

# 一般道実走行における 自車センサ情報を用いた車線変更の検出

根山 亮<sup>1,a)</sup> 福島 真太郎<sup>1</sup> 宮崎 勝彦<sup>2</sup> 那和 一成<sup>1</sup>

**概要：**本論文では、速度や操舵角など安価な自車センサ情報のみを用いた車線変更検出手法について述べる。車両走行データは、a) 多変量かつ時系列、b) 走行時の周辺環境により変化しやすい、c) 明確な区切りがないという特徴を持つ。そのため、分析に用いる特徴量やアルゴリズムは自明ではない。提案手法では、複数のセンサ情報を組合せ、データを数秒間の窓に分割し、汎用的な教師あり学習アルゴリズムにより車線変更を検出する。一般道実走行データを用いた実験において、アンサンブル学習やインスタンスに基づく学習を用いれば、他のアルゴリズムより高い検出精度が得られることを確認した。

## 1. まえがき

移動体通信の広帯域化や計算機の低消費電力化を背景に、ネットワークを介して通信し、安全かつ効率的に走行できる次世代の自動車（コネクティッド・カー）が注目されている。

一方で、電気自動車の普及により、今後、車の“走る・曲がる・止まる”機能のコモディティ化が進む可能性がある。そのため、高度運転支援や自動運転といった、より高い安全性や快適性を提供する高付加価値なサービスの創出が求められている。

こうした中、大量の車両走行データから得られる集合知が新価値創造のブレークスルーになりえると我々は考える。コネクティッド・カーから大量の車両走行データを収集、分析すれば、集合知を使った運転支援など新たなサービスを創出できる可能性がある。

我々は、ドライバーに負担のかかる運転行動の1つである車線変更に着目し、車両走行データから、車が車線変更をしている区間を検出する手法の開発に取り組んだ。車線変更を精度よく検出できれば、その結果を統計処理し、運転中のドライバーに対して車線変更すべきポイントを事前に案内するなど、これまでではできなかった細やかな情報提供が可能になる。我々の提案するシステムおよびサービスの構成を図1に示す。

我々の取り組む車線変更検出の特長は、一般道の実走行データを対象とし、車両の速度や操舵角など、比較的安価

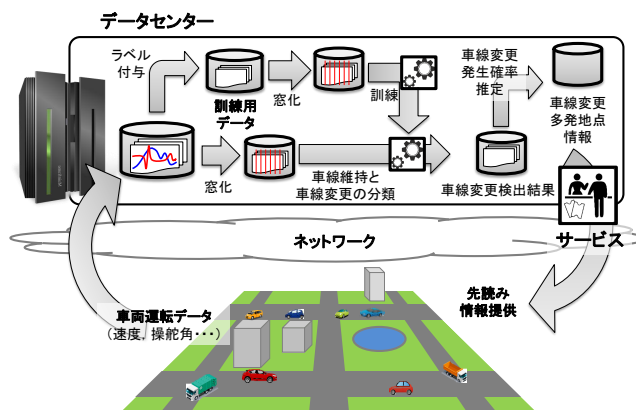


図1 提案システム・サービスの構成

な自車センサ情報のみを入力属性として用いることである。一般道の走行は高速道のそれと比較し、道路形状が複雑で周辺車や信号などによる外乱が多く、車線変更の検出は難しい。しかし、サービスの適用範囲を広げるには、一般道でも車線変更を検出できることが望ましい。また、カメラなど高価なセンサに頼ることなく、安価な自車センサ情報のみを使って車線変更を検出できれば、より多くの車線変更発生情報を集められる。これにより、サービスの適用範囲をさらに広げられる。

車両走行データは多変量かつ明確な区切りのない時系列データであり、分析に用いる特徴量やアルゴリズムは自明ではない。我々が取り組む課題は、1) 車線変更の検出に有効な入力属性の選択、2) 切れ目のない時系列データからの特徴ベクトルの抽出、および3) 適用するアルゴリズムの選定の3点である。

<sup>1</sup> 株式会社トヨタ IT 開発センター

<sup>2</sup> トヨタ自動車株式会社

<sup>a)</sup> ryo-neyama@jp.toyota-itc.com

表 1 車線変更検出に使う入力属性

入力属性	自転車情報 (本論文)	周辺環境 情報	ドライバ 情報
センサ	速度, 操舵角など	カメラなど	カメラなど
可用性	高 常時高精度	低 環境により 低信頼または 利用不可	低 環境により 低信頼または 利用不可
経済性	高 大衆車にも 搭載可	低 高級車のみ 搭載可	低 高級車のみ 搭載可
容易性	低 自転車挙動のみ 利用	高 車線内位置など 利用	高 顔向きなど 利用

### 1.1 自転車情報のみを使うことの重要性

一般に, 車線変更検出に用いられる入力属性は, 表 1 に示すように 1) 自転車情報, 2) 周辺環境情報, および 3) ドライバ情報の 3 種類が考えられる. 我々は, 車両の速度や操舵角など “自転車情報” のみを使う手法が可用性と経済性が高く有望と考える. “周辺環境情報” や “ドライバ情報” は, センサが高価で環境条件に左右される<sup>\*1</sup> ため経済性と可用性が低い. 高価なセンサは台数の少ない高級車にしか搭載できないため, 大量のデータを集めにくい. 以上の理由から, 車線変更を検出した結果を統計処理し集合知として活用するには “自転車情報” のみを使うのが望ましいと判断した.

