

平成 26 年 名古屋大学 学位論文



データと事象駆動型システムモデルに 基づいた診断と予測

指導教官 鈴木 達也 教授
 稲垣 伸吉 講師
 田崎 勇一 助教

名古屋大学 大学院 工学研究科
機械理工学専攻 電子機械工学分野
メカトロニクス設計講座 モビリティシステムグループ

山口 拓真

Copyright © TAKUMA YAMAGUCHI

Output Data : 2014 年 1 月 16 日

目次

第 1 章	序論	1
1.1	事象駆動型システムの診断と予測	2
1.1.1	事象駆動型システムの数理モデル	2
	(A) 事象駆動システムとそのモデルの分類	2
	(B) マルコフ過程	4
	(C) 時間を含めたマルコフ過程	5
	(D) 事象駆動型システムに対する準マルコフ過程	6
1.2	診断や予測問題における従来手法	7
1.2.1	診断問題	7
1.2.2	予測問題	9
1.3	データ活用の必要性	11
1.4	研究目的	12
1.5	本論文の構成	12
第 2 章	Bayesian Network を用いた分散型故障診断法	15
2.1	はじめに	15
2.1.1	ベイジアンネットワークの学習手法	16
2.2	故障診断問題	17
2.2.1	Bayesian Network 概説	18
2.2.2	Bayesian Network による因果関係の表現と確率推論法	20
2.2.3	故障診断手順および故障診断法	22
2.3	準マルコフ過程を用いた故障診断	23
2.3.1	確率密度関数の定義	24
2.3.2	確率密度関数の推定	25
2.3.3	AIC に基づくヒストグラムの階級幅決定法	25

2.3.4	ゼロ頻度問題への対策	27
2.3.5	推定すべき確率密度関数の数	29
2.3.6	局所的故障診断方法	30
2.4	ベジアンネットワークを用いた大域的診断器	31
2.4.1	ベジアンネットワークの条件付き確率計算	32
2.5	ネットワーク構造学習	33
2.5.1	KL 情報量を用いたネットワーク学習法	34
2.5.2	Kullback-Leibler 情報量	34
2.5.3	アークの重要度評価法	34
2.6	検証実験	36
2.6.1	自動搬送ライン	36
2.6.2	故障候補	38
2.6.3	グラフ構造	38
2.6.4	診断結果	40
2.6.5	閾値による差異	42
2.7	本章のまとめ	43
第 3 章	プロセスの順序性に着目した故障診断	45
3.1	はじめに	45
3.2	問題設定	46
3.2.1	診断対象のシステム	46
3.2.2	故障診断問題の定式化	47
3.2.3	各トークンにおける時間付きイベント	48
3.2.4	各変数間の関係	49
3.3	システムの故障診断	51
3.3.1	トークンに着目した故障確率の計算	51
3.3.2	観測に基づくトークンイベント系列の生起確率計算	51
3.3.3	故障確率の計算	52
3.4	時間付きマルコフモデル	52
3.4.1	制御指令系列による影響	54
	一時停止制御	54
	速度変化動作	54

3.4.2	トークンイベント系列の生起確率の計算	56
3.5	TTP の学習方法	57
3.6	実験環境	57
3.6.1	自動搬送ライン	57
	故障状況	58
3.6.2	実験結果	59
3.7	本章のまとめ	62
第 4 章	動的計画法を用いた車の走行予測	63
4.1	はじめに	63
4.2	PDTT と SDTT の表現	63
4.3	PDTT の最尤推定問題	65
4.3.1	最尤推定問題の定式化	65
4.3.2	準マルコフ過程を用いた PDTT のモデル化	65
4.3.3	マルコフモデルの状態遷移確率	67
4.3.4	動的計画法に基づく最尤推定	69
4.4	実験	70
4.4.1	実験条件	70
4.4.2	実験結果	70
4.5	本章のまとめ	73
第 5 章	結論	77
5.1	本研究の成果	77
5.1.1	確率理論を用いた事象駆動型システムの状況の推定	77
5.1.2	準マルコフ過程を用いた予測器の設計	78
5.1.3	観測データに基づく分散型手法の適用	78
5.2	今後の課題	79

謝辞

参考文献

目次

1.1	An example of discrete event system	2
1.2	Category of event driven system	3
1.3	An example of discrete time Markov model	6
1.4	An example of continue time Markov model	6
1.5	An example of semi Markov model	7
1.6	Configuration of smart grid, HEMS and electric vehicle	10
1.7	The structure of this paper	13
2.1	Block diagram of relationship between event-driven system and de- centralized fault diagnoser	17
2.2	Observation in subsystem k	18
2.3	Example of Bayesian Network (Burglar Alarm)	19
2.4	Example of Bayesian Network (Proposed method)	21
2.5	Diagram of decentralized fault diagnosis procedure	23
2.6	The example of timed transition probability	24
2.7	Procedure of estimating a probability density function	25
2.8	Example before integrating histogram	27
2.9	Example after integrating histogram	27
2.10	Example of discounting result (Laplas's law) before discounting	29
2.11	Example of discounting result (Laplas's law) after discounting	29
2.12	Bipartite Bayesian Network for fault diagnosis	31
2.13	Example of Bayesian Network	31
2.14	Bayesian Network Example 1	33
2.15	Bayesian Network Example 2	33
2.16	Example of the difference between TTPs	35

2.17	Diagram of the transfer line and definition of events	37
2.18	Simulator of automatic transfer line	37
2.19	Bayesian Network structures	39
2.20	Relation between threshold value and the number of arcs	39
2.21	Success rate and the number of arcs	41
2.22	The number of TTPs and arcs	42
3.1	Simulated transfer line	46
3.2	Observation of timed events	47
3.3	Diagnoser and event-driven system	48
3.4	The event observation of parallel processes	49
3.5	Relation between observations and faults: Grey nodes mean observ- able ones, white nodes non-observable ones	50
3.6	The example of Left-to-Right TMM representing a process	53
3.7	A example of $a_{v_{n_i}^i v_{n_{i+1}}^i}^r(\tau)$	55
3.8	$f_{v_{n_i}^i v_{n_{i+1}}^i}^r(\tau, \mathbf{c})$ after a stop action control	55
3.9	$f_{v_{n_i}^i v_{n_{i+1}}^i}^r(\tau, \mathbf{c})$ after a speed change control	55
3.10	Diagram of simulated transfer line	57
3.11	The relation between average success rate and learning data	61
4.1	Profile of departure and travel time (PDTT) over 1 day from the present time t	64
4.2	PDTT represented by the state of the remaining travel time to return home	66
4.3	Left-to-Right Markov model of a PDTT	66
4.4	Initial state probabilities based on the observation	68
4.5	SDTT of the examinee for 188 days	70
4.6	Typical patterns of the PDTT	71
4.7	Prediction and verification of Pattern 1 from $t = 1$ to 9	72
4.8	Prediction and verification of Pattern 2 from $t = 1$ to 9	73
4.9	Prediction and verification of Pattern 3 from $t = 1$ to 9	74

表目次

2.1	Set of sensor labels in each subsystem	36
2.2	Evaluation value V_{ij} of arc from R^i to \mathbf{E}^j	38
2.3	Result of diagnostic method	40
3.1	Event labels in the transfer line	58
3.2	Fired events of processes for each destination	59
3.3	Command set for each actuator	59
3.4	Candidates of faults	59
3.5	Success rate of diagnosis with proposed method	60
3.6	Success rate of diagnosis with conventional method	60
3.7	The number of TTP in each method	60
3.8	Success rate of diagnosis when setting is changed	62
4.1	Statistics of departure and travel time (SDTT)	64
4.2	The average agreement rate for one day	71
4.3	Calculation time of proposed method and full search method	71

第 1 章

序論

離散事象システム (Discrete Event System: DES) とは、離散的な事象によりシステムの挙動が特徴づけられるシステムのことである。この離散事象システムは内部状態に離散的な状態を持ち、状態が不連続に変化する。また、時間など連続的な情報も含有してもよく、離散事象システムに分類される対象は生産システムや製造プロセス、コンピュータにおけるハード/ソフトウェア、通信システムなどが挙げられ、現代社会を支えるシステムの多くは離散事象システムに分類することができる。さらに、人間の行動も意図や目的が離散的に切り替わることから離散事象システムであると認識することもできる。

