自己回帰入出力隠れマルコフモデルを用いた運転行動のモデル化

Driving Behavior Modeling Using Autoregressive Input-Output Hidden Markov Models

○正　赤井 直紀 (名古屋大)　平山 高嗣 (名古屋大)　モラレス ルイス 洋一 (名古屋大)
赤木 康宏 (名古屋大)　劉 海龍 (名古屋大)　村瀬 洋 (名古屋大)

Naoki AKAI, Nagoya University, akai@coi.nagoya-u.ac.jp
Takatsugu HIRAYAMA, Nagoya University　Luis Yoichi MORALES, Nagoya University
Yasuhiro AKAGI, Nagoya University　Hailong LIU, Nagoya University
Hiroshi MURASE, Nagoya University

This paper presents a driving behavior modeling method using autoregressive input-output hidden Markov models (AIOHMMs). First, model parameter learning and discrimination ways using the AIOHMMs are detailed. In experiments, we model four driving maneuvers with driver's eye-gaze and ego-vehicle localization information and compare maneuver discrimination performances by four types of HMMs. Additionally, we discuss modeling performance by the AIOHMMs through the experiments.

Key Words: Driving Behavior Modeling, Autoregressive Input-Output Hidden Markov Models

1　はじめに

本稿では，自己回帰入出力隠れマルコフモデル (autoregressive input-output hidden Markov models: AIOHMMs) [1] を用いた運転行動のモデル化について述べる．まず，関連する研究を簡潔にまとめ，次に AIOHMMs による運転行動のモデル化に関する定式化について述べる．その後，実際の運転行動データを用いたモデル化を行い，そのモデルの性能に関して考察する．

2　関連研究

隠れマルコフモデル (HMMs) [2] は時系列を有する現象をモデル化するために広く利用されている．HMMs とは，可観測の出力が離散的な潜在変数に依存すると仮定し，その潜在変数が時系列的に依存するモデルである．しかしオリジナルの HMMs には現象を説明するための能力に欠如する場合があり，その拡張法が提案されてきた．自己回帰 HMMs (AHMMs) [3] は，現在の出力が過去の出力に依存するような現象をモデル化できる．入出力 HMMs (IOHMMs) [4] は，出力と潜在変数が外部からの入力に依存するような現象をモデル化できる．これらは 1990 年代に提案された方法であるが，2015 年に，Jain らによって AIOHMMs が提案された [1]．AIOHMMs は，AHMMs と IOHMMs を組み合わせたものと解釈できる．

運転行動のモデル化に関しては，これまでにも多くの研究例が報告されている [5]．特に近年では，LSTM の様な深層学習を用いたモデル化 [6] も提案されており，モデル化の精度は非常に向上している．一方で AIOHMMs は，その直感的なグラフィカルモデルの解釈のし易さから，モデル化された事象の解釈性の高さという観点から有用性があると考えられる．本稿では，AIOHMMs を用いたモデル化の結果から，運転行動モデルに関する解釈も試みる．また，AIOHMMs を用いたモデル化に関しては，すでに文献 [1, 7] で述べられているが，本稿では実装方法がわずかに異なる方法に関して述べる．これは，潜在変数の入力の条件付き確率分布を，より柔軟に表現する効果を持つ．

3　AIOHMMs を用いたモデル化

図 1 に，AIOHMMs のグラフィカルモデルを示す．白，灰色のノードはそれぞれ潜在，可観測変数を表す．以下では，本モデルによる学習と予測のプロセスに関して述べる．

3.1　学習

3.1.1　EM アルゴリズムによる最適化

学習時には，N 個のシーケンスからなる訓練データ $\mathcal{D} = \{(\mathbf{x}^{(n)}_{1:T_n}, \mathbf{y}^{(n)}_{1:T_n}) | n = 1, ..., N\}$ を持つものとする．1 つの訓練データは，1 から T_n のシーケンスを持ち，入力変数 $\mathbf{x}^{(n)}_{1:T_n}$ ($\mathbf{x}^{(n)}_t \in \mathcal{R}^M$) と出力変数 $\mathbf{y}^{(n)}_{1:T_n}$ ($\mathbf{y}^{(n)}_t \in \mathcal{R}^L$) を有する．潜在変数 $\mathbf{z}^{(n)}_{1:T_n}$ ($\mathbf{z}^{(n)}_t \in \mathcal{R}^S$) は，以下の条件を持つ: $z^{(n)}_{i,t} \in \{0, 1\}$, $\sum_{i \in S} z^{(n)}_{i,t} = 1$．学習における目的は，以下に示す対数尤度を最大化するハイパーパラメータ $\mathbf{\Theta}$ を求めることである．