### 1.2 関連研究

車線変更を検出する手法は, これまでにいくつか報告されている. それらを 1) 入力属性, 2) 訓練・評価用のデータセット, および 3) アルゴリズムに分類したものを表 2 に示す.

車線変更検出に使う入力属性としては, ほとんどの先行研究において, 自転車情報のみならず, 車線内位置や先行車距離といった周辺環境情報を利用し [7][8][12], さらにドライバの顔向きや視線といったドライバ情報を利用していることが分かる [9][10]. これは, 車線変更するドライバの意図を早期に予測するためであるが, 利用する情報が増えれば, より多くのセンサが必要で, 経済性を損なう.

車線変更を学習し, 精度を評価するために使うデータセットとしては, かつては取得が容易なことからドライビング・シミュレータの仮想的なデータが使われていたが [6][12], 近年は高速道を運転した実データが使われるようになり [7][8][9], さらに一般道の実データも使われている [10]. 一般に, 車線変更の検出はシミュレータ・データより実データの方が難しく, 実データの中でも高速道より一般道の方が困難である. 表 2 から分かるように, 我々が対象とする自転車情報だけを使った一般道における車線変更

\*1 交差点での白線途切れ, 不鮮明な白線, トンネル内や都市部ビル群下での衛星電波受信難, 降雨や陰影による不鮮明化などによる.

検出に関する研究事例は見あたらない.

車線変更を検出するアルゴリズムには, SVM (Support Vector Machine) や RVM (Relevance Vector Machine) が使われることが多い [7][8][9][10]. 一般道での車線変更検出には大量のデータを効率よく学習し, かつ, 複雑な構造を持つデータであっても過学習を抑えるロバスト性が求められることから, 本論文ではアンサンブル学習およびインスタンスに基づく学習を採用した.

なお, 車両挙動を解析することを目的として, 教師なし学習により車両走行データを記号化する手法も提案されている [13][16][17]. これらの研究は, 車両走行データ分析のための汎用的な基盤を提供することを目指しており, 車線変更を含む車両挙動の分類にも適用できる可能性がある.

### 1.3 本論文の貢献

提案手法では, 1) 速度や操舵角などの複数の自転車センサ情報を組合せ, 2) データを数秒程度の時間窓に区切った生データを使い, 3) 汎用的な教師あり学習アルゴリズムにより, 車線変更を検出する. 実験の結果, 東京都内の一般道を走行する実データにおいて, アンサンブル学習やインスタンスに基づく学習により, 他のアルゴリズムより高い検出精度が得られることを確認した.

## 2. 自転車情報を用いた車線変更検出

ここでは, 自転車情報を用いた車線変更検出の問題設定と解決手段を述べる.

### 2.1 対象とするデータ

$\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$  を多変量時系列データとし,  $\mathbf{x}_t$  を時刻  $t$  における  $d$  次元ベクトルとする.  $\mathbf{x}_t = \{x_t^{(1)}, \dots, x_t^{(d)}\}$  と表され,  $x_t^{(i)} \in \mathbb{R}$  は時刻  $t$  における  $i$  次元目の入力属性を表す. 本論文では, 入力属性として, 速度, 前後加速度, ヨー角速度<sup>\*2</sup> および操舵角の一部または全部を用いる.

データ  $\mathbf{X}$  には, 車両が車線を維持しているか, 変更しているかを表すラベル  $\mathbf{Y} = \{y_1, \dots, y_t\}, y_t \in \{0, 1\}$  が付与される. 時刻  $t$  において, 車両が車線を維持していれば  $y_t = 0$ , 変更していれば  $y_t = 1$  となる. 具体的には, 図 2 に示すように, 車線を区分する白線上に車両がある状態を  $y_t = 1$  とし, それ以外を  $y_t = 0$  とする<sup>\*3</sup>.

データは, 東京都内一般道の特定経路を 2 名のドライバが, のべ 110 回繰り返し走行し, 取得した. サンプルング周期は 10Hz とした.

生データの例を図 3 に示す. 車線を維持している区間と車線を変更している区間における各センサ値をそれぞれ重

\*2 車両の道路面に対し水平方向の角速度.

\*3 車線維持と車線変更の境界付近での中間的な車両挙動による検出精度への影響について, 第 3.2 節で議論する.

表 2 車線変更検出手法の比較

		本論文	Mandalia2005[8]	Kuge2000[6]	Kumar2013[7]	Morris2011[10]	Salrucci2004 [12]	McCall2007 [9]
入力属性	自車情報	✓	✓	✓	✓	✓	✓	✓
	周辺環境情報		✓		✓	✓	✓	✓
	ドライバ情報					✓		✓
データセット	シミュレータ			✓			✓	
	高速道		✓		✓	✓		✓
	一般道	✓				✓		
アルゴリズム	Ensemble, Instance-based	SVM	HMM	SVM+BF	RVM	Model Similarity	RVM	

