学習モデルグラフ上での仮説検証に基づく機械学習モデル生成方法の提案と評価

M2017SE011　白崎　悠太
指導教員　青山　幹雄

1　はじめに

機械学習の発展とともにそれを応用して解決可能な問題領域が広がっている. 様々なドメインにおいて機械学習を用いたソフトウェアシステムを開発する必要性が高まっており, 多くの場合, 教師あり学習によって機械学習モデルを生成している. しかし, 機械学習アルゴリズムを用いるだけでは要求する認識精度を達成するようなモデルを生成することは困難である. 要求される認識精度を達成するために, 学習モデルの最終的な認識精度に基づく仮説検証によってフィーチャ(特徴量)設計とモデルのチューニングが繰り返し行われている. しかし, 学習へのフィーチャの影響分析が未確立であるため, フィーチャの変更によって学習が効率的に行われているかを分析できていない.

本稿ではフィーチャ設計を活かした学習モデル生成プロセスの確立を目的として, 学習過程をプロパティグラフモデルでモデル化することによって, プロパティグラフ上での仮説検証に基づく機械学習モデル生成方法を提案する.

2　研究課題

本稿では, 研究背景を踏まえ以下の 3 点を研究課題として設定する.

(1) プロパティグラフによる機械学習モデルのモデル化方法の確立
(2) プロパティグラフによるフィーチャ分析方法の確立
(3) 実データへの適用による有効性と妥当性の評価

3　関連研究

3.1　機械学習

機械学習とは, データから自動的にパターン認識し予測を行う技術である[2]. データから自動的に学習しモデルを生成するため機械学習の学習過程はブラックボックスであることが知られている. 学習を効率的に行うために, どのフィーチャからモデルを生成するかのフィーチャ選択方法が提案されている[4]. また, フィーチャを制御し学習を効率化するために画像データに対してフィーチャの学習への貢献を分析する方法が提案されている[3]. しかし, 認識精度からの分析であり学習へのフィーチャの影響を十分に分析できていない.

3.2　フィーチャ工学(Feature Engineering)

フィーチャ工学とは, 機械学習の適用対象となる問題の本質を表現したフィーチャをデータから生成する技術体系である[8]. 機械学習の予測精度, 学習コストを向上させる技術である. フィーチャ設計とは, 主に探索的データ分析, データクレンジング, フィーチャ生成, 変換, 選択などのアクティビティから構成され, モデリングの結果によって検証され反復的に行われる. また, データ構造によってフィーチャ設計プロセスは異なり, 画像, 文章, グラフなどに対しては深層学習を用いてフィーチャを抽出する表現学習が提案されている[1].

3.3　プロパティグラフ

プロパティグラフとは, グラフ構造上で表現が困難であったデータ間の意味定義を表現可能とするセマンティックグラフモデルの拡張である. ノードとエッジに属性の集合をプロパティとして付与可能である[6].

4　アプローチ

4.1　フィーチャ工学の導入による学習

本稿では, 教師あり学習を対象とする. 教師あり学習によるモデル生成では, 学習モデルが入力としてとることできるフィーチャの構造や学習アルゴリズムに適した方法でデータを抽象化しフィーチャを獲得するフィーチャ設計を行う必要がある. 特に, フィーチャをテーブルデータから抽象化する場合にはドメイン知識に基づいてフィーチャ選択を反復的に行う方法が効果的である. しかし, データ駆動のアプローチではモデルの予測結果から修正点を特定することは容易ではなく, 試行錯誤を繰り返すことによるコストがかかる. そのため, フィーチャ設計がモデル生成におけるプロセスの中で最も時間的コストがかかるプロセスとなっている. この点に着目し, 本稿では, 学習に対して支配的なフィーチャを特定することによって学習を制御する機械学習モデル生成方法を提案する.

提案方法では, ニューラルネットワークの学習過程からフィーチャ分析を行うアプローチをとる. 学習過程をモデル化し, フィーチャ分析を可能とするため, 新たに, 学習モデルグラフを提案する. 学習モデルグラフとして, 学習モデルの構造的観点からグラフによってモデル化可能, かつバッチサイズやエポックなどの学習に関するデータを付加可能なプロパティグラフを採用した.

4.2　仮説検証型反復学習プロセス

データに対する学習モデルグラフ上での仮説検証型の反復学習プロセスを提案する(図 1). 反復を通して各プロセスを詳細化する. 学習モデルグラフ上で学習過程における学習の差分を分析することによって学習に影響を与えるフィーチャを特定し, それに基づいて学習する.