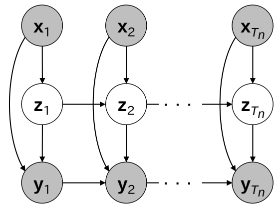

Fig.1 The graphical model of AIOHMMs.

$$l(\mathbf{\Theta}; \mathcal{D}) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} \ln p(\mathbf{y}^{(n)}_t | \mathbf{x}^{(n)}_t; \mathbf{\Theta}) \tag{1}$$

しかしながら，潜在変数が存在するため，この対数尤度を直接最大化することは困難である．そのため，EM アルゴリズムを用いてこの対数尤度を最大化する．

EM アルゴリズムにおける E ステップでは，完全データに対する対数尤度の期待値 $Q(\mathbf{\Theta}; \hat{\mathbf{\Theta}})$ を求める．

$$Q(\mathbf{\Theta}; \hat{\mathbf{\Theta}}) = E[l_c(\mathbf{\Theta}; \mathcal{D}_c) | \hat{\mathbf{\Theta}}, \mathcal{D}] \tag{2}$$

ここで $\hat{\mathbf{\Theta}}$ は現在のハイパーパラメータ，l_c は完全データに対する対数尤度，$\mathcal{D}_c = \{(\mathbf{x}^{(n)}_{1:T_n}, \mathbf{y}^{(n)}_{1:T_n}, \mathbf{z}^{(n)}_{1:T_n}) | n = 1, ..., N\}$ は完全データである．$\mathcal{D}_c$ は潜在変数もデータとして含むため，完全データと呼ばれる．l_c は以下の様に定義される．

$$l_c(\mathbf{\Theta}; \mathcal{D}_c) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} \ln p(\mathbf{y}^{(n)}_t, \mathbf{z}^{(n)}_t | \mathbf{x}^{(n)}_t; \mathbf{\Theta}) \tag{3}$$

EM アルゴリズムにおける M ステップでは，上述の期待値を最大化するハイパーパラメータを求める．

$$\mathbf{\Theta} = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta}; \hat{\mathbf{\Theta}}) \tag{4}$$

おおよそのパラメータは，その導関数を 0 にする値を求めることで最大化できるが，いくつかのパラメータはある種の制約を有する．そのような場合は，ラグランジュの未定乗数法を用いて最適化を行う．なお M ステップにおいて $\hat{\mathbf{\Theta}}$ は，固定値であるとみなされることに留意されたい．

AIOHMMs を用いてモデル化をするために，本稿では以下のハイパーパラメータを使用する $\mathbf{\Theta} = \{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i, \mathbf{a}_{i,1}, \mathbf{a}_{i,}, \mathbf{b}_i, \boldsymbol{\pi}_l, \mathbf{W}_l\}$ ($i = 1, ..., S, l = 1, ..., L$)．以下では，これらのパラメータを説明する．

3.1.2 出力確率

まず，出力の確率はガウス分布に従うと仮定する．AIOHMMs では，出力が過去の出力と現在の入力に依存するため，この関係を線形ガウスモデルにより表現する．

$$p(\mathbf{y}_t^{(n)}|\mathbf{x}_t^{(n)},\mathbf{y}_{t-1}^{(n)},\mathbf{z}_t^{(n)}) = \prod_{i=1}^{S}\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i)^{z_{i,t}^{(n)}}$$
$$\boldsymbol{\mu}_{i,t}^{(n)} = \begin{cases}(1+\mathbf{a}_{i,1}^T\mathbf{x}_t^{(n)})\boldsymbol{\mu}_i & (t=1)\\ (1+\mathbf{a}_i^T\mathbf{x}^{(n)}+\mathbf{b}_i^T\mathbf{y}_{t-1}^{(n)})\boldsymbol{\mu}_i & (t\geq 2)\end{cases} \tag{5}$$