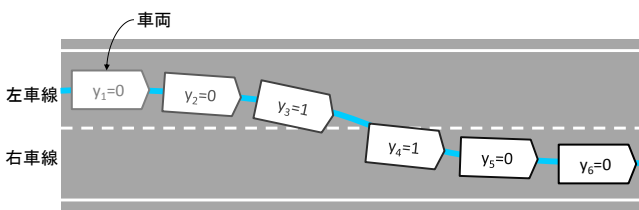


図 2 ラベル付与の方法

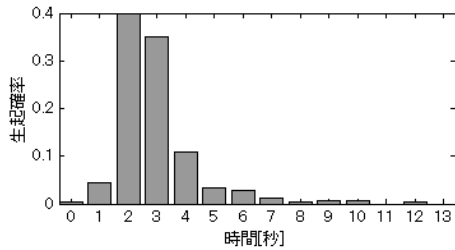


図 4 車線変更にかかる時間の分布

ねて図示している。この図から車線維持と車線変更の間に明らかな差異を見つけるのは難しいが、強いて特徴を述べれば、車線変更中は、車線維持中に比べ、1) 速度が維持されやすく、2) ヨー角速度と操舵角が特徴的な変化をしやすいと考えられる。

車線変更を精度よく検出するには、その特徴を捉えるのに適切な時間窓長を知る必要がある。図 3 に示唆されるように、1 回の車線変更にかかる時間は必ずしも同じではない。データ中に含まれる 357 回の車線変更にかかった時間の分布を図 4 に示す。多くの車線変更はおおよそ 2 ~ 4 秒程度の時間を要している (平均 3.40, 標準偏差 1.47)。このことから、入力データを数秒程度の時間窓に分割して扱えば、車線変更の特徴を取り出せると考えられる。

## 2.2 前処理の検討

車線変更の特徴を捉えるのに適切な入力属性の組合せは自明ではなく、実験的に確かめる必要がある。そこで、属性

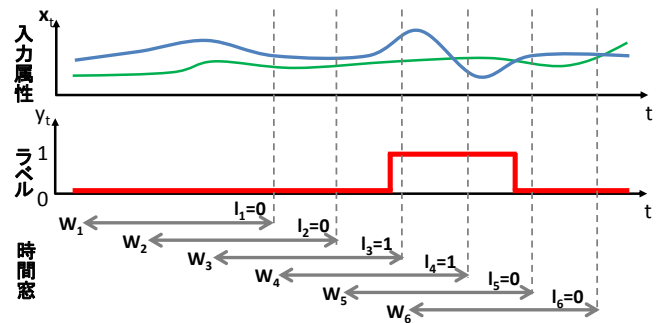


図 5 時間窓の定義

集合  $S = \{ \text{速度, 前後加速度, ヨー角速度, 操舵角} \}$  の、すべての空でない部分集合  $T (T \subseteq S, T \neq \emptyset)$  に対して網羅的に試すこととした。

次に、入力データから数秒程度の時間窓を作成する。時間窓を直観的に理解するため、その定義を図 5 に示す。車線変更を細かい分解能で検出するため、時間窓には一定間隔の重なりを持たせ、時間窓を  $W_k = \{ \mathbf{x}_{sk+1}, \dots, \mathbf{x}_{sk+w} \}$  と定義する。ただし、 $w$  は窓幅、 $s (< w)$  は移動幅、 $k$  は非負整数である。

時間窓  $W_k$  には車線変更しているか否かのラベル  $l_k = y_{sk+w}$  が付与される。つまり、時間窓に付与するラベルとして時間窓内の最後の要素  $\mathbf{x}_{sk+w}$  に付与されたラベル  $y_{sk+w}$  を使う。最後の要素が車線変更であるかどうかは、時間窓内の系列  $\{ \mathbf{x}_{sk+1}, \dots, \mathbf{x}_{sk+w} \}$  によって決まると考えるためである。

最後に、アルゴリズムに入力する特徴ベクトルを定義する。ここで、関数  $\mathcal{F}_T(\mathbf{x}_t)$  を  $\mathbf{x}_t$  から属性集合  $T$  に含まれる属性を取り出す手続きとすると、時間窓  $W_k$  に対応する特徴ベクトル  $\mathbf{F}_k$  は  $\mathbf{F}_k = \{ \mathcal{F}_T(\mathbf{x}_{sk+1}), \dots, \mathcal{F}_T(\mathbf{x}_{sk+w}) \}$  と定義される。なお、特徴量は入力属性ごとに平均 0, 標準偏差 1 となるように正規化 ( $z$ -normalization) する。

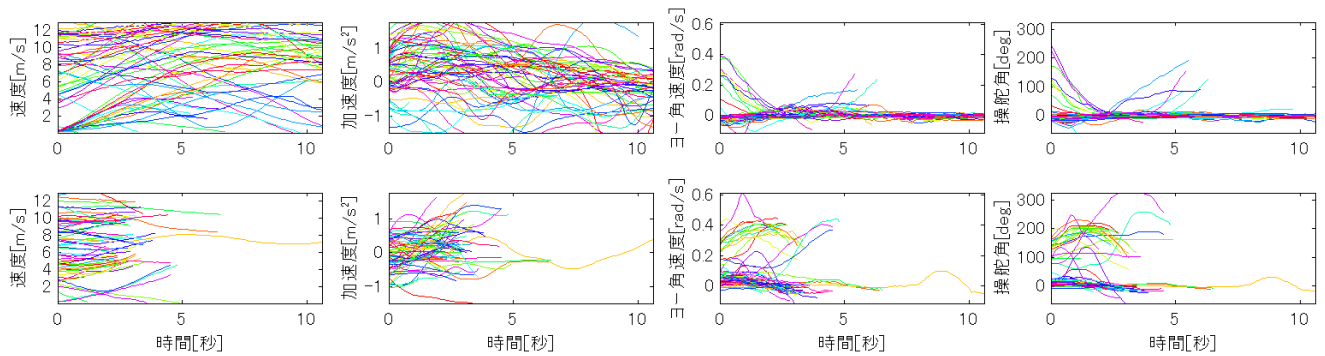


図 3 生データの例；車線維持（上段），車線変更（下段）

## 2.3 問題設定

本論文で解く問題を次のように定義する。

[前提] 訓練用データ  $D_{tr} = \{X_{tr}, Y_{tr}\}$ ，評価用データ  $D_{te} = \{X_{te}, Y_{te}\}$ ，窓幅  $w$ ，移動幅  $s$ ，および関数  $\mathcal{F}_T(x_t)$  が与えられる。