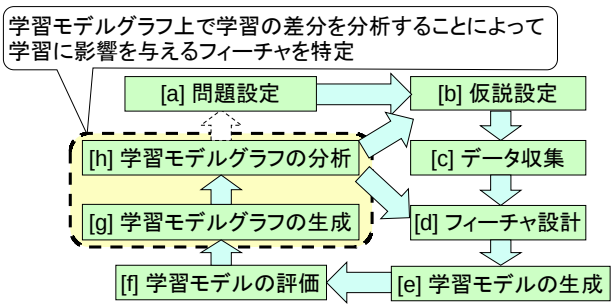

図 1 アプローチ

5 学習モデルグラフ上での仮説検証型機械学習モデル生成方法

5.1 提案プロセス

図 1 のプロセスを詳細化した提案プロセスを図 2 に示す.

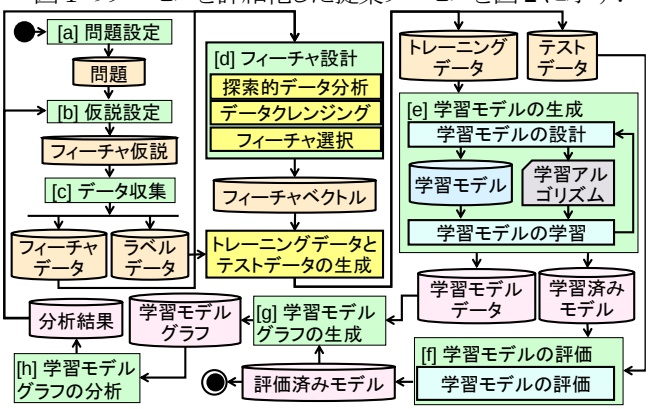

図 2 提案プロセス

[a] 問題設定

機械学習モデルが要求を満たすためにモデルが解くべき問題を設定する. プロセスを繰り返すにつれて問題設定は詳細化される.

[b] 仮説設定

問題解決に必要だと推定されるデータについての仮説を設定する. 学習モデルグラフの分析によって分析対象として収集すべきデータを見直す.

[c] データ収集

仮説から分析に必要なデータを取得する.

[d] フィーチャ設計

学習モデルの入力となるフィーチャベクトルを生成する. 探索的データ分析, データクレンジング, フィーチャ選択の 3 ステップからなる.

[e] 学習モデルの生成

学習モデルを設計と学習の 2 ステップから生成する.

[f] モデルの評価

テストデータを用いて目的関数に基づき学習モデルの汎化誤差を評価する. モデルの評価結果が要求を満たしている場合, 提案プロセスを終了する.

[g] 学習モデルグラフの生成

学習モデルデータからプロパティグラフとして差分を分析する学習モデルグラフを生成する.

[h] 学習モデルグラフの分析

生成した学習モデルグラフから重みの変化率を分析することで, フィーチャの影響力を分析する. 分析結果から仮説設定とフィーチャ設計を見直す.

5.2 フィーチャ設計

(1) 探索的データ分析

フィーチャを抽出するための仮説を設定するために探索的データ分析を行う. 探索データ分析では, データ間の分布と構造を可視化し分析することによって, 有効だと推定されるデータクレンジング方法とフィーチャ選択の仮説を設定する. フィーチャについての仮説はプロセスを繰り返すごとに詳細化する.

(2) データクレンジング

探索的データ分析で明らかになった仮説に基づきデータクレンジングを行う. データクレンジングでは, 欠損値の補完, 不要なデータの削除, 外れ値の削除をし, 形式が異なるファイルから得られるデータを 1 つのデータ形式として再構成する. データクレンジングによってモデルの認識精度の向上を図る.

(3) フィーチャ選択

フィーチャ選択では, 仮説に基づきデータから予測に影響を及ぼすと想定されるフィーチャを選択する. 選択したフィーチャを標準化しフィーチャベクトルとして再構成する.

5.3 学習モデルの生成

(1) 学習モデルの設計

モデル設計では, モデリングに必要なモデル構造と学習アルゴリズムを設計する. 得られるフィーチャの形式と求められる出力の形式に基づいてモデルを設計する. モデル設計後に最適化アルゴリズム, 学習数(epoch), バッチサイズ, 学習率を決定する. 予測結果の精度によって学習モデルの設計を見直す.

(2) 学習モデルの学習

学習アルゴリズムに基づいて設計したセンサ学習モデルのパラメータを最適化する.

5.4 学習モデルグラフの生成

(1) 学習モデルグラフの定義

学習モデルグラフとは, 学習モデル構造と学習のための付加データをプロパティグラフとして表現したグラフである.