ここで $\mathbf{a}_{i,1}$，$\mathbf{a}_i$，$\mathbf{b}_i$ は，i 番目の潜在変数の状態に関連する係数ベクトルである．なお，文献 [1, 7] では $\mathbf{a}_{i,1}$ が利用されていないが，時刻 1 の出力変数は他の出力変数と比べて依存数が少ないため，$\mathbf{a}_{i,1}$ は別途設けている．

3.1.3 初期確率

HMMs において時刻 1 以外の潜在変数は，1 時刻前の潜在変数に依存する．この潜在変数間の遷移は，遷移行列を用いることでモデル化できる．これに関しては次項にて述べる．入力変数が存在しない場合には，時刻 1 における潜在変数に関する確率分布，すなわち初期確率は，以下の様に表される．

$$p(\mathbf{z}_1^{(n)}) = \prod_{i=1}^{S}\pi_i^{z_{i,1}^{(n)}} \tag{6}$$

$\boldsymbol{\pi}\in\mathcal{R}^S$ は以下の条件を持つ: $(\boldsymbol{\pi})_i=\pi_i\geq 0$，$\sum_{i=1}^{S}\pi_i=1$．

入力変数を有する場合は，初期確率は入力変数に依存する．ここで我々は，入力変数のセットが L 個のクラスタに分類されるものと仮定する．実装では，入力変数セットに対して事前に K-means クラスタリングを適用する．ここで新たな変数 $c_t^{(n)}\in\{1,...,L\}$ を導入し，これは入力変数が何番目のクラスタに属するかを表すものとする．表記の簡便化のため，$c_t^{(n)}$ を添字として利用する場合には c と記述する．これにより，AIOHMMs における初期確率は以下の様に表される．

$$p(\mathbf{z}_1^{(n)}|\mathbf{x}_1^{(n)}) = \prod_{i=1}^{S}\pi_{ic}^{z_{i,1}^{(n)}} \tag{7}$$

なお，$(\boldsymbol{\pi}_c)_i=\pi_{ic}$ である．これは，初期確率が最初の入力変数により決まることを意味する．なお，文献 [1, 7] では初期確率に関する議論は行われていない．

3.1.4 遷移行列

入力変数を有しない HMMs の場合，潜在変数間の遷移は，$S\times S$ の行列により表現される．

$$p(\mathbf{z}_t^{(n)}|\mathbf{z}_{t-1}^{(n)}) = \prod_{i=1}^{S}\prod_{j=1}^{S}w_{ij}^{z_{i,t}^{(n)}z_{j,t-1}^{(n)}} \tag{8}$$

なお，$p(z_t^{(n)}=j|z_{t-1}^{(n)}=i)=(\mathbf{W})_{ij}=w_{ij}\geq 0$ である．また $\mathbf{W}$ は遷移を表現する確率であるため，以下の条件を満たす．

$$\sum_{j=1}^{S}w_{ij}=1 \tag{9}$$

M ステップにおいてこの条件は，ラグランジュの未定乗数法を用いて最適化することで満たされる．

前項で述べた通り，我々は入力変数が L 個のクラスタに分類されると仮定している．そのため AIOHMMs における遷移行列は以下の様に表される．

$$p(\mathbf{z}_t^{(n)}|\mathbf{z}_{t-1}^{(n)},\mathbf{x}_t^{(n)}) = \prod_{i=1}^{S}\prod_{j=1}^{S}w_{ijc}^{z_{i,t}^{(n)}z_{j,t-1}^{(n)}} \tag{10}$$

ここで $p(z_t^{(n)}=j|z_{t-1}^{(n)}=i,\mathbf{x}_t^{(n)})=(\mathbf{W}_c)_{ij}=w_{ijc}$ である．

文献 [1, 7] では，遷移行列は以下の様に表される．

$$p(z_t^{(n)}=j|z_{t-1}^{(n)}=i,\mathbf{x}_t^{(n)}) = \frac{\exp\left(\mathbf{w}_{ij}^T\mathbf{x}_t^{(n)}\right)}{\sum_{l=1}^{S}\exp\left(\mathbf{w}_{il}^T\mathbf{x}_t^{(n)}\right)} \tag{11}$$