[問題]  $D_{tr}$  から作成された時間窓とラベルをアルゴリズムにより学習し，その結果を用いて  $D_{te}$  から作成された時間窓  $W_k$  のラベル  $l_k$  を予測する．予測されたラベルを  $\hat{l}_k$  としたとき， $\hat{l}_k = l_k$  であれば正解，さもなければ不正解とする。

## 2.4 提案手法

車線維持と車線変更を分類するアルゴリズムとして，我々はアンサンブル学習およびインスタンスに基づく学習に着目した。

アンサンブル学習は，単独では精度の低い複数の弱分類器を組合せ，汎化性能を向上させるアルゴリズムである．アンサンブル学習には，訓練データを複数のグループ<sup>\*4</sup>に分けてグループごとに弱分類器を作るバギングや訓練データから得られた予測誤差からデータの重み付けを変えて精度を上げるブースティングが知られる。

こうしたアンサンブル学習の特長は車線変更の検出に対しても有効であると我々は考える．車線変更を含む運転行動は，動的な周辺環境により大きな影響を受ける．例えば，路上駐車を避けるための回避を伴う車線変更もあれば，カーブ付近では車線維持と見分けにくい車線変更もある．そのため，重要な特徴量はその時々周辺の環境によって異なる．アンサンブル学習はこうした複雑なデータに適していると考えられる。

本論文では，バギングの1つとして *Random Forests* (**RF**) [2] を採用した．**RF** は，入力として与えられた全訓練データを  $T$  組の群に無作為に分け， $T$  個の決定木を作成する<sup>\*5</sup>．決定木を作成する過程においては， $a$  個の属

性を無作為に選び，分類にもっとも寄与する属性を選択する．全データの全属性を使う手法と比べ，ノイズに対するロバスト性が高いとされ，包括的なデータセットを用いた精度評価においても高い精度が得られることが報告されている [3]．

また，ブースティングの1つとして *LogitBoost* (**LB**) [5] を採用した．**LB** は  $J$  個の弱分類器を加法モデルとして組合せ，強学習器を作成する．訓練時には， $1, \dots, j-1$  番目の弱学習器による分類結果から各サンプルのクラス確率を求め，分類境界付近の誤分類されやすいサンプルに，より多くの重みを割り振ることによって  $j$  番目の弱学習器を作成する．訓練データ量が少ない場合にも  $J$  を十分に大きくすることで，複雑な構造を持つデータを分類できることが期待される．本論文では，弱学習器として回帰株（1次元かつ1階層の回帰木）を用いた。

インスタンスに基づく学習 [15] も有望である．前述の通り，車両挙動は周辺環境により大きく変化しうる．そのため，単純なモデルでは表現できない可能性がある．一方で，インスタンスに基づく学習は，データの構造が複雑でモデル化が難しい場合にも，訓練データ量を十分に増やせば高い精度を達成できる可能性がある。

本論文では，インスタンスに基づく学習として  $k$ -最近傍法を用いた． $k$ -最近傍法では，あるサンプルが与えられると，全訓練データのうちサンプルとの距離が上位  $k$  件に含まれる訓練データに付与されたラベルの多数決により，サンプルのラベルを決める．本論文では，距離基準として時間方向の伸縮を許さないユークリッド距離 (**EC+kNN**) と伸縮を許す *Dynamic Time Warping* (**DTW+kNN**) [11] を用いた．訓練データ量が少ない場合は伸縮を許す **DTW+kNN** の性能がよく，訓練データ量を十分に増やせるのであれば **EC+kNN** が有効になることが報告されている [14]．

以上の理由から，本論文ではアンサンブル学習 (**RF**，**LB**) およびインスタンスに基づく学習 (**EC+kNN**，

\*4 1つのサンプルが複数のグループに重複して属することを許す。

\*5 決定木については，第3節で述べる。

DTW+kNN) を採用した。

### 3. 評価実験

提案手法の有効性を確かめるため、第2節で述べた東京都内の一般道を走行した実データを用いた実験を行った。実験は、Intel Core i7/2.9GHzのCPU、16GBのRAMを搭載したMicrosoft Windows 7 Pro 64bitの計算機上で行った。

実験では、110回の全走行を5つの群に無作為に分け、5分割交差検定を行い、精度とそのばらつきを評価した。訓練用と評価用の走行データを分離することで、実用に近い条件での精度を導く。

時間窓の窓幅は  $w = 50, 75, 100, 125, 150$  (各5, 7.5, 10, 12.5, 15秒間隔)とし、移動幅は  $s = 10$  (1秒)とした。訓練と評価に用いるデータの窓数は窓幅や分割により異なるが、一例を挙げると\*6、訓練用データでは車線変更が873サンプル、車線維持が7,428サンプルであり、評価用データではそれぞれ210サンプル、1,815サンプルである。