離散事象システムではシステムの状態の遷移を記述しているため、生起した事象 (イベント) の系列を信号源として捉え、その系列の順序に着目してモデル化する事象駆動型システムであると捉えることができる。そのようなモデルでは、時間などの要因によりイベントの生起が確率的な振る舞いをすることがあり、確率的なモデルで表現した方が好ましいケースも少なくない。また、確率的に事象駆動型システムを記述することにより、イベント系列の生起確率など確率論上の枠組みで議論することが可能となる。

事象駆動型システムを確率的に表現する手法として、準マルコフ過程がある。イベントの生起時間間隔も表現できるため、他のモデルに比べ表現能力が高く、他のモデルでは検討できない項目も検討できる。しかし、時間要素を含むためパラメータ数が増えてしまい、パラメータの同定や計算量が問題となる。また、対象とするシステムが大規模、複雑である場合はモデルのパラメータは非常に多く、経験的に設定することは難しいため、データから適切に設定する必要がある。

計算量やパラメータの学習量を削減するアプローチとして、システムあるいはモデルをいくつかの要素に分割する分散型手法が有効である。しかしながら、分散型手法では分散化した要素間の関係を記述する必要があり、これらの関係はシステムの動作に依存して決められるため、実際に稼働させて確認しなければならない。このような場合には、観測されたデータに基づき適切に要素間の関係を求めた方がよい。そこで本研究では、蓄積されたデータに基づいた分散型手法を用いることにより、現実的な時間内で計算できることを確認する。

1.1 事象駆動型システムの診断と予測

離散事象システムでは、そのシステムの特徴が離散イベントによって特徴付けられるため、生起するイベントに着目してシステムのモデル化やシステムの性質を議論することが多い。そのため、離散事象システムを事象駆動型システムとして、離散的な状態と生起するイベントの因果性からシステムを捉えることにより、様々なモデル化が可能となる。

1.1.1 事象駆動型システムの数理モデル

(A) 事象駆動システムとそのモデルの分類

事象駆動型システムとはあるイベントの生起（センサの発火など）により、制御や動作が切り替わるシステムである。事象駆動型システムではイベントが生起することにより、次の状態に遷移し、そして次のイベントが促されるとい過程の繰り返しとして動作が進行する。本節では事象駆動型システムについて説明し、そのモデルについて述べる。

事象駆動型システムの例として Fig. 1.1 のような生産プロセスを考える。この生産プロ

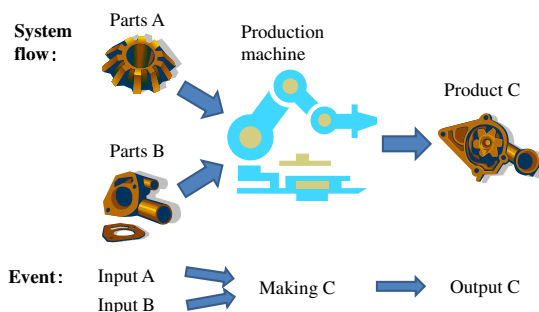


Fig.1.1 An example of discrete event system

セスでは部品 A と B が生産ロボットに投入され、製品 C を作成する。生起するイベントは “Input A”, “Input B”, “Making C”, “Output C” の 4 つであり、Input A と Input B が生起した後に製品 C の製造が開始され、Making C のイベントが生起する。そして、製品 C が完成して Output C のイベントが生起する。このとき、生起する系列の順序はパターン 1(Input A, Input B, Making C, Output C) とパターン 2(Input B, Input A, Making C, Output C) の二つの場合が考えられ、この系列の順序が事象駆動型システムにおいてモデル化される。

事象駆動型システムは、ダイナミカルシステムにおける微分方程式のような強力かつ汎用モデルは存在せず、事象駆動型システムの種類や目的とする問題の対象などに応じて様々なモデルが提案されている。Fig 1.2 のように、事象駆動型システムはイベントの生起順序のみに着目した論理モデルと、生起順序だけではなく生起時刻の両方を考慮した時

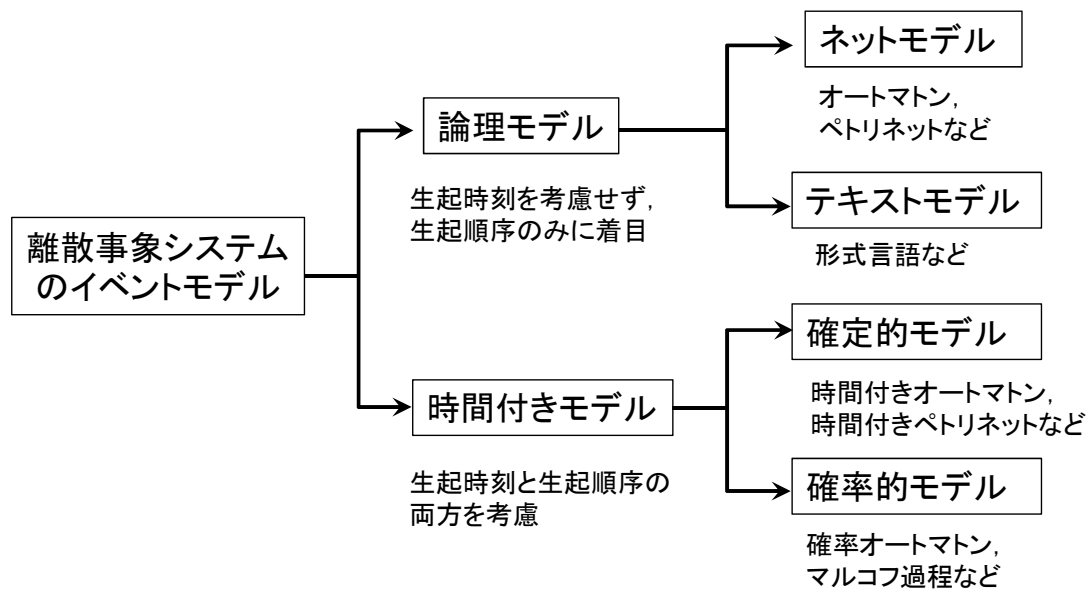


Fig.1.2 Category of event driven system

間付きモデルに分けることができる [1], [2].

■**論理モデル** 論理モデルではシステムの状態をグラフ構造で表現したネットモデルとイベントの系列をテキスト形式で表現するテキストモデルに分けることができる。

オートマトン [3], [4] やペトリネット [5] はネットモデルに分類される。ペトリネットはプレース、トランジション、トークンと呼ばれる記号を用いて、システムの状態をグラフ表現したネットモデルである。ペトリネットはイベントの並列性などを表現することができ、またそのモデル構造もシステムの構造をそのまま表現しているものも多い。また状態の可到達性、有界性、活性などシステムの基本特性を調べることができる [6]。オートマトンではグラフを用いて外部から逐次的に入力されるイベントに反応して変化する状態を表現したネットモデルである。ペトリネットとは異なり、イベントの並列性を表現することはできないが、現在の状態から次の状態の遷移の条件を直接記述しているため、システムの状態が定義されていれば容易にモデル化できることも多い。

テキストモデルの例として形式言語 [7] が挙げられる。形式言語では、生起したイベントの系列をアルファベットと呼ばれる記号と文法に従い言語として記述する。オートマトンとは密接な関係があり、いくつかの種類の形式言語はオートマトンと同等であることが示されている。ただし、イベントの系列のみをモデル化しているため、システムの状態との対応関係は直接表現することはできない。