ここで $\mathbf{w}_{ij}\in\mathcal{R}^M$ はハイパーパラメータである．初期確率や遷移行列の各要素は，最適化を行うと 0 か 1 に収束しやすい．そのため，柔軟な遷移を表現するために，入力変数に依存する複数の確率を導入することは効果的であると考えれる．

3.1.5 E ステップ

式 (5)，(7)，および (10) を用いることで，式 (3) は以下の様に書き変えられる．

$$\begin{aligned}l_c(\boldsymbol{\Theta};\mathcal{D}_c) &= \sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{i=1}^{S}z_{i,t}^{(n)}\ln\pi_{ic}\\ &+\sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{i=1}^{S}z_{i,t}^{(n)}\ln\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i)\\ &+\sum_{n=1}^{N}\sum_{t=2}^{T_n}\sum_{i=1}^{S}\sum_{j=1}^{S}z_{i,t}^{(n)}z_{j,t-1}^{(n)}\ln w_{ijc}\end{aligned} \tag{12}$$

ここで，$\mathbf{z}_t^{(n)}$，および $\mathbf{z}_t^{(n)}$ と $\mathbf{z}_{t-1}^{(n)}$ に関する条件付き確率分布を以下の様に定義する．

$$\gamma_{i,t}^{(n)} \stackrel{\mathrm{def}}{=} p(z_{i,t}^{(n)}=1|\mathbf{x}_{1:T_n}^{(n)},\mathbf{y}_{1:T_n}^{(n)}\boldsymbol{\Theta}), \tag{13}$$

$$\xi_{ij,t}^{(n)} \stackrel{\mathrm{def}}{=} p(z_{i,t}^{(n)}=1,z_{j,t-1}^{(n)}=1|\mathbf{x}_{1:T_n}^{(n)},\mathbf{y}_{1:T_n}^{(n)}\boldsymbol{\Theta}), \tag{14}$$

式 (13)，(14) を用いることで，期待値は以下の様に計算できる．

$$\begin{aligned}Q(\boldsymbol{\Theta};\hat{\boldsymbol{\Theta}}) &= \sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{i=1}^{S}\gamma_{i,t}^{(n)}\ln\pi_{ic}\\ &+\sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{i=1}^{S}\gamma_{i,t}^{(n)}\ln\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i)\\ &+\sum_{n=1}^{N}\sum_{t=2}^{T_n}\sum_{i=1}^{S}\sum_{j=1}^{S}\xi_{ij,t}^{(n)}\ln w_{ijc}\end{aligned} \tag{15}$$

$\gamma_{i,t}^{(n)}$ と $\xi_{ij,t}^{(n)}$ を効率的に計算するために，フォワード・バックワードアルゴリズムを用いることができる [8]．また，長い時系列データの計算を行う場合に，アンダーフローが発生し計算が不安定になる問題がある．この問題は，スケーリングアルゴリズムにより回避できる [8]．本稿では，$\gamma_{i,t}^{(n)}$ と $\xi_{ij,t}^{(n)}$ を計算するための最低限の計算過程を示し，詳細は文献 [8] に譲る．

$$\gamma_{i,t}^{(n)} = \hat{\alpha}_{i,t}^{(n)}\hat{\beta}_{i,t}^{(n)} \tag{16}$$

$$\xi_{ij,t}^{(n)} = \frac{\hat{\alpha}_{i,t}^{(n)}w_{ijc}\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i)\hat{\beta}_{j,t+1}^{(n)}}{s_{t+1}^{(n)}} \tag{17}$$

$$s_t^{(n)}\hat{\alpha}_{i,t}^{(n)} = \left(\sum_{j=1}^{S}\hat{\alpha}_{j,t-1}^{(n)}w_{ijc}\right)\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i) \tag{18}$$

$$s_t^{(n)} = \sum_{i=1}^{S}\left(\sum_{j=1}^{S}\hat{\alpha}_{j,t-1}^{(n)}w_{ijc}\right)\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i) \tag{19}$$

$$s_{t+1}^{(n)}\hat{\beta}_{i,t}^{(n)} = \sum_{j=1}^{S}w_{ijc}\mathcal{N}(\mathbf{y}_t^{(n)}|\boldsymbol{\mu}_{i,t}^{(n)},\boldsymbol{\Sigma}_i)\beta_{j,t+1}^{(n)} \tag{20}$$