ニューラルネットワークの学習モデル構造を表現可能なプロパティグラフモデルを定義する. 学習モデルデータから生成する学習モデルグラフのメタモデルの定義を図 3 に示す.

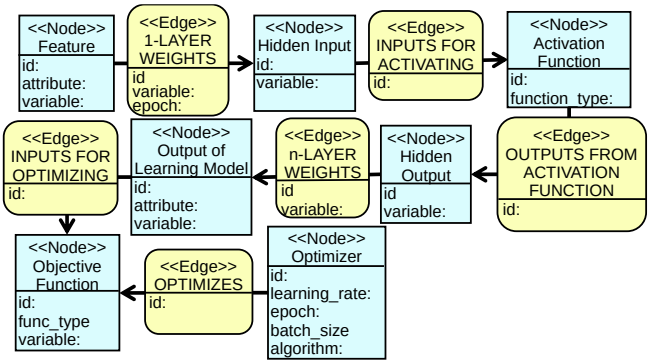

図 3 学習モデルグラフのメタモデル

表 1 にノードの定義と付与するプロパティ, 表 2 にエッジの定義と付与するプロパティ, 表 3 にノードとエッジに付与したプロパティの定義を示す.

表 1 ノードの定義とプロパティ

ノード	定義	プロパティ
Feature	学習モデルへの入力(フィーチャ)	attribute, variable
Hidden Input	活性化関数への入力	variable
Activation Function	活性化関数	id, func_type
Hidden Output	活性化関数の出力	variable
Output of Learning Model	モデルの最終出力	attribute, variable
Objective Function	目的関数	id, func_type, variable
Optimizer	最適化情報を保持するノード	id, learning_rate, epoch, batch_size, algorithm

表 2 エッジの定義とプロパティ

エッジ	定義	プロパティ
1-LAYER WEIGHTS	各ノード間をつなぐ第 1 層の重み付け	id, variable, epoch
INPUTS FOR ACTIVATING	活性化関数ノードへの入力	id
OUTPUTS FROM ACTIVATION FUNCTION	活性化関数ノードからの出力	id
n-LAYER WEIGHTS	各ノード間をつなぐ第 n 層の重み付け	id, variable
INPUTS FOR OPTIMIZING	目的関数への入力	id
OPTIMIZES	パラメータを最適化	id

表 3 プロパティの定義

プロパティ	定義
id	ノードとエッジの id
attribute	フィーチャの属性名
variable	ノードとエッジが保持する値
func_type	関数名
learning_rate	学習率
epoch	学習数
batch_size	バッチサイズ
algorithm	最適化アルゴリズム

(2) 学習モデルグラフの生成

メタモデルの定義にしたがってグラフデータベースにデータを挿入し学習モデルグラフを生成する.

5.5 学習モデルグラフの分析

重みの平均変化率の大きい重みと結ばれているフィーチャノードを特定することによって学習に影響を与えるフィーチャを特定する(図 4). そのために学習数 e から学習数 e'における第 l 層 i 番目のフィーチャノードに連結する重みの平均変化率を式(1)により定義する.

$$R_j^{(l)} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|w_{i,j,e'}^{(l)} - w_{i,j,e}^{(l)}\right|}{\left|w_{i,j,e}^{(l)}\right|} \quad (1)$$

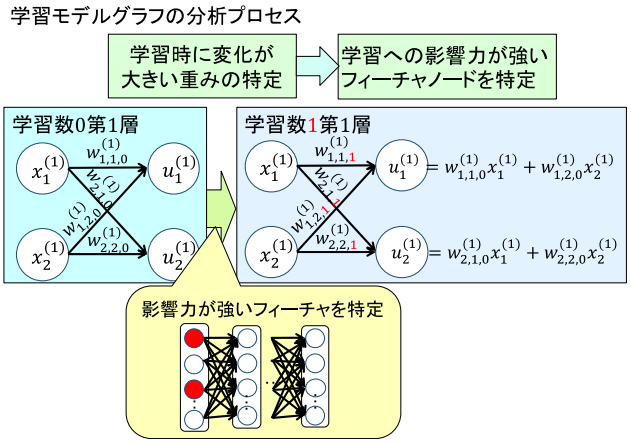

図 4 フィーチャの影響分析

6 プロトタイプの実装

6.1 実装環境

プロトタイプの実装環境を表 4 に示す.

表 4 ソフトウェアコンポーネント

コンポーネント	コンポーネント名	バージョン
OS	Ubuntu	18.04
実装言語	Python	3.6.4
深層学習フレームワーク	Chainer	4.2.0
評価結果の可視化	ChainerUI	0.2.0
データ加工ツール	Pandas	0.20.3
可視化フレームワーク	Dash	0.30.0
グラフデータベース	Neo4j	3.4.7

6.2 プロトタイプのアーキテクチャ

提案方法を評価するために実装したプロトタイプのアーキテクチャを図 5 に示す.