■**時間付きモデル** 時間付きモデルでは時間を確定的に取り扱う確定的モデルと確率分布として取り扱う確率的モデルがある。

確定的モデルには時間付きオートマトンや時間付きペトリネット [8] が挙げられる。これらのモデルは論理モデルのオートマトンやペトリネットに、状態の停留時間や事象の発生時間間隔など時間の要素を取り入れたモデルである。ただし、この時間の要素は5秒後に遷移する、あるいは7秒以内に遷移するなど、確定的な情報として組み込まれる。このように時間の要素を取り入れることにより、時間的な挙動と生起するイベントの関係を記述することができ、モデルの表現能力が高くなる。しかし、このモデルは状態の遷移に関して時間を考慮しなければならないため、論理モデルに比べ解析や検証において困難になることが多い。

確率的モデルは時間を確率分布によって表現するモデルであり、確率オートマトンやマルコフ過程などによって表現できる。確定的なモデルでは時間を条件として考えていたが、確率的なモデルでは時間を確率分布として表現しているため、様々な状況を考慮してシステムを表現することができる。また、イベントの生起に関して確率的に表現しているため、事象駆動型システムを確率理論を用いて表現でき、他のモデルに比べ表現能力が高いことが特徴である。そのため、事象駆動型システムの時間やイベントに関して平均的なふるまい、つまり、性能評価に用いられることも多い。また、時間間隔のばらつきも表現できるため、時間付きのイベント系列の再現も行うことができる。しかしながら、確率をパラメータとして持つため、確率の同定や確率を用いた計算が課題になることが多い。

このように、時間付きの確率モデルは確率理論を用いているため、表現能力が高く、他のモデルでは検討できない部分も評価できることが多い。そこで、次節では事象駆動型システムにおける、時間付きの確率モデルの重要な性質であるマルコフ過程について詳しく述べる。

(B) マルコフ過程

時間付き確率モデルでは、現在の離散的な状態に依存して次の状態への遷移を確率で表現している。この性質はマルコフ過程と呼ばれ、時間付き確率モデルでは重要な性質である。本節ではマルコフ過程について述べ、その分類について説明する。

マルコフ過程とは次の状態が現在の状態のみに依存する確率過程である。マルコフ過程では過去の状態に依存せず、現在の状態のみで状態遷移確率を表現できるため、計算コストの削減、学習量の削減、最適性や最適解の検証を行うことができる。

時刻 t におけるシステムの状態を x_t とし、その確率変数を X_t とする。このとき、マルコフ過程であるとは

$$P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_1 = x_1) = P(X_t = x_t | X_{t-1} = x_{t-1}) \quad (1.1)$$

となることである。

現在の状態のみで次の状態への遷移を表現しなければならないため、厳しい条件であると思われるが、状態を拡大して考慮することにより、マルコフ過程として取り扱うことが

できる．たとえば，次の状態の遷移が N 前までの状態までに依存する (N 階マルコフ過程) 場合

$$\begin{aligned} P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_1 = x_1) \\ = P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-N} = x_{t-N}) \end{aligned} \quad (1.2)$$

を考える．このとき，状態を $\mathbf{x}_t = \{x_t, x_{t-1}, \dots, x_{t-N+1}\}$ ，その確率変数を \mathbf{X}_t として表現すると，

$$\begin{aligned} P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-N} = x_{t-N}) \\ = P(X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_{t-N-1} = x_{t-N-1} | \\ X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-N} = x_{t-N}) \\ = P(\mathbf{X}_t = \mathbf{x}_t | \mathbf{X}_{t-1} = \mathbf{x}_{t-1}) \end{aligned} \quad (1.3)$$

と書き直すことができる．このとき，システムは一階のマルコフ過程として見なすことができる．このように，システムの状態を拡大することにより，マルコフ過程として表現できる．

(C) 時間を含めたマルコフ過程

時間情報を含めたマルコフ過程として離散時間マルコフ連鎖 [9]，連続時間マルコフ連鎖 [10]，準マルコフ過程が存在する．

離散時間マルコフ連鎖では式 (1.1) のように離散時刻ごとに状態の遷移を考える．このとき，状態遷移だけではなく，自己遷移も実現される．つまり，現在の状態に滞在することを自己遷移によって表現され，単位時間ごとに状態が遷移する．一方，連続時間マルコフ連鎖では，連続時間においてマルコフ過程を考える．このとき，マルコフモデルの状態は単位時間ごとに遷移せず，状態遷移イベントが生じたある時刻において状態が遷移する．そのため，離散時間マルコフ連鎖のように自己遷移により状態の滞在を表現する必要がない．

離散時間マルコフモデルでは，単位時間ごとの状態の系列を表現しているのに対し，連続時間マルコフモデルでは，時間付きの状態遷移の系列を表現している．例として，晴れ，曇り，雨の天候をマルコフモデルによりモデル化する場合を考える．離散時間マルコフモデルでは，Fig. 1.3 のような状態遷移を各離散時刻ごとに考える．そのため，一日目：晴れ，二日目：晴れ，三日目：曇りなど単位時間ごとの状態系列を表現している．また，同じ状態を連続して取り得る．一方，連続時間マルコフモデルでは，Fig. 1.4 のように状態の遷移のみに着目してモデル化される．そのため，一日目：晴れ，三日目：曇り，六日目：雨など時間的に離れた系列を表現している．また，状態が遷移した場合のみを考慮しているので，同じ状態を取らない．しかしながら，この二つのマルコフ連鎖は共通して時間情報を確定的に取り扱い，状態の遷移のみを確率的に取り扱っている．そのため，どのような場合においても現在の状態が同じであるならば，次の状態に遷移する確率は同じである．(1 日だけ曇り，5 日連続曇りなど状態の滞在時間によらず，状態のみに依存する)．

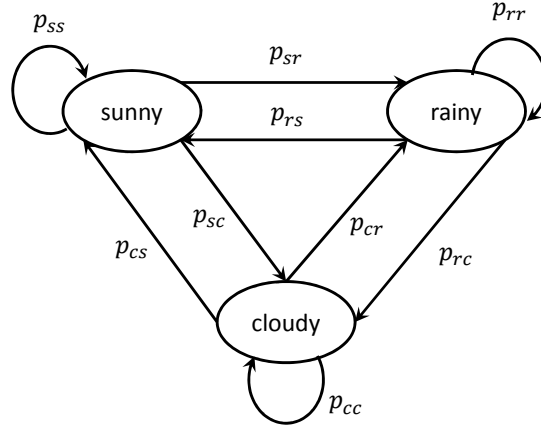


Fig.1.3 An example of discrete time Markov model

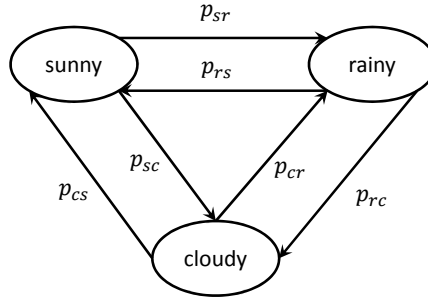


Fig.1.4 An example of continue time Markov model

状態の滞在時間に応じて状態遷移確率を変動させるマルコフ過程として，準マルコフ過程が存在する．準マルコフ過程では現在の状態の滞在時間に応じて，状態遷移確率が変化する．現在の状態の滞在時間が τ ， τ の確率変数が T であるとき，状態の遷移確率は

$$P(X_{t+\tau} = x_{t+\tau}, T = \tau | X_t = x_t) \quad (1.4)$$

として表現でき，時変の関数であると見なすことができる．この時，Fig. 1.3 の例を準マルコフ過程を適用すると，Fig. 1.5 として表現できる．

準マルコフ過程は他のマルコフ過程とは異なり，時間経過による確率の変動や，状態の遷移時間間隔をモデル化している．そのため，事象駆動型システムにおけるイベントの時間変動を表現でき，表現能力が高い．しかし，離散事象システムにおける組み合わせ問題，時間情報によるパラメータの増加など学習量や計算量の多さが問題となる場合がある．

(D) 事象駆動型システムに対する準マルコフ過程

時間付き確率モデルでは確率理論を導入しているため，事象駆動型システムの他のモデルとは異なり，確率推論を行うことができる．特に準マルコフ過程では，状態遷移確率を時変な関数として取り扱うため，時間による確率の変動を陽に取り扱うことができる．そ