$\hat{\alpha}_{i,t}^{(n)}$ と $\hat{\beta}_{i,t}^{(n)}$ は再帰的に計算することができ，その初期値は以下の様になる: $\hat{\alpha}_{i,1}^{(n)}=\pi_{ic}\mathcal{N}(\mathbf{y}_1^{(n)}|\boldsymbol{\mu}_{i,1}^{(n)},\boldsymbol{\Sigma}_i)$，$\hat{\beta}_{i,T_n}^{(n)}=1$．

3.1.6 M ステップ

M ステップでは，$\gamma_{i,t}^{(n)}$ と $\xi_{i,t}^{(n)}$ は固定値として扱われる．それぞれのハイパーパラメータによる導関数を求め，それを 0 とする方程式を解くことでパラメータを更新する．

$$\mathbf{a}_{i,1} = \left(\sum_{n=1}^{N} \gamma_{i,1}^{(n)} \mathbf{x}_1^{(n)} \mathbf{x}_1^{(n)T} \right)^{-1} \sum_{n=1}^{N} \gamma_{i,1}^{(n)} \left(\frac{\mathbf{x}_1^{(n)} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_1^{(n)}}{\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i} - \mathbf{x}_1^{(n)} \right) \tag{21}$$

$$\mathbf{a}_i = \left(\sum_{n=1}^{N} \sum_{t=2}^{T_n} \gamma_{i,t}^{(n)} \mathbf{x}_t^{(n)} \mathbf{x}_t^{(n)T} \right)^{-1} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \gamma_{i,t}^{(n)} \left(\frac{\mathbf{x}_t^{(n)} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_t^{(n)}}{\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i} - \mathbf{x}_t^{(n)} - \mathbf{b}_i^T \mathbf{y}_{t-1}^{(n)} \mathbf{x}_t^{(n)} \right) \tag{22}$$

$$\mathbf{b}_i = \left(\sum_{n=1}^{N} \sum_{t=2}^{T_n} \gamma_{i,t}^{(n)} \mathbf{y}_{t-1}^{(n)} \mathbf{y}_{t-1}^{(n)T} \right)^{-1} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \gamma_{i,t}^{(n)} \left(\frac{\mathbf{y}_{t-1}^{(n)} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_t^{(n)}}{\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i} - \mathbf{y}_{t-1}^{(n)} - \mathbf{a}_i^T \mathbf{x}_t^{(n)} \mathbf{y}_{t-1}^{(n)} \right) \tag{23}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} c_{i,t}^{(n)} \gamma_{i,t}^{(n)} \mathbf{y}_t^{(n)}}{\sum_{n=1}^{N} \sum_{t=1}^{T_n} c_{i,t}^{(n)2} \gamma_{i,t}^{(n)}} \tag{24}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{i,t}^{(n)} \mathbf{V}_{i,t}^{(n)}}{\sum_{n=1}^{N} \sum_{t=1}^{T_n} \gamma_{i,t}^{(n)}} \tag{25}$$

ここで $c_{i,1}^{(n)} = 1 + \mathbf{a}_{i,1}^T \mathbf{x}_1^{(n)}$，$c_{i,t}^{(n)} = 1 + \mathbf{a}_i^T \mathbf{x}_t^{(n)} + \mathbf{b}_i^T \mathbf{y}_{t-1}^{(n)}$ $(t \geq 2)$，$\mathbf{V}_{i,t}^{(n)} = (\mathbf{y}_t^{(n)} - c_{i,t}^{(n)} \boldsymbol{\mu}_i)(\mathbf{y}_t^{(n)} - c_{i,t}^{(n)} \boldsymbol{\mu}_i)^T$ である．なお，$N = 1$ の場合は逆行列が定義できなくなり，$\mathbf{a}_{i,1}$，$\mathbf{a}_i$，および $\mathbf{b}_i$ が更新できないことに留意されたい．

初期確率は以下の様に更新される．

$$\pi_{il} = \frac{\sum_{n=1}^{N} \mathbb{1}(c_t^{(n)} = l) \gamma_{i,1}^{(n)}}{\sum_{n=1}^{N} \sum_{j=1}^{S} \mathbb{1}(c_t^{(n)} = l) \gamma_{j,1}^{(n)}}, \tag{26}$$

