu ら [2] はログ文字列を決定木を利用して解析し、障害発生に関係するログ情報を抽出する研究を提案している。しかし、決定木は命題論理であるため、様々なデータの関係性を簡潔に表現することが難しい。

Generating Readable Rules for Server Failure Detection using Inductive Logic Programming
Takuma Shimada†, Ryo Hatano†, and Hiroyuki Nishiyama†
†Graduate School of Sci. and Tech.
Tokyo Univesity of Science

3 手法

本研究では、ILP を用いて、サーバーの障害をリソースログについて解析することによって検出し、障害の原因を特定する。ILP の大きな特徴として、出力 (ルール) が一階述語論理形式で得られ、可読であるという特長がある。これは障害レポートの作成に役立つと考えられる。さらに、時系列を表現する述語を導入することで比較的容易に時系列データを扱うことができる。

一方で ILP はほかの機械学習手法と比較して計算時間が大きい欠点がある。特に述語の数が多い際に顕著である。本手法では述語数を削減するために特徴選択を行うことで、比較的短時間での最適水準を探索の可能を行う。

3.1 学習データの収集

まず、学習を行うために、サーバーリソースについてのログ収集を行う。本研究ではディスクの読み込み書き込みに関するもの、アプリケーションごとのメモリ・CPU 使用率に関するもの、インターフェースの送受信に関するもの、プロセス数、ロードアベレージについてのリソースログについて収集した.

3.2 特徴選択

精度を高めつつ計算時間を短縮するために変数選択を行う。今回扱うログデータは判別に寄与しない特徴や相関が大きな特徴を多く含んでおり、そのまま学習を行うと計算時間が非常に大きくなる。本研究ではランダムフォレスト (以下 RF) によって求めた Gini 不純度を利用し、寄与の小さい特徴を除外した。

3.3 特徴量の離散化

ILP は学習データを背景知識として一階述語論理式に変換しなければならないため、前処理としてデータを離散値に変換する。本研究では各リソースについて基本統計量に基づく離散化と階層型ク

ラスタリングによる離散化を行った。

3.4 ILPによる学習・ルール抽出

特徴選択を行ったデータについてILPを利用した機械学習を行う。学習についてパラメータとなる正事例下限数・負事例上限数を変化させながら数回繰り返し、都度精度を求めながら最適な水準を探索する。精度は3分割交差検証、変化のさせ方は勾配法を利用する。

4 実験及び結果

ログ情報を常に記録している実験用サーバに、障害を発生させ、サンプルとなるデータを作成する。サンプルはサーバーの障害要因として一般的であるDoS攻撃(フラッド型)とメモリリークに関するものの2つの要因について作成・収集を行った。

4.1 DoS攻撃による障害

特徴選択を行った結果、重要度が大きい特徴は、メモリに関するもの、ネットワークインターフェースに関するもの、ロードアベレージに関するものであった。精度について、正事例下限数50・負事例上限数5が最適水準であった。表1より特徴選択を行っても高い精度で判別できることが分かった。

表1: 分類精度

	述語数	precision	recall	f1
特徴選択なし	198	0.784	0.833	0.807
特徴選択あり	40	0.912	0.787	0.845

上で示した水準において、72個のルールが抽出された障害の特徴から、ネットワーク関係の述語を中心とし、そのほかメモリやロードアベレージに関する述語を含むルールが数多く生成された。また、ILPの特長である時系列要素を含むルールを抽出することができた。

代表的なルールを述語形式(1)及び図1に示す。

$$class(A) \leftarrow if\_tx\_bytes\_eth1(A, bigHigh) \wedge before(A,B) \wedge load\_3(B, average) \quad (1)$$

ここで、$if_tx_bytes_eth1$はネットワークアダプタの使用量、$load_3$はロードアベレージの15分間の平均を示す述語である。フラッド攻撃を受けたときの特徴である急激なネットワーク使用量の増加をルールとして抜き出せている。またこのほかに、急激なメモリの使用量の増加を示すようなルールも抽出された。

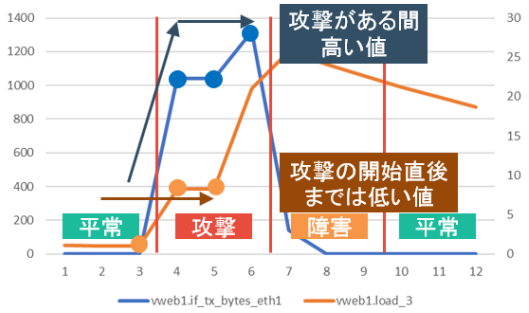

図1: DoS攻撃による障害発生ルールの一例

4.2 メモリリークによる障害

もっとも重要度が大きい特徴はネットワークに関するものであった。次に重要度が大きいものがメモリに関するものであった。メモリに関する特徴(全体91個のうち16個)の寄与率の合計は約23.4%であった。

次にILPによりルールを生成した。ルールは例えば以下のようなログを一目見ただけでは自明でないような複雑なものがいくつか得られた。

$$class(A) \leftarrow \Delta\_mem\_cached(A, upLow) \wedge mem\_free(A, bigMid), \wedge \Delta\_mem\_slab(A, upLow) \wedge \Delta\_swap\_used(A, dwnLow) \quad (2)$$

5 結論

本研究ではILPを用いてサーバーの障害を検知し、原因の解析を行った。今回の実験によってフラッド型のDoS攻撃とメモリリークに関して時系列要素を含むルールを抽出できることを示した。また決定木では表現が難しい、複雑な関係を持つ可読性の高いルールを生成することができた。

参考文献

[1] Tao Wang, Wenbo Zhang, Jun Wei, and Hua Zhong. Fault detection for cloud computing systems with correlation analysis. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 652–658. IEEE, 2015.

[2] Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael I Jordan. Detecting large-scale system problems by mining console logs. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 117–132. ACM, 2009.