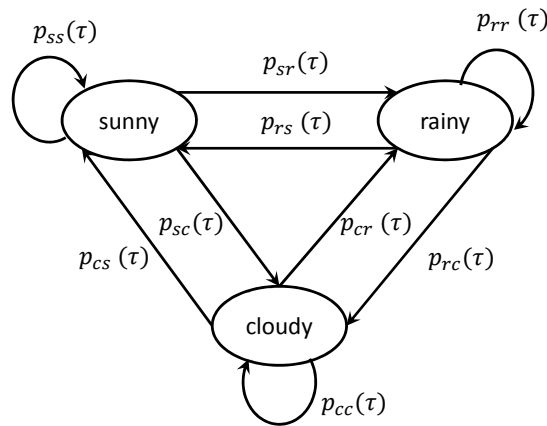


Fig.1.5 An example of semi Markov model

のため、事象駆動型システムに対して準マルコフ過程を適用することにより、時間経過による状況の変化を考慮した確率推論が可能となる。

事象駆動型システムにおける確率推論の対象としてはシステムの診断問題、未来の状態予測問題の二つが挙げられる。システムの診断問題とは、隠れ状態やシステムの動作に影響を与える要因などを推定する問題であり、例としては故障診断問題がある。一方、状態予測問題とは、未来の状態や出力の系列を推定する問題である。いずれの問題においても、対象のシステムを事象駆動型モデルとして表現し、そのモデルを用いて診断器、あるいは予測器を作成することにより実現される。

1.2 診断や予測問題における従来手法

1.2.1 診断問題

事象駆動型システムにおける診断問題では、隠れ状態やシステムの動作に影響を与える要因が診断の対象となる。本節では診断問題の中でも多く研究されている故障診断問題について説明する。

■**故障診断問題** プログラマブルロジックコントローラ (Programmable Logic Controller: PLC) に代表される事象駆動型システム (Event-Driven System) は、化学プラント (Chemical Plant: CP) や工場自動化 (Factory Automation: FA) などの分野において、巨大プラントから局所アクチュエータに至るまで、幅広い階層における各種生産設備の自動制御に用いられており、制御器全体の約 8 割と極めて高い割合を占めている [11]。これらの生産設備に対する要求は従来の高速・高効率といった観点だけではなく、安全・高信頼へと変わりつつあり、これらの要求にこたえるシステム設計論の確立が急務となっている。近年では生産システムの大規模複雑化に伴い、運用業務で要求される専門的知識

や処理作業は増大する一方であり、人為的ミスが多く割合を占めるようになったため、従来の技術者の経験に頼った運用方式ではシステムの信頼性を保障することが難しくなっている。このような背景の下、システムの故障の検知やその原因・重要性の判断、そしてそれらに対する適切な対応や復帰制御といった一連のタスクを計算機により自動化し、故障検出・同定を行う故障診断システム、ならびに故障からの復旧などのための故障復帰システムの実用化 [12], [13] が望まれている。

イベントとはセンサーの発火情報であり、事象駆動型システムとはイベント信号をもとに制御されるシステムである。事象駆動型システムにおける故障診断手法の多くはイベントに着目しており、正常時と故障時に観測されるイベントの系列をモデル化し、そのモデルの違いにより故障を診断する。離散事象モデルを用いた故障診断法 [14]~[16] としてはいくつか研究されている。Sampath ら [15] は事象駆動型システムにおける、センサーから生起する可観測なイベントと故障の生起など不可観測のイベントを考慮して、システムをオートマトンモデルで表現した。このモデルをオブザーバとして動作させることにより、不可観測な故障の生起を診断することができる。Riascos ら [14] や Antonio ら [16] はシステムをペトリネットを用いて表現することにより、故障診断を行った。これらの手法は可診断性を定義することができ、診断可能な故障は必ず検出することができる。しかし、これらは確定的手法に分類され、システムの内部状態を正確にモデル化できる場合のみに有効である。システムの内部状態を正確にモデル化できない場合、これらの確定的なモデルを用いるよりも、確率的なモデル [12], [17]~[19] を用いる方が、不確実性を許容している点でより現実的なモデルが得られる。また、確定的な手法ではイベントの生起順序のみに着目するため、他の故障や正常状態とイベントの生起順序が変化しない場合は故障を診断することはできない。しかしながら、確率的な手法ではイベントの生起確率に差がある場合は、その差から故障を診断することができる。

確率的なモデルを用いた手法として、準マルコフ過程である時間付きマルコフモデルによる確率的手法 [17], [18] がある。このモデルはイベントの生起順序だけではなく、生起時間間隔の確率の変動も表現しているため、確定的なモデルでは診断できないような故障を診断できる。

しかしながら、時間付きマルコフモデルはイベントの生起時間間隔を確率的に表現するため、多くの学習データを必要とする。また、離散的なイベントの組み合わせを考慮しなければならないため、計算量が多いという問題点がある。これらの問題は対象のシステムの大きさや複雑さに応じて顕著になっていくため、生産システムのような大規模・複雑なシステムにおいて問題となる。

このような学習量や計算量の多さが問題となる場合の有効的なアプローチとして分散型手法がある。これはモデルをいくつかの部分に分割して表現、計算する手法である。このようにシステムをいくつかの小さなサブシステムの集合と捉えることにより、学習量や計算量の削減が期待できる。

本論文では、まず故障診断問題に対して準マルコフ過程を適用する。そして、分散型手

法により問題となる学習量や計算量の削減を検討する。

1.2.2 予測問題

本研究で対象とする予測問題とは過去の観測情報とシステムのモデルを用いて、未来の出力を予測する問題である。連続系のダイナミカルモデルでは、現在状態とシステムへの入力によって出力が一意に定められるため、比較的容易に問題を記述することができる。しかし、事象駆動型システムのモデルの場合は、システムの状態における遷移順序のみを記述しているため、未来の状態遷移の系列は複数種類ある場合がある。そのため、状態遷移し得る各状態遷移系列に対して出力も予測する必要があるため、複数種類の出力が予測できる。例えば、Fig. 1.1.1 のシステムでは Input A, Input B, Making C, Output C というイベントの生起順序と Input B, Input A, Making C, Output C というイベントの生起順序の二種類の動作がある。予測問題では、これらの動作に対して未来の出力を予測する。このとき、事象駆動型システムの予測問題ではいくつかの未来の出力が考えられるため、どの候補が尤もらしいかを議論する必要があるため、確率モデルを用いて確率推論を行う必要がある。

事象駆動型システムの確率モデルとしては、確率オートマトン、確率ペトリネット [8], マルコフモデル [20], [21] が挙げられる。確率オートマトンや確率ペトリネットでは、表現する対象や目的は異なるが、それらのモデルの確率推論では可達グラフを考えマルコフ過程を用いて議論されることが多い。そのため、モデルが異なっていたとしても、予測問題ではモデルをマルコフモデルとして表現して考えることができる。

予測問題においてよく用いられるモデルは単純マルコフモデルや、隠れマルコフモデル (Hidden Markov Model: HMM)[21] が挙げられる。HMM とは観測できない隠れ状態を持つマルコフモデルであり、明示していない隠れ状態を持つため情報源の隠れた性質や傾向を表現することができる。この性質を持つため、HMM は診断器や識別器として用いられることが多いが、予測器として用いることができる。岸本ら [22] は HMM を用いて車の停止行動をモデル化し、そしてドライバの操作量を予測した。停止行動は意図などの隠れた離散状態変化を含むため、HMM によってモデルを行っている。しかしながら、HMM において獲得された隠れ状態は実際のシステムの状況と対応しているとは限らず、隠れ状態の意味付けをシステムと対応させることは難しい。

隠れ状態を持たない単純マルコフモデルを用いた予測も研究されている。Begleiter ら [23] は可変な次元数のマルコフモデルを用いて、自然言語中に現れる有限な記号列の遷移則を学習し、その予測器を作成した。この問題では隠れ状態を持たないため、単純マルコフモデルとして表現することができる。このように、HMM と単純マルコフモデルの違いは隠れ状態を持つかどうかであるので、どちらのモデルが適しているかは予測の対象によって異なる。