ここで $\mathbb{1}(\cdot)$ はインディケータであり，括弧内の条件が真のとき 1，それ以外に 0 となる．

式 (9) に示すように，遷移行列は確率としての制約を持つ．ラグランジュの未定乗数法を用いて制約を満たしながら最適化を行うために，以下の関数を新たに定める．

$$J(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) \stackrel{\text{def}}{=} Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) + \sum_{j=1}^{S} \sum_{l=1}^{K} \lambda_{jl} \left(1 - \sum_{i=1}^{S} w_{ijl} \right) \tag{27}$$

ここで λ_{jl} はラグランジュ乗数である．この関数の w_{ijl} に関する導関数は以下の様になる．

$$\frac{\partial J(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}})}{\partial w_{ijl}} = \sum_{n=1}^{N} \sum_{t=2}^{T_n} \mathbb{1}(c_t^{(n)} = l) \xi_{ij,t}^{(n)} \frac{1}{w_{ijl}} - \lambda_{jl} \tag{28}$$

最終的に，w_{ijl} は以下の様に更新される．

$$w_{ijl} = \frac{\sum_{n=1}^{N} \sum_{t=2}^{T_n} \mathbb{1}(c_t^{(n)} = l) \xi_{ij,t}^{(n)}}{\sum_{i=1}^{S} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \mathbb{1}(c_t^{(n)} = l) \xi_{ij,t}^{(n)}} \tag{29}$$

なお，$\lambda_{jl} = \sum_{i=1}^{S} \sum_{n=1}^{N} \sum_{t=2}^{T_n} \mathbb{1}(c_t^{(n)} = l) \xi_{ij,t}^{(n)}$ は，$\sum_{i=1}^{S} w_{ijl} = 1$ の制約を用いることで求められる．

文献 [1, 7] では，式 (11) に示す $\mathbf{w}_{ij}$ は勾配法により最適化されると言及されている．しかしながら，その勾配計算方法に関しては言及されていない．

3.2 予測

学習した HMMs を用いることで，出力変数 $\mathbf{y}_{1:T_n}^{(n)}$ が得られる確率を予測することができる．これは，フォワードアルゴリズムを用いて以下の様に計算される [8].

$$p(\mathbf{y}_{1:T_n}^{(n)}) = \sum_{i=1}^{S} \alpha_{i,T_n}^{(n)}, \tag{30}$$

ここで $\alpha_{i,T_n}^{(n)}$ は，式 (18) から $s_t^{(n)}$ を取り除くことで計算できる．

4 運転行動モデリング

4.1 実験プラットフォーム

実験では，3D LiDAR (HDL-32e) を搭載した YAMAHA 製のゴルフカートを利用する．文献 [9] で述べられている位置推定法を利用し，高精度に車両の 3 次元位置を取得できる．また CAN と IMU から，車両の移動速度などの情報を得ることができる．さらにドライバには，視線計測装置 (Tobii Pro Glasses 2) を着用させ，視線情報も取得する．

4.2 入力変数

3D LiDAR と地図データを用いて，入力変数を作成する．まず 3D LiDAR を中心とした極座標系の一定範囲内を 25 等分し，一定高さに含まれる最小の計測値 25 個を記録する．さらに同様の範囲を推定位置を中心とした極座標系に設け，推定位置から地図に存在する障害物までの最小距離 25 個を記録する．なお，それぞれ値が含まれない場合には，-1 を代入する．この 50 個の要素を，1 つの入力変数として利用する．

4.3 出力変数

車両の移動情報，およびドライバの視線計測情報を用いて出力変数を作成する．移動情報は並進・角速度，視線情報は 3 次元のベクトルである．これらの値に対して約 0.5 秒間の，生，絶対値，差分値，差分値の絶対値の平均と分散を計算する．これら 40 個の要素を，1 つの出力変数として利用する．

4.4 次元削減

M ステップでは逆行列を求めるため，入出力変数の要素数が大きい場合には，計算の不安定さや計算速度の低下を引き起こす．またこれらの要素は不要な情報も含むと考えられる．そこで PCA を用いて，入出力変数の次元削減を行う．どちらの変数に対しても，寄与率が 90 % を越えるようにしながら，入力変数を 30，出力変数を 3 次元にまで圧縮する．