本データは車線変更の窓数が車線維持の窓数に対して少なく(約1:9)、不均衡である。一般に、クラス数が不均衡なデータを使って訓練を行うと、分類結果が多数を占めるクラスに偏り、精度が低下することが知られており、それを解決する手法が提案されている[4]。

本論文では、クラス数が不均衡のまま訓練する実験に加え、車線維持を無作為にアンダーサンプリングすることで、車線維持の窓数を車線変更の窓数に近づける実験も行った。アンダーサンプリングした車線維持の窓数の比率は、車線変更の窓数1に対して各1, 2, 4, 8である。

実験に使うアルゴリズムは、第2節で述べたRF, LB, EC+kNN および DTW+kNN に加え、以下のものを比較対象とする。

[SVM (SVM)] 高い汎化性能を持つ分類アルゴリズムの1つで、車線変更を検出する先行研究でもよく使われている[8][7][9]。データに合わせて適切なカーネル関数を選べば、精度を向上できる可能性がある。

[決定木 (DTree)] クラスを木構造により分類するアルゴリズムである。RFが木構造を複数作成するのに対し、DTreeは木構造を1つしか作成しないため過学習しやすいと考えられる。RFとDTreeの精度を比較することで、RFのロバスト性を検証できる。決定木には複数の手法が存在するが、本論文ではCARTを用いる。

[ランダム (RND)] 車線維持と車線変更を一様乱数により分類する。比較のベースラインとして用いる。

表3 アルゴリズムとパラメータ

アルゴリズム	パラメータ
共通	窓幅 $w = 50, 75, 100, 125, 150$ 窓数比率 $r = 1, 2, 4, 8, -$ (不均衡)
RF	木の数 $T = 100, 150, \dots, 300$ 属性数 $(\dim(\mathbf{F}_k))^q, q = 0.4, 0.5, 0.6$
LB	弱学習器数 $J = 200$
EC+kNN	$k$ -最近傍法の $k = 1, 3, 5, 7, 9$
DTW+kNN	DTWのバンド幅=窓幅の20% $k$ -最近傍法の $k = 1, 3, 5, 7, 9$
SVM	カーネル=線形, RBF( $\sigma = 3$ )
DTree	手法=CART
RND	車線変更の分類確率 $p = 0.1, 0.2, \dots, 0.9$

表4 混同行列

		予測されたラベル	
		車線変更	車線維持
真のラベル	車線変更	真陽性 $T_p$	偽陰性 $F_n$
	車線維持	偽陽性 $F_p$	真陰性 $T_n$

#### 3.1 精度

アルゴリズムとパラメータを変えて、複数の条件下での実験を行った。使用したアルゴリズムとパラメータを表3に示す。

精度の評価指標として、適合率、再現率、F-値、および正解率を求めた。適合率  $P$ 、再現率  $R$ 、F-値  $F$  は、車線変更の検出に着目した指標で、表4に示す混同行列 (confusion matrix) および式(1)により定義される。

$$\begin{aligned}
 P &= \frac{T_p}{T_p + F_p} \\
 R &= \frac{T_p}{T_p + F_n} \\
 F &= \frac{2PR}{P + R}
 \end{aligned} \tag{1}$$

一方、正解率  $A_{cc}$  は、車線維持と車線変更を正しく分類した割合を表し、式(2)により定義される。

$$A_{cc} = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{2}$$

一般に、車線変更は車線維持に比べて発生頻度が低く、車線変更を精度よく検出するのは難しい。しかし、第4節で述べる車線変更の発生頻度を使ったサービスを実現するには、車線変更を精度よく検出する必要がある。すなわち、正解率が高いことよりも、F-値が高いことがより強く求められる。そのため、本論文では、まず結果のうちF-値が高いものに着目し、次に正解率が高いものに着目する。なお、一般に適合率と再現率の両立は難しいが、誤警報を防ぎたいサービス側の要求から適合率が高いことを重視する\*7。

5分割交差検定の結果、F-値が平均的に最大となる実験

\*7 ドライバに注意を促すサービスでは、誤検出が多いとドライバーがその機能を見捨てるもしくは無効化してしまう。一方、ドライバーが主体の運転では、未検出はある程度許される。

\*6 窓幅  $w = 75$ 、車線維持のアンダーサンプリング(後述)を行わない場合。

表 5 各アルゴリズムの精度 (F-値最大; 括弧内は標準偏差)

アルゴリズム	適合率	再現率	F-値	正解率
RF	0.733 (0.032)	0.762 (0.036)	0.747 (0.024)	0.968 (0.005)
LB	0.695 (0.034)	0.748 (0.050)	0.720 (0.036)	0.967 (0.004)
EC+kNN	0.712 (0.036)	0.723 (0.047)	0.717 (0.037)	0.964 (0.005)
DTW+kNN	0.712 (0.036)	0.723 (0.047)	0.717 (0.037)	0.964 (0.005)
SVM	0.652 (0.041)	0.745 (0.041)	0.694 (0.034)	0.965 (0.005)
DTree	0.633 (0.048)	0.623 (0.049)	0.627 (0.039)	0.952 (0.007)
RND	0.107 (0.007)	0.924 (0.023)	0.193 (0.011)	0.117 (0.008)