マルコフ過程を用いることにより、状態の遷移や出力を確率で表現することができ、確

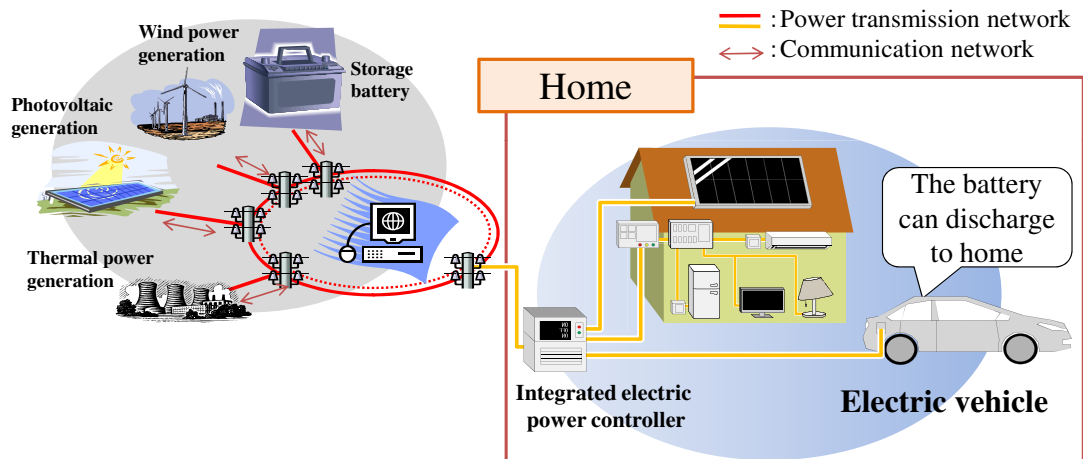


Fig.1.6 Configuration of smart grid, HEMS and electric vehicle

率推論を用いて予測することができる。しかしながら，確率理論では確率分布をパラメータとして持つため，確率分布をデータを用いて適切に決定する必要がある。また，マルコフ過程の状態遷移の数は予測の期間に応じて組み合わせ的に増加する。そのため，予測問題では計算時間が問題となることがある。

事象駆動型システムの予測問題では，予測対象を確率的なモデルでモデル化することができる。できれば，確率推論として定式化することができる。そのため，予測問題の対象となるシステムは多くのシステムを挙げることができる。本論文では，予測問題の一例として車の使用行動について取り上げ，その問題点を検討する。

■車の走行予測問題 電気自動車 (Electric Vehicle:EV) やプラグインハイブリッドカー (Plug-in Hybrid Vehicle:PHV) は大容量の蓄電池を備えていることから，高度交通システム (Intelligent Transportation System:ITS) の分野だけではなく，エネルギーマネジメントシステム (Energy Management System:EMS) の分野においても活躍が期待されている。EV や PHV が乗り物 (車) としての働きだけではなく，自宅や他の設備に対する電力ストレージとしての働きができることを示している ([24], see, e.g. Fig. 1.6) 。また，EV, PHV の充電は長い時間を要することから，柔軟な充電や放電スケジュールをするためのシステムの開発が望まれている。Caramanis ら [25] は車の到着と出発のパターン (Profile of the Departure and Travel Time: PDTT) の情報が既知であるとして，EV や PHV の充電手法を提案した。しかし，多くの場合，未来の PDTT を得ることは難しい。車載蓄電池の充放電計画を行うためには，車の PDTT の予測問題は重要な課題である。

車の走行の予測はいくつか研究されている。Soheil ら [26] や Ali ら [27] は町や市などの地域の車の走行行動を対象に調査を行った。彼らは地域内の車を個別の車としてではなく，車群として捉え，地域内の車の走行傾向について議論した。Mei ら [28] や Chun ら [29] は出発と到着の統計データ (Statistics of Departure and Travel Time: SDTT) や環

境情報、目的地などの情報から到着時刻を予測する手法を提案している。これらのシステムはカーナビゲーションシステムを対象としており、出発時刻については予測の対象としていない。Dick ら [30], [31] は人間の移動行動を trip と activity に分割し、SDTT から運転行動を予測する手法を提案している。これらは予め人間の行動パターンをモデル化する必要があり、走行回数などを予め設定する必要がある。

EV や PHV を ITS と EMS の両方のシステムにおいて最大限有効活用するためには、各車の走行傾向に合わせた PDTT の予測は重要な問題である。車の使用状況は車を乗り物 (ITS) として使用している状態と蓄電池 (EMS) として使用している状態の二状態として表現でき、PDTT はそれらのバイナリ変数の系列として表現できる。しかしながら、PDTT はバイナリ変数の組み合わせとして表現されるため、その状態空間はとても大きくなってしまふ (例として一日を 48 ステップに離散化した場合、実現数は 2.8×10^{14} 通りとなる)。そのため、リアルタイムで PDTT を予測するためには計算量を削減する工夫が必要となる。

本論文では、まず準マルコフ過程を用いて車の走行予測問題のモデル化を行う。そして、問題となる計算量の多さを、動的計画法を用いて解決を目指す。

1.3 データ活用の必要性

事象駆動型システムのモデリングのアプローチとしては、システムの熟知者によるシステムの設計に基づくモデリングと過去の蓄積されたデータに基づくモデリングがある。ペトリネットなどの確定的なモデルでは、状態の遷移則を記述すればよいので、設計に基づきモデリングされることが多い。一方、確率的なモデルでは、状態の遷移確率を設定する必要があるため、データに基づきモデリングされる。

また、事象駆動型システムは比較的簡単なロジックで表現されるため、システムの設計に基づくモデリングによって表現されることが多い。しかしながら、システムが大規模・複雑な場合は全体の制御ロジックを把握することは困難であるため、設計に基づくモデリングは多大な労力を要することが多い。このような場合は、過去の蓄積されたデータから生起するイベントの順序規則を抽出し、その規則に基づきモデリングする、データに基づくモデリングが有効となる。

システムをデータに基づき確率によって表現することにより、生起しやすいイベントと生起しにくいイベントを実システムに合わせて表現できる。診断問題と予測問題では、この生起のしやすさによって診断や予測を行うため、データに基づきモデリングすることは重要である。しかしながら、データに基づくモデリングでは多くの学習データを必要とする場合がある。これはシステムが大規模・複雑な場合に顕著となり、モデル構築自体が不可能となる場合もある。

この学習量の多さを解決するためのアプローチとして分散型手法 [32] がある。システ

ムをいくつかの要素として記述することにより，システムをより小さく，簡単に表現できる．しかし，分散型手法では分割された構造の因果関係を記述する必要がある．この分割化された要素間の因果関係も，データから適切に設定する必要がある．

上記のように，事象駆動型システムのモデルにおける診断や予測問題では，データに基づきモデルを適切に設定する必要がある．次節では，この点を踏まえ，研究目的を設定する．

1.4 研究目的

事象駆動型システムに対して準マルコフ過程を用いてモデル化することにより，様々な性質や検証を行うことができる．しかしながら，モデルの複雑さから学習量や計算量の多さなどが問題となる場合がある．本論文では以上のことがらを踏まえ，以下の三点を目的として設定する．

1. **確率理論を用いた事象駆動型システムの状況の推定** 第一に，事象駆動型システムにおける準マルコフ過程の特徴として，確率理論を用いて時間情報を陽に取り扱うことができる点が挙げられる．そこで，本論文では生起するイベントに影響を与える要因を“状況”として捉え，観測されるイベント系列から状況を推定する．そして，状況として故障状況について考え，その有効性を検討する．
2. **準マルコフ過程を用いた予測器の設計** 第二に，事象駆動型システムにおける準マルコフ過程の特徴として，予測が行えることが挙げられる．事象駆動型システムのモデルでは，イベントの順序性のみをモデル化したものが多く，次にどのイベントが生起しやすいかなど議論できないことが多い．一方，準マルコフ過程では時間情報を含め尤度を計算できるため，予測器として運用することができる．例として車の走行予測について考え，議論する．
3. **観測データに基づく分散型手法の適用** 第三に，事象駆動型システムにおける準マルコフ過程の特徴として，モデルの複雑さ故に学習データ量や計算量が多くなってしまいうことが挙げられる．これらの問題の解決法として分散型手法の適用が挙げられる．本論文では，観測データに基づきシステム，モデルなどを適切に分割し，学習データ量や計算量が低減できることを確認する．