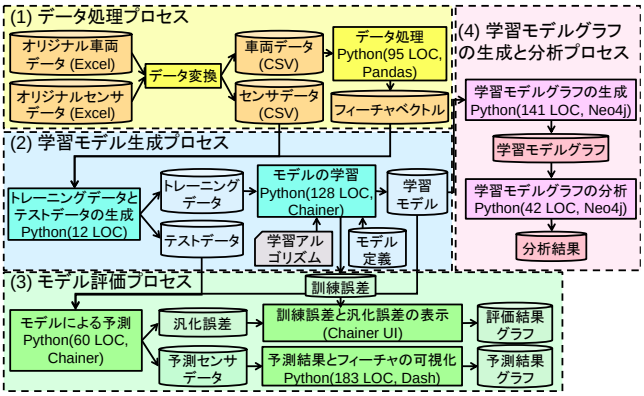

図 5 プロトタイプのアーキテクチャ

7 自動車超音波センサデータへの適用と評価

7.1 適用の目的

実際の車両データとセンサデータに対してプロトタイプを適用し, 提案方法の有効性と妥当性を評価する.

7.2 適用対象

路面反射波のセンサデータに対して提案方法を適用した. データの特徴を表している電圧の最大値付近を抽出するために最大値付近のデータ以外をクレンジングした. 学習モデルが出力するデータ数を表 5 に示す. 直接波とは, 超音波の発信と受信を同一のセンサが行う場合であり, 間接波とは別のセンサが行う場合である. センサがドライであるとはセンサに水滴が付着していない状態であり, ウェットは水滴が付着している状態を示す.

表 5 適用対象の出力データ数

	反射時間 [ms]		データ数
クレンジング前(全データ)	min	max	
	0.15	37.95	253
クレンジング後(直接波)	7.35	10.35	20
クレンジング後(間接波)	7.35	14.85	50

7.3 フィーチャの影響分析

重みの平均変化率を図 6 と図 7 に示す. 直接波(5-10 epoch)と間接波(25-30 epoch)で重みの平均変化率がピークを迎えるエポック数に違いがある. これは直接波(5 個)よりも間接波(10 個)の入力フィーチャ数が多く, 学習の進行が遅かったことに起因すると推定される. また, 直接波よりも間接波の影響度が高いフィーチャ数が 2 個多い. これは直接波よりも間接波の方が反射波電圧(出力)に影響を与えるセンサの環境条件が多いことに起因すると推定される.

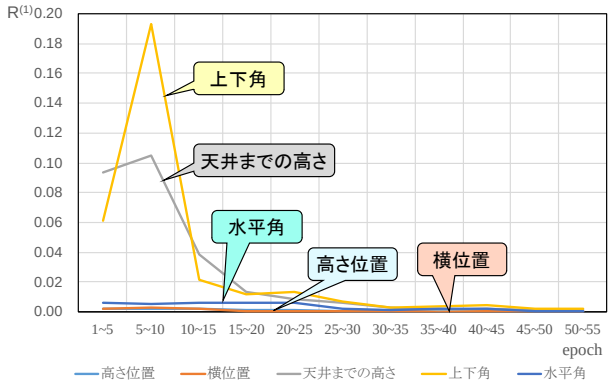

図 6 重みの平均変化率(直接波)

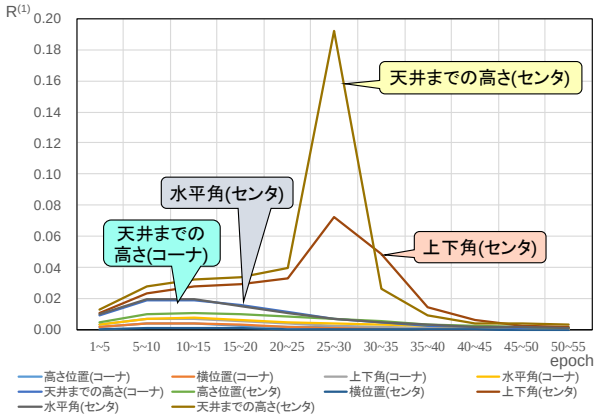

図 7 重みの平均変化率(間接波)

7.4 評価方法

評価関数として平均二乗誤差 式(4)を採用し訓練誤差と汎化誤差の評価をした. nは入力データ数, f_iはi番目の入力x_iに対するモデルの出力, y_iはi番目の入力x_iに対する教師データの反射波電圧を表す.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \qquad (1)$$

7.5 評価結果

評価方法にしたがい生成した学習モデルの認識精度を評価した結果を図 8 に示す.