条件における結果の、アルゴリズムごとの平均値および標準偏差を表 5 に示す。また、その際の実験条件を表 6 に示す。

RF において、最大の適合率、F-値および正解率が得られた。SVM や DTree より相対的な精度が高い。また、RF の F-値のばらつきは比較的小さい。したがって、RF のノイズに対するロバスト性が裏付けられた。窓幅  $w = 75$  (7.5 秒間) であり、LB や SVM と比べ短い期間の情報を使って分類している。

LB においても、比較的高い F-値が得られた。一方、窓幅は  $w = 125$  (12.5 秒間) で、候補として与えた全属性を使っており、多くの情報を必要とする。計算コストを考えれば、少ない情報で車線変更を検出できるのが望ましい。

EC+kNN および DTW+kNN も比較的高い F-値が得られた。窓幅は  $w = 75$  (7.5 秒間) で、ヨー角速度と操舵角という車両の主に横方向の運動に関連する属性を使い、少ない情報から分類している。短い時間や限られた属性での車線変更検出が可能であれば、計算コストの観点から有利である。

時間方向の伸縮の観点からは、伸縮を許す DTW+kNN は伸縮を許さない EC+kNN と一致する結果となった。精度が同程度であれば、計算コストの小さい EC+kNN が優れていると言える。先行研究 [14] によると訓練データ量が少ない場合は DTW+kNN の方が高い精度が得られると想定されたが、今回は伸縮を許す効果は得られなかった。原因については、今後、データ量や DTW のバンド幅を変えて、検証する必要がある。

各アルゴリズムの精度のばらつきを見ると、正解率は比較的ばらつきが小さいのに対して、F-値のばらつきが大きい。上位 4 アルゴリズム (RF, LB, EC+kNN, および DTW+kNN) の F-値の平均値にはあまり差がなく、精度の有意差は今後検証が必要である。

これまでの、F-値が最大となる結果を見てきたが、同様に、正解率が平均的に最大となる実験条件における結果を見る。

表 6 各アルゴリズムの実験条件 (F-値最大)

アルゴリズム	$w$	$r$	入力属性	パラメータ
RF	75	4	速度 ヨー角速度 操舵角	$T = 300$ $q = 0.4$
LB	125	4	速度 前後加速度 ヨー角速度 操舵角	$M = 200$
EC+kNN	75	4	ヨー角速度 操舵角	$k = 9$
DTW+kNN	75	4	ヨー角速度 操舵角	$k = 9$
SVM	125	-	ヨー角速度	$RBF$ $\sigma = 3$
DTree	75	-	速度 ヨー角速度	-

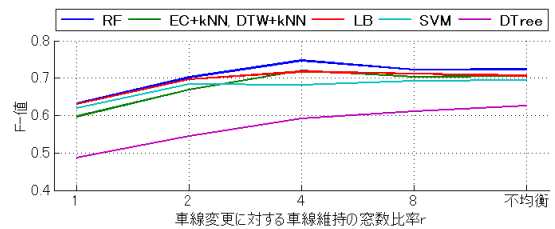


図 6 F-値に対するアンダーサンプリングの影響

結果は割愛するが、いずれのアルゴリズムにおいても、正解率がわずかに上がるだけで、再現率は上昇するが適合率が大きく低下した。正解率が最も高いアルゴリズムは RF であり、その精度は、適合率 0.605、再現率 0.827、F-値 0.698、および正解率 0.970 であった。これは、発生頻度の少ない車線変更の適合率を上げるよりも、より多くの車線維持を検出したほうが正解率が高くなるためである。前述の通り、実現したいサービスの要求から、正解率より F-値が高く、再現率より適合率が高いことが求められる。よって、正解率をある程度犠牲にしても、F-値を上げることが望ましい。

車線変更の窓数に対する車線維持の窓数の比率  $r$  は、F-値最大の場合が正解値最大の場合と比べて、相対的に大きかった (それぞれ  $r = 4, 2$ )。すなわち、車線変更の検出精度を上げるには、車線維持の窓数を減らしすぎはいけないことが分かる。実際、図 6 に示すように F-値最大の実験条件において  $r$  のみ変化させた場合、 $r = 1$  より  $r = 4$  の方が F-値が高い。これは、図 3 から分かるように車線維持にもばらつきがあり、車線維持の窓数を減らすと必要な情報が失われることが原因と考えられる。更なる精度向上には、車線変更のオーバーサンプリングも検討する必要がある。

以上の結果から、本論文では、アンサンブル学習やインスタンスに基づく学習の精度が高く、他のアルゴリズムより車線変更検出に適していると結論づけた。特に、計算コス

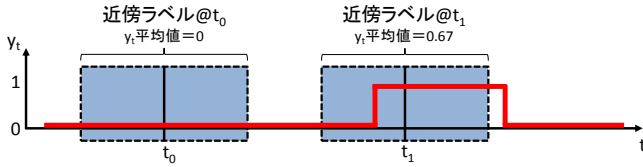


図 7 近傍ラベルの平均値

トをふまえた実用性の観点からは、**RF** および **EC+kNN** が有望である \*8。

### 3.2 車線変更の誤検出要因

前節の実験結果から、アンサンブル学習およびインスタンスに基づく学習の精度が高いことが示されたが、適合率は最大でも 0.733 であり、改善の余地がある。本節では、車線変更の誤検出要因を考える。

誤検出要因の 1 つとして、車線維持と車線変更の境界付近での中間的な車両挙動が考えられる。中間的な車両挙動が学習結果においても分類境界付近に存在し、誤検出してしまっている可能性がある。