1.5 本論文の構成

本論文では Fig. 1.7 のような 5 章構成により構成される．

2 章は事象駆動型システムの故障診断問題について概要を説明し，準マルコフモデルを用いて事象駆動型システムをモデル化する．そして，計算量や学習量の多さが問題となるため，システムをいくつかのサブシステムに分割し，ベイジアンネットワークを用いて因

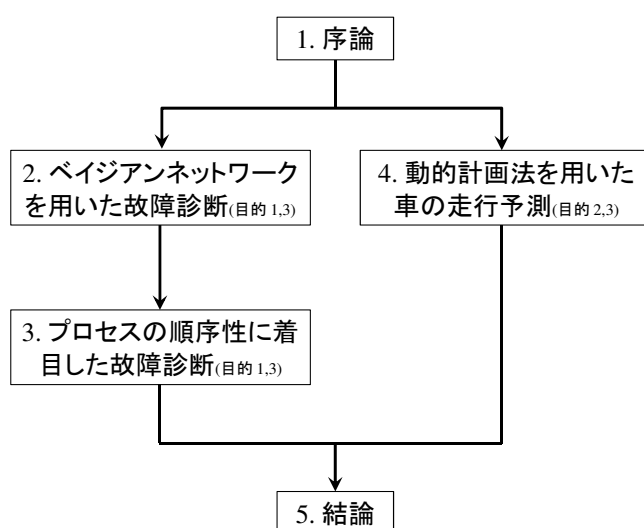


Fig.1.7 The structure of this paper

果関係を記述する故障診断手法について説明する．まず，2.1 節では準マルコフ過程を用いた故障診断手法とその問題点について述べ，ベイジアンネットワークを用いた分散型手法について説明する．2.1.1 節では，観測されるイベントと故障を定義し，故障診断手法の定式化を行う．また，局所診断器と大域診断器のアーキテクチャにて表現される分散型故障診断手法について説明する．そして，観測されるイベント系列を準マルコフ過程である，時間つきマルコフモデルによりモデル化し，局所診断器における故障診断手法を 2.3 節にて述べる．2.4 節では，ベイジアンネットワークを用いた大域診断器の故障診断手法について説明する．ベイジアンネットワークはサブシステム間の因果関係を表示したグラフであり，その構造は診断精度や計算量に大きく寄与するため，2.5 節にてベイジアンネットワークの構造決定手法を説明する．2.6 節は自動搬送システムに分散型故障診断手法を適用し，その性能を検討する．

3 章では，事象駆動型システムの順序性に着目する．事象駆動型システムでは，生起したイベントに関連して制御が行われるシステムであるため，生起するイベントはある程度の順序性を持つことが期待できる．このような順序性を準マルコフ過程に導入することにより，生起するイベントに関する組み合わせ問題を低減することができる．3.1 節では，対象とするシステムの概要について説明する．3.2 節では，故障診断問題の定式化と順序性についての表現について述べる．3.3 節では，順序性に着目した分散化アーキテクチャについて説明し，3.4 節では順序性に着目した時間つきマルコフモデルについて述べる．3.5 節では，時間つきマルコフモデルのパラメータである時間つき状態遷移確率について議論する．3.6 節では自動搬送ラインにおけるイベントの順序性について説明し，故障診断手法を適用する．

4 章では，動的計画法を用いて車の走行予測を行う．4.1 節では，車の走行予測問題の概要について述べる．車の走行行動は走行している，あるいは走行していないという 2 状態

で表すことができ、4.2 節では、この表現に基づく車の走行パターンとその表現に基づく統計データの集積方法について述べる。4.3 節では、車の走行予測の定式化を行う。しかしながら、車の走行パターンは組み合わせ問題を含んでいるため、解の計算には多くの時間を要する。この問題を解決するため、車の走行行動に対して準マルコフ過程を用いてモデル化し、動的計画法を適用する。そして、4.4 節において、被験者のデータを用いてシミュレーション実験を行い、その性能を評価する。

第 2 章

Bayesian Network を用いた分散型故障診断法

2.1 はじめに

故障診断の手法として連続時間系と離散事象系がある。工場の自動搬送ラインなどの大規模システムを対象とする場合、詳細な連続時間モデルよりも、構成要素間の因果関係にのみ着目した離散事象モデルのほうが必要な情報のみを表現できるため有用である。離散事象モデルを用いた故障診断法として、オートマトン、ペトリネットを用いた手法 [14][15] が提案されている。しかし、これらは確定的手法に分類され、システムの内部状態を正確にモデル化できる場合にのみ有効である。システムの内部状態を正確にモデル化できない場合、これらの確定的なモデルを用いるよりも、確率的なモデルを用いる方が、不確実性を許容している点でより現実的なモデルが得られる。確率的なモデルを用いた手法として、準マルコフ過程である、時間付きマルコフモデルによる確率的手法 [17][18] がある。これらの手法はイベント間の生起時間間隔を統計的に表現したモデルを用いるが、集中型故障診断手法であるためにシステムが大規模である場合は計算量が問題となる。

大規模なシステムに適用する場合はシステムをいくつかの単純な構造に分割する分散型故障診断 [32], [33] が有用である。文献 [32] の手法では、まずシステムを幾つかのサブシステムに分割し、それぞれを時間付きマルコフモデルを用いて表現し、局所診断器を作成する。その局所診断器とベイジアンネットワーク (Bayesian Network:BN)[20], [21] を用いて、システム全体の故障診断を行う。BN は故障と観測イベント系列の関係を表現しており、その構造により診断精度と計算量が左右される。しかしながら、ネットワーク構造はシステムの特性やその制御則など専門家の経験的知識から構築され、ネットワークを構築するには専門の知識を要するため、適切なネットワーク構造を構築することは容易ではない。したがって、システムから得られる情報によってネットワーク構造を構築する方法が求められている。

2.1.1 ベイジアンネットワークの学習手法

BN は確率変数間の因果関係を表現した確率的なグラフィカルモデルであり、ノードとアークから構成される。ノードは確率変数を、アークはそれらの因果関係の有り無しを表し、変数間の因果関係は条件付き確率によって表現される。各ノードに条件付き確率が割り振られ、これらを用いて確率的な推論が行われる。BN の構造とはノード間のアークの引き方であり、アークの有り無しによってノードの条件付き確率が変わり、その結果として推論結果が大きく変化する。

データから BN の構造を学習する手法はいくつか提案されており、スコアベース学習 [34], [35] と制約ベース学習 [36], [37] に大別される [38]。スコアベース学習は与えられたデータに対する BN の構造の尤度を計算し、最も尤度の高いものを選択する手法である。この手法における問題は探索すべき BN の候補が多すぎることである。例えばノード数 10 の BN の全候補は 4.2×10^{18} にも及び*1、ノード数の増加に伴い指数的に増えてしまう。制約ベース学習はデータから各ノード間の条件付き独立性を検証し、独立性の度合から BN の構造を構築するという手法である。この手法ではアークの向きは唯一に定まらず、構造を一意に決定できないことが問題となる。これらの学習法は問題設定に応じて、適切な手法を選択すべきである。本論文の問題設定として二部グラフの BN を考える。このときアークの向きは一意に定まり、制約ベース学習は適用しやすくなるため、提案手法では制約ベース学習を用いる。