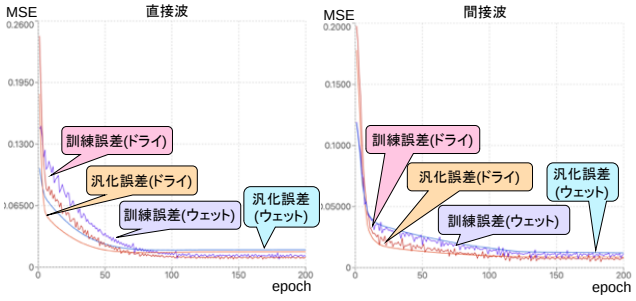

図 8 認識精度

8 考察

8.1 学習モデルグラフによる学習のモデル化

提案した学習グラフモデルはグラフモデルの中でも表現能力の高いプロパティグラフを用いて定義することによって, 分析に必要なデータを捨象することなくニューラルネットワークの構造と学習過程を表現できる. クエリによりサブグラフを取り出し可視化することで設計した学習モデルを分析することが可能になり, 学習モデル設計の効率化が期待できる.

8.2 学習モデルグラフの分析と反復学習プロセス

特定した学習に影響を与えるフィーチャに基づき, フィーチャ選択の検証が可能になる. 従来の試行錯誤を伴う発見的なアプローチにより多大な時間がかかっているフィーチャ設計を効率化可能になる. さらに, フィーチャ分析に基づく反復学習プロセスにより学習を一定の水準で制御可能になると期待できる. これによって, フィーチャ設計とそれを用いた反復プロセスによる機械学習モデル生成がソフトウェア工学と同様構造化された工学的方法となることが期待できる.

8.3 実データへの適用による有効性と妥当性評価

性質が異なる直接波と間接波の反射波電圧データに対して影響度が高いフィーチャを特定可能であることを確認した. さらに, 特定したフィーチャを中心に一定の精度で学習モデルを生成可能であることを確認した.

8.4 先行研究との比較

先行研究[3,4]ではモデルの出力となる認識精度からフィーチャの影響度を分析している. しかし, 入力フィーチャによっては認識精度に現れないフィーチャの影響がある. 本稿では局所的な分析によって影響力が強いフィーチャを特定した.

9 今後の課題

(1) プロパティグラフの特性を活かした分析方法の検討
(2) 他データ, 他の学習モデルへの適用

10 まとめ

本稿では学習過程を学習モデルグラフとしてモデル化しフィーチャの学習への影響力分析による学習モデルグラフ上での仮説検証に基づく反復プロセスによる機械学習モデル生成方法を提案した. 学習に影響を与えるフィーチャの特定とそれを用いた反復プロセスによって, 試行錯誤を伴う発見的なアプローチによる学習モデル開発における多大なコストの低減と学習プロセスの制御が可能になる. 提案方法は機械学習システムの開発プロセスがソフトウェア開発プロセスと同様な工学的プロセスとなる一アプローチとして期待できる.

謝辞: センサデータをご提供頂いた株式会社デンソーの関係各位とご討論頂いた林健吾氏に感謝する.

参考文献

[1] G. Dong, et al., Feature Engineering for Machine Learning and Data Analytics, CRC Press, 2018.
[2] I. Goodfellow, et al., Deep Learning, MIT Press, 2016.
[3] P. W. Koh and P. Liang, Understanding Black-box Predictions via Influence Functions, Proc. of Machine Learning Research, Vol. 70, Jul. 2017, pp. 1885-1894.
[4] J. Li, et al., Feature Selection: A Data Perspective, ACM Comput. Surv., Vol. 50, No. 6, Dec. 2017, 45 pages.
[5] Neo Technology, neo4j, 2016, http://neo4j.com/.
[6] I. Robinson, et al., Graph Databases, 2nd ed., O'Reilly, 2015.
[7] S. Ozdemir, et al., Feature Engineering Made Easy, Packt, 2018.
[8] Preferred Networks, Chainer, https://chainer.org/.
[9] 白崎 悠太, 他, セマンティックグラフモデルを用いたステークホルダ分析方法の提案と評価, SES 2017 論文集, 情報処理学会, Aug.-Sep. 2017, pp. 98-105.
[10] 白崎 悠太, 他, 深層学習を用いた超音波センサのモデル化方法と適用評価, 自動車技術会 2018 年秋季大会, 講演予稿集, 自動車技術会, Oct. 2018, pp. 1-6.