この仮説を検証するため、近傍ラベルの平均値と検出結果の関係を調べた。近傍ラベルとは、図 7 に示すように、ある時刻  $t_0$  を中心とする 3.4 秒間 \*9 の時間に含まれる生データのラベル集合  $L_{t_0} = \{y_{t_0-17}, \dots, y_{t_0}, \dots, y_{t_0+17}\}$  を表す。近傍ラベルの平均値  $E(L_{t_0})$  は

$$E(L_{t_0}) = \frac{\sum_{k=-17}^{17} y_{t_0+k}}{34+1}$$

となる。  $y_t \in \{0, 1\}$  なので、  $E(L_{t_0})$  は  $[0, 1]$  の値を取る。

$E(L_{t_0})$  は時刻  $t_0$  近傍における車線変更の割合を表す。  $E(L_{t_0}) \simeq 0$  であれば概ね車線を維持し、  $E(L_{t_0}) \simeq 1$  であれば概ね車線を変更している。仮に中間的な車両挙動が誤検出の要因であれば、偽陽性と偽陰性における  $E(L_{t_0})$  は、それぞれ 0 および 1 に近い値にはならないと予想される。以上のように、検出結果と  $E(L_{t_0})$  の関係を見ることで、仮説が検証できると考えた。

検出結果と近傍ラベルの平均値  $E(L_{t_0})$  の関係を、図 8 に示す。ここでは、F-値が平均的に最大となる実験条件における **RF** の結果を用いた。真陽性、偽陰性、偽陽性および真陰性に対して、  $E(L_{t_0})$  の分布を箱ひげ図として表した。箱内の線、箱、ひげ、および+記号は、それぞれ中央値、第 1, 3 四分位の範囲、正常値の範囲、およびはずれ値を表す。

真陰性の  $E(L_{t_0})$  は 0 付近に分布しているのに対し、偽陽性のそれは必ずしも 0 に近い値を取っていない。このことから、偽陽性は中間的な車両挙動が要因の 1 つであるこ

\*8 各アルゴリズムの計算コストについては第 3.3 節で議論する。

\*9 1 回の車線変更にかかる平均時間を用いた。

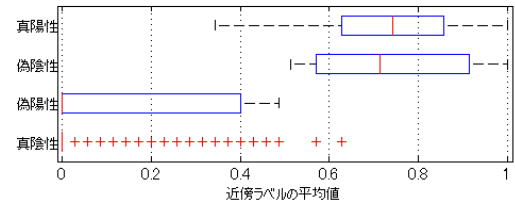


図 8 検出結果と近傍ラベルの平均値の関係

表 7 各アルゴリズムの計算量

アルゴリズム	訓練	分類
<b>RF</b>	$\mathcal{O}(MN_{tr} \log^2 N_{tr})$	$\mathcal{O}(MN_{te} \log N_{tr})$
<b>LB</b>	$\mathcal{O}(MN_{tr})$	$\mathcal{O}(MN_{te})$
<b>EC+kNN</b>	$\mathcal{O}(1)$	$\mathcal{O}(MN_{tr}N_{te})$
<b>DTW+kNN</b>	$\mathcal{O}(1)$	$\mathcal{O}(M^2N_{tr}N_{te})$
<b>SVM(RBF)</b>	$\mathcal{O}(M^2N_{tr})$	$\mathcal{O}(MN_{te})$
<b>DTree</b>	$\mathcal{O}(MN_{tr} \log N_{tr})$	$\mathcal{O}(MN_{te} \log N_{tr})$

とが分かる。一方で、偽陰性の  $E(L_{t_0})$  は真陽性のそれとあまり違いがない。このことから、偽陰性の要因は、中間的な車両挙動によるものではないと言える。

偽陽性が車線変更前後の中間的な車両挙動により発生しているということは、偽陽性については、実用上問題とならない可能性がある。他方、偽陰性については、今後改善の余地がある。

本節では、車線変更検出の更なる精度向上のため、車線変更の誤検出要因を調べた。

### 3.3 計算コスト

実験に使用したアルゴリズムによる訓練および分類にかかる計算量の平均を表 7 に示す。  $M$  を特徴ベクトルの長さ、  $N_{tr}$  を訓練用データのサンプル数、  $N_{te}$  を分類対象データのサンプル数とする。

**RF** は、**SVM** と比べ、訓練時の特徴ベクトルの長さに対する計算量が小さい。このことは、多変量時系列を扱う際には重要である。**EC+kNN** は訓練時には計算が不要な代わりに、分類時の計算量が大きく、このままでは実用的ではない。訓練時に訓練用データの索引を構成しておき、分類処理を高速化する手法も提案されている [1]。適切な最適化を行うことにより、**RF** や **EC+kNN** は実用的な速度が得られると考える。

## 4. アプリケーション

本節では、車線変更の検出を用いた具体的なアプリケーションとして、車線変更が発生しやすい場所をドライバに事前に通知するサービスを考える。車線変更はドライバに負担のかかりやすい運転行動の 1 つである。車線変更の発生しやすい場所を知らせることで、ドライバが余裕をもって車線を変更できるようになると考える。