条件付き独立性はノードに割り振られている条件付き確率から評価できる。提案手法における BN はセンサの発火時系列の観測と故障の因果関係を表現しており、条件付き確率を求めるには、時間付きの観測系列の発生確率を計算する必要がある。しかし時間は連続量であり、かつその組み合わせである系列を考えるので、発生確率を計算することは困難である。このような場合では、条件付き確率を直接計算することができないので、ネットワーク構造を構築することは難しい。そこで本論文では、局所診断のモデルである TMM から、アークの重要度を示す指標を間接的に求めることにより、BN の構造を構築する。

本研究では、サブシステム間の因果関係をベイジアンネットワーク (Bayesian Network:BN) を用いて表現し、システム全体を考慮に入れた大域的な故障診断を行う。本章では、提案する BN の構築法を用いた事象駆動型システムの故障診断法に関して説明する。

2.2 節にて、本論文で扱う故障診断の問題設定について説明する。次に 2.2.1 節において、提案手法の中核を担う BN に関する概論を記す。BN とは、変数を表すノードと変数間の依存関係を表す有向リンクからなる非循環グラフである。そして、BN の概要およびその応用分野について簡単に述べる。次に、BN の具体例として “Burglar Alarm” を紹介

*1 アークは順方向、逆方向、なしを考え、かつ閉路を含まないベイジアンネットワークの総数である

し、その例を基にして BN を用いた確率推論法を示す。

2.2.2 節では、2.2.1 節で説明した一般的な BN を拡張し、本論文で対象とする事象駆動型システムの故障診断に特化した形式を示し、さらに故障推論方法に関して説明する。提案手法では、各サブシステムの故障事例をラベルとする離散確率変数と観測したイベント系列をそれぞれノードとし、そして、故障-イベント系列間の因果関係を BN のグラフ構造によって表現する。その後、全サブシステムに関してイベント系列が観測可能な場合に、それらのイベント系列を基にして、あるサブシステムの故障状況を推論することが可能となる。

2.2.3 節では、故障診断の手順および具体的な故障診断法について述べる。故障診断の手順は、「学習過程」と「診断過程」に大別することができ、それぞれの過程について順を追って説明する。

2.2 故障診断問題

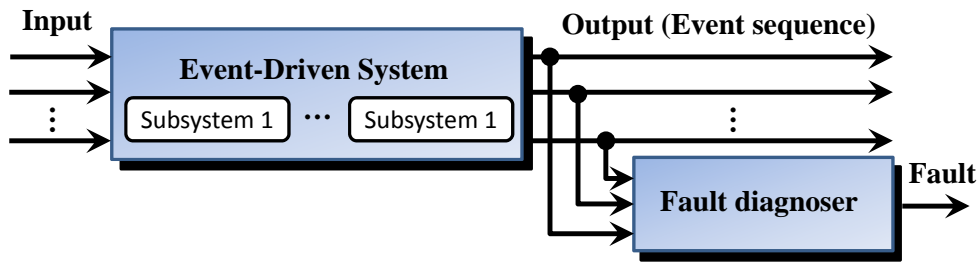
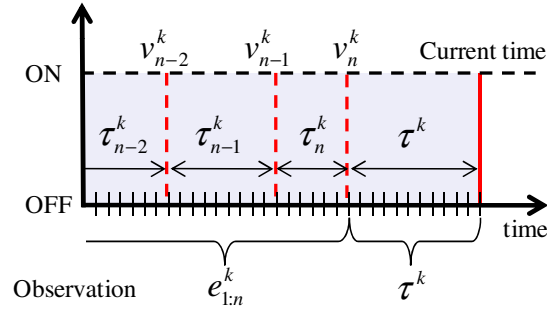


Fig.2.1 Block diagram of relationship between event-driven system and decentralized fault diagnoser

システムの特長や物理的影響を考慮して、システム全体を幾つかのサブシステムに分割できるものとする。以後、システム全体を m 個のサブシステムに分割したものとして議論を進める。各サブシステム毎にセンサの活性を示す時間付きイベントが系列として観測できるものとする。イベントは発火したセンサのラベルとその時刻のセットである時間付きイベントとして観測される。サブシステム k において n 回目に観測された時間付きイベントを次のように定義する。

$$e_n^k = (v_n^k, \tau_n^k). \quad (2.1)$$

v_n^k は発火が観測されたセンサのラベルであり、 τ_n^k は前回、つまり $n-1$ 回目に、観測されたイベントからの時間間隔である。なお時間付きイベントの時間は時刻で表されることもあるが、本論文ではモデルの性質から時間間隔で表現する。初回に発火したセンサのラベル v_0^k は TMM の初期値を設定するために用い、それ以降の時間付きイベント系列 $e_{1:n}^k = (e_1^k, e_2^k, \dots, e_n^k)$ を観測として用いる。Figure 2.2 はサブシステム k において観測されたセンサの発火とそれらの時間間隔を表しており、観測イベント系列 \mathbf{e}^k を時間付き


 Fig.2.2 Observation in subsystem k

イベント系列 $e_{1:n}^k$ と、センサが発火していない時間間隔 (最後のイベントから現在時刻までの時間間隔) τ^k で定義する:

$$\mathbf{e}^k = (e_{1:n}^k, \tau^k). \quad (2.2)$$

ここでシステム全体の故障要因の集合は既知であるとし、故障の確率変数を R , R の実現値を $r \in \{r_0, \dots, r_\kappa\}$ と定義する. このとき観測イベント系列 \mathbf{e}^k に対して, 局所診断器は以下の計算を行う.

Given : Observed event sequence \mathbf{e}^k ,

Calculate : Local fault probabilities for all $r \in \{r_0, \dots, r_\kappa\}$

$$P(R = r | \mathbf{E}^k = \mathbf{e}^k).$$

ただし, \mathbf{e}^k の確率変数を \mathbf{E}^k とする. また大域診断における, 現在のシステムの故障 r^* を同定する故障診断問題は次のように書ける.

Given : Local fault probabilities for all $r \in \{r_0, \dots, r_\kappa\}$

$$P(R = r | \mathbf{E}^1 = \mathbf{e}^1), \dots, P(R = r | \mathbf{E}^m = \mathbf{e}^m),$$

Find : $r^* = \arg \max_r P(R = r | \mathbf{E}^1 = \mathbf{e}^1, \dots, \mathbf{E}^m = \mathbf{e}^m),$

$$r \in \{r_0, \dots, r_\kappa\}.$$

2.2.1 Bayesian Network 概説

Bayesian Network(BN) とは, 複数の確率変数間の定性的な因果関係をグラフ構造によって表し, 個々の変数間の定量的な関係を条件付き確率で表した確率的グラフィカルモデル (または確率的ネットワークモデル) である [39][40][41]. BN は, 確率変数と, その間の依存関係を表すグラフ構造, 条件付き確率, の集合によって定義される. そして, 確率変数はノード (グラフ構造中の丸印) として, 変数間の因果関係は有向リンク (グラフ構造中の矢印) で示される. 例えば, 確率変数 X_i から確率変数 X_j への依存関係がある場

合, ノード X_i からノード X_j へリンクを張ることでその依存関係を表現する. BN は, 従来計算機にとって比較的扱いにくかった不確実な情報を扱うことができるモデルであり, 人工知能や情報処理を中心とした分野で盛んに研究がなされている [42][43]. 以下に列挙する様に, BN の応用例はかなり幅広い分野に至る.

- 医療 [44]・故障診断 [45][46][47]
- プランニング・制御
- パターン認識 (音声認識, 文字認識, 状況認識)[48]
- ユーザモデリング [49]
- データマイニング [50]

BN の具体例として “Burglar Alarm” を Fig. 2.3 に示し, その例を用いて BN による確率推論法に関する説明を行う. Fig. 2.3 は, 泥棒 (Burglar) が侵入すると警報装置 (Alarm) が作動すること, また警報装置は地震 (Earthquake) によっても作動してしまうこと, そして地震が起きるとラジオ (Radio) で伝えられることをモデル化した BN である. 各ノードに対してそれぞれ, B: [泥棒が侵入している], A: [警報装置が作動している], E: [地震が発生している], R: [ラジオで情報が流されている] という命題を割り当て, それらの命題に対して『真 (1)』, 『偽 (0)』の 2 つの状態をとる確率変数をそれぞれ B, A, E, R により表すものとする. つまり各確率変数は実現値が 0 または 1 である離散確率変数である. このように BN のノードとして, 離散または連続な確率変数を割り当てるのが一般的である. また, ここでは条件付き確率および事前確率も既知であるとする. 通常では, 事前に得られた大量の統計データを最も良く説明するように条件付き確率を決定しなければならない. そのために, EM アルゴリズム^{*2}や MCMC 法^{*3}などを用いた学習アルゴリズムが提案されているが, 計算時間や収束精度などの困難が多く, あまり実用化には至っていない [41].