4.5 ハイパーパラメータの初期値

出力確率に関するパラメータ $\boldsymbol{\mu}_i$，$\boldsymbol{\Sigma}_i$ の初期値は，出力変数セットに対する K-means クラスタリングの結果を用いる．なお，クラスタ数と AIOHMMs の潜在変数の状態数は等しいものとする．線形ガウスモデルを計算するための係数 $\mathbf{a}_{i,1}$，$\mathbf{a}_i$，および $\mathbf{b}_i$ はすべて $\mathbf{0}$ とする．初期確率 $\boldsymbol{\pi}_l$，および遷移行列 $\mathbf{W}_l$ は，初期段階ではすべて一様確率とする．

5 実験

5.1 運転データ

大学構内で 6 名分の運転データを集めた．大学構内には複数のレーンがないため，レーンチェンジの操作は見られなかった．一方で歩行者や自転車，他車などの交通参加者が前方に存在し，速度を低下するような操作が見られた．この様な操作は，まとめて追従と区分する．本実験では，直進，左折，右折，追従の 4 操作を扱う．合計で約 4 時間分のデータを収集し，直進 192，左折 72，右折 87，追従 43 のセットを取得した．これら約半分のデータを学習に利用し，残りの半数で識別の性能を検証する．

5.2 学習結果

本実験では，潜在変数の状態数 S は 5 と固定し，HMMs，AHMMs，IOHMMs，および AIOHMMs によるモデル化の比較を行った．図 2 には，各 HMMs によるそれぞれの操作に対する学習過程における対数尤度の変化を示す．学習初期のパラメータは固定であるが，それぞれの HMMs による対数尤度の向上の仕方が異なることが確認できる．

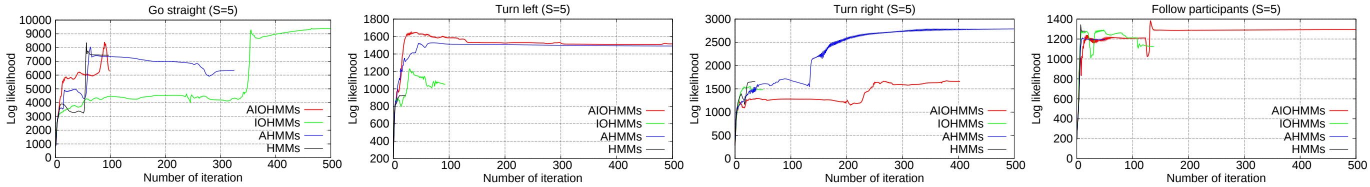

Fig.2 Log likelihoods in the learning phase.

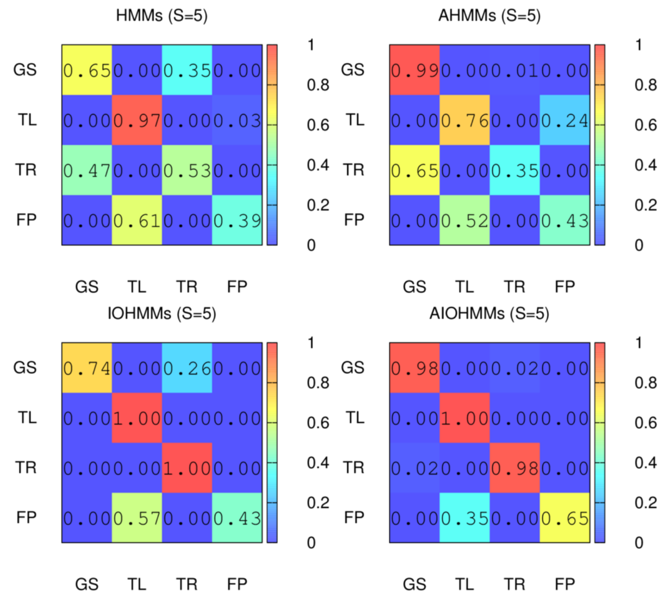

Fig.3 Confusion matrices by each HMM. GS, TL, TR, and FP denote maneuvers of go straight, turn left, turn right, and follow participants, respectively.