同じ場所を走行した多数のデータから車線変更を検出

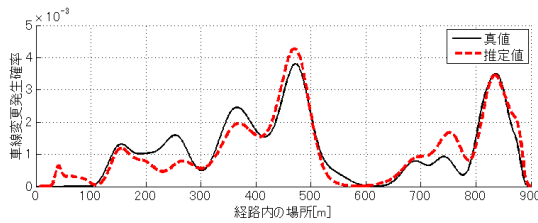


図 9 車線変更の確率分布

表 8 確率分布の推定誤差；小さいほどよい

アルゴリズム	2乗平均平方根誤差	最大誤差
RF	3.737e-04	1.693e-03
DTree	3.823e-04	1.856e-03
EC+kNN	4.107e-04	2.120e-03
DTW+kNN	4.107e-04	2.120e-03
SVM	4.181e-04	1.425e-03
LB	4.226e-04	1.776e-03

し、場所ごとの車線変更発生の確率分布を事前に推定すれば、車線変更の発生確率が高い場所を見つけられる。

車線変更の検出結果から推定した車線変更の確率分布の例を、図 9 に示す<sup>\*10</sup>。横軸に経路内の場所（経路起点からの距離）を示し、縦軸に車線変更の発生確率を示す。推定した確率分布が、車線変更のラベルから求めた真値に近いほど、推定精度が高いと言える。この例では、推定値は概ね真値に近い値を取っているが、一部の場所で誤差が大きいことが分かる。

確率分布の推定における 2 乗平均平方根誤差（5 分割交差検定の平均値）と最大誤差（同最大値）を表 8 に示す。車線変更検出により推定された車線変更の確率分布の誤差においては、RF が他のアルゴリズムより優れていると言える。インスタンスに基づく学習（EC+kNN および DTW+kNN）は、誤差が相対的に大きく、特定の場所での精度が低い可能性がある。

## 5. むすび

本論文では、速度や操舵角など安価な自転車センサ情報のみを用いた車線変更検出手法について述べた。提案手法では、複数のセンサ情報を組合せ、データを数秒間の窓に分割し、アンサンブル学習やインスタンスに基づく学習により車線変更を検出した。その結果、一般道実走行データを用いた実験において、他のアルゴリズムより高い検出精度が得られることを示した。

## 参考文献

[1] Bentley, J. L.: Multidimensional binary search trees used for associative searching, *Communications of the ACM*, Vol. 18, No. 9, pp. 509–517 (1975).

\*10 ここでの真値と推定値は、F-値最大の実験条件における RF の結果のうち、2 乗平均平方根誤差が中央値を取る分割の組合せから求めた。

[2] Breiman, L.: Random forests, *Machine learning*, Vol. 45, No. 1, pp. 5–32 (2001).

[3] Caruana, R. and Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms, *Proceedings of the 23rd international conference on Machine learning*, ACM, pp. 161–168 (2006).

[4] Chawla, N. V., Japkowicz, N. and Kotcz, A.: Editorial: special issue on learning from imbalanced data sets, *ACM Sigkdd Explorations Newsletter*, Vol. 6, No. 1, pp. 1–6 (2004).

[5] Friedman, J., Hastie, T., Tibshirani, R. et al.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), *The annals of statistics*, Vol. 28, No. 2, pp. 337–407 (2000).

[6] Kuge, N., Yamamura, T., Shimoyama, O. and Liu, A.: A driver behavior recognition method based on a driver model framework, Technical report, SAE Technical Paper (2000).

[7] Kumar, P., Perrollaz, M., Lefevre, S. and Laugier, C.: Learning-based approach for online lane change intention prediction, *Intelligent Vehicles Symposium (IV), 2013 IEEE*, IEEE, pp. 797–802 (2013).

[8] Mandalia, H. M. and Salvucci, M. D. D.: Using support vector machines for lane-change detection, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 49, No. 22, SAGE Publications, pp. 1965–1969 (2005).

[9] McCall, J. C., Wipf, D. P., Trivedi, M. M. and Rao, B. D.: Lane change intent analysis using robust operators and sparse bayesian learning, *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 8, No. 3, pp. 431–440 (2007).

[10] Morris, B., Doshi, A. and Trivedi, M.: Lane change intent prediction for driver assistance: On-road design and evaluation, *Intelligent Vehicles Symposium (IV), 2011 IEEE*, IEEE, pp. 895–901 (2011).

[11] Müller, M.: Dynamic time warping, *Information retrieval for music and motion*, pp. 69–84 (2007).

[12] Salvucci, D. D.: Inferring driver intent: A case study in lane-change detection, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 48, No. 19, SAGE Publications, pp. 2228–2231 (2004).

[13] Sannomiya, C., Kurihara, K., Tanaka, Y., Kashiwakura, T. and Nawa, K.: Analysis of temporal vehicle behaviors using bag of dynamical systems, *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, IEEE, pp. 1396–1401 (2014).

[14] Xi, X., Keogh, E., Shelton, C., Wei, L. and Ratanamahatana, C. A.: Fast time series classification using numerosity reduction, *Proceedings of the 23rd international conference on Machine learning*, ACM, pp. 1033–1040 (2006).

[15] Xing, Z., Pei, J. and Keogh, E.: A brief survey on sequence classification, *ACM SIGKDD Explorations Newsletter*, Vol. 12, No. 1, pp. 40–48 (2010).

[16] 谷口忠大, 山下元気, 長坂翔吾, 坂東誉司, 竹中一仁, 人見謙太郎: 運転時系列のベイズ二重分節解析によるチャック抽出, 計測自動制御学会論文集, Vol. 49, No. 11, pp. 1047–1056 (2013).

[17] 本田崇人, 松原靖子, 根山亮, 櫻井保志: 車両走行センサデータを用いた自動パターン検出, *WebDB Forum* (2015). 掲載予定.