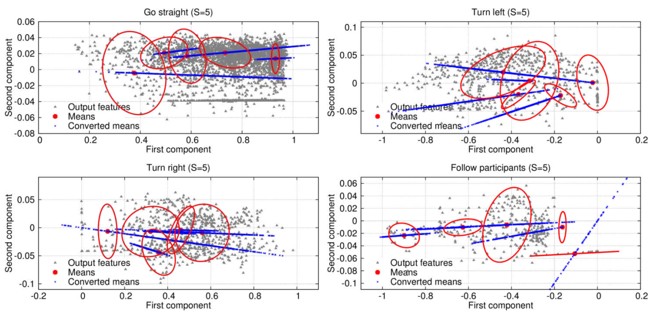

Fig.4 Fitting results by the AIOHMMs.

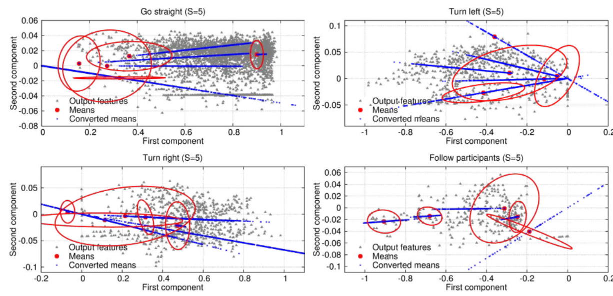

Fig.5 Fitting results by the AHMMs.

5.3　識別結果とモデル化に対する考察

図 3 に，それぞれの HMMs による操作の識別結果を示す．AIOHMMs による識別性能が高く，AHMMs と HMMs による識別性能が低いという結果となった．HMMs によるモデル化の場合，線形ガウスモデルを用いていないため，他の HMMs と比較して識別性能が低下したと考えられる．

図 4 と 5 は，AIOHMMs と AHMMs による出力変数の適合結果を示す．本図において，青点は線形変換されたガウス分布の平均，赤線はガウス分布の分散を示している．両者は同じく線形ガウスモデルを利用しているが，その適合結果は異なる結果となっている．AIOHMMs による適合結果では，ガウス分布の平均がデータ点の内部に存在し，かつ線形変換によりデータの分布を捉えている傾向が見てとれる．一方で AHMMs による適合結果では，ガウス分布の平均がデータ点の外部に存在し，線形変換により強引にデータの分布を捉えているように見える．

これらの結果から，図 4 に示す AIOHMMs によるモデル化結果おいては，各潜在変数の状態が分割された運転行動の中心的なデータを捉えているものと考えられる．そして，過去の運転行動や外界の情報が適切に運転行動の変化傾向を説明するパラメータが学習されているものと考えられる．

6　おわりに

本稿では，AIOHMMs を用いた運転行動のモデリングに関して述べた．実験から，AIOHMMs によるモデル化は，運転行動の中心的なデータを捉え，かつ過去や外界の情報を適切に利用したモデル化を実現する可能性を示した．今後は，AIOHMMs により学習された結果の解析を行っていく予定である．

参考文献

[1] A. Jain et al. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. *CoRR*, abs/1504.02789, 2015.

[2] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. IEEE*, volume 77, pages 257–286, 1989.

[3] B.-H. Juang et al. Mixture autoregressive hidden Markov models for speech signals. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 33:1404–1413, 1985.

[4] Y. Bengio et al. An input output HMM architecture. In *Proc. NIPS*, pages 427–434, 1994.

[5] H. Berndt et al. Continuous driver intention recognition with hidden Markov models. In *Proc. IEEE ITSC*, pages 1189–1194, Oct 2008.

[6] A. Jain et al. Brain4Cars: Car that knows before you do via sensory-fusion deep learning architecture. *CoRR*, abs/1601.00740, 2016.

[7] E. L. Zec et al. Statistical sensor modelling for autonomous driving using autoregressive input-output HMMs. In *Proc. IEEE ITSC*, pages 1331–1336, 2018.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag, Berlin, Heidelberg, 2006.

[9] N. Akai et al. Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning. In *Proc. IEEE ITSC*, pages 1147–1152, 2017.