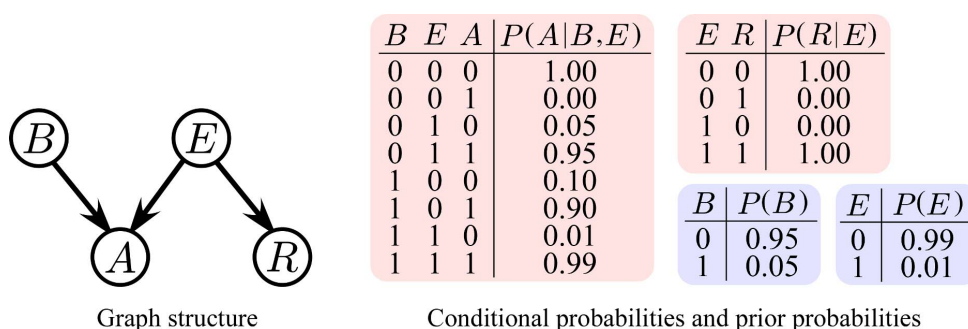


Fig.2.3 Example of Bayesian Network (Burglar Alarm)

^{*2} 観測できない隠れたパラメータが存在する時に最尤推定を行うための汎用手法であり, Newton 法や勾配法と同様, 反復法によって局所最適解を求めるアルゴリズムである.

^{*3} マルコフ連鎖モンテカルロ法 (Markov Chain Monte Carlo) のこと.

再び, BN を用いた確率推論法の説明に戻ることにする. まず, グラフの構造から全ての確率変数に関する同時確率が式 (2.3) の様に一意に決定される.

$$P(B, A, E, R) = P(B)P(A|B, E)P(E)P(R|E) \quad (2.3)$$

BN を用いた確率推論は, この同時確率を周辺化した周辺事後確率により行われる. 例えば, 警報が作動した時に泥棒が侵入している確率を推論する場合, この確率は式 (2.3) の同時確率を利用すると, 以下の様にして算出することができる.

$$\begin{aligned} P(B = 1|A = 1) &= \frac{P(A = 1, B = 1)}{P(A = 1)} \\ &= \frac{\sum_E \sum_R P(B = 1)P(A = 1|B = 1, E)P(E)P(R|E)}{\sum_B \sum_E \sum_R P(B)P(A = 1|B, E)P(E)P(R|E)} \\ &= \frac{P(B = 1) \sum_E P(A = 1|B = 1, E)P(E)}{\sum_B P(B) \sum_E P(A = 1|B, E)P(E)} \\ &= 0.833 \end{aligned} \quad (2.4)$$

さらに, 地震のニュースをラジオで聞いた場合, つまり新たに証拠 (Evidence) が観測された場合, 泥棒が侵入した確率は以下の様に修正される.

$$\begin{aligned} P(B = 1|A = 1, R = 1) &= \frac{P(A = 1, B = 1, R = 1)}{P(A = 1, R = 1)} \\ &= \frac{\sum_E P(B = 1)P(A = 1|B = 1, E)P(E)P(R = 1|E)}{\sum_B \sum_E P(B)P(A = 1|B, E)P(E)P(R = 1|E)} \\ &= \frac{P(B = 1) \sum_E P(A = 1|B = 1, E)P(E)P(R = 1|E)}{\sum_B P(B) \sum_E P(A = 1|B, E)P(E)P(R = 1|E)} \\ &= 0.052 \end{aligned} \quad (2.5)$$

上記の例の様に BN を用いることで, 一部の変数を観測した時のその他の変数に関する確率分布を求めたり, 確率値が最も大きい状態をその変数の予測結果として得ることができる. つまり, 観測された変数の情報 (e) から, 求めたい確率変数 (X) の確率値, すなわち事後確率 $P(X|e)$ を求めることができ, 「異常を観測したセンサの情報 (状態) からシステムの故障原因を推定する」といったように, 故障診断の枠組みでも BN が利用可能であると考えられる.

2.2.2 Bayesian Network による因果関係の表現と確率推論法

本論文の問題設定では, サブシステム間の因果関係, 具体的にはサブシステム間をわたる故障とイベント系列間の因果関係を記述する必要があり, 以下に説明する BN を用いてその因果関係を表現する. なお BN は Fig. 2.4 のような二部グラフを考えることとする.

まずシステム全体を m 個のサブシステムに分割したことを考慮して, システムの故障 R をサブシステム k の故障 R^k に分割できると仮定する.

$$R = (R^1, R^2, \dots, R^m). \quad (2.6)$$

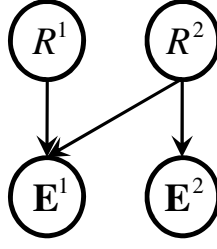


Fig.2.4 Example of Bayesian Network (Proposed method)

また, r^k を R^k の実現値とし,

$$r^k \in \{r_0^k, r_1^k, \dots, r_{n_k}^k\}, \quad (2.7)$$

とする. ただし n_k はサブシステム k における故障数である. また, r_0^k は正常であることを示すラベルである. 提案手法において, 各サブシステムの故障 R^k と観測したイベント系列 \mathbf{E}^k をそれぞれノードとする. 通常, BN のノードには離散/連続確率変数を割り当てるが, 本論文では系列 (Sequence) もノードに割り当てることが可能であるとして今後の議論を進める. これは, 系列をまとめて確率変数の実現値の一つとみなしているためである.

各サブシステムで想定した故障と観測されるイベント系列間の因果関係を考慮しネットワーク構造について考える. サブシステム k において故障が生じた場合, かつ, サブシステム l で観測されるイベント系列 \mathbf{E}^l が変化するならば, ノード R^k からノード \mathbf{E}^l へアークを張られる. ここで, 各サブシステムの故障間に直接的な因果関係が無い^{*4}と簡略化しており, ノード R^k からノード R^l へといったアークは考えないものとする. ただし, 各故障 R^k は観測イベント系列 \mathbf{E}^k について条件付き独立であるため, 各故障同士は完全に因果関係がないわけではないことに注意されたい. このようにして, システムの特性や物理的影響などの事前知識や専門家の経験的知識からネットワーク構造を構築する. 本研究ではシステムの制御則から故障と観測系列の因果関係を考慮し, ネットワーク構造を構築する.

以下, Fig. 2.4 のようなネットワーク構造を構築した場合の確率推論法に関して説明する. まず, 構築したネットワーク構造から全ての確率変数に関する同時確率が一意に決定され, Fig. 2.4 の場合は式 (2.8) のように表現できる.

$$P(R^1, R^2, \mathbf{E}^1, \mathbf{E}^2) = P(R^1)P(\mathbf{E}^1|R^1, R^2)P(R^2)P(\mathbf{E}^2|R^2). \quad (2.8)$$

BN を用いた確率推論は, ベイズの定理や周辺化により行われる. 例えば, 全サブシステムに関してイベント系列 ($\mathbf{E}^1, \mathbf{E}^2$) が観測された場合, サブシステム 1 においてある故障

^{*4} これは各故障 R^k の間にアークが存在しないことを意味しており, 各サブシステムの故障は互いに別の原因によって生起するということを意味している. もし因果関係を見逃さない場合はそれらのサブシステムを併合して取り扱うことによって表現することができる.

r_α^1 が生じている確率は以下の様に計算できる.

$$\begin{aligned}
 & P(R^1 = r_\alpha^1 | \mathbf{E}^1 = \mathbf{e}^1, \mathbf{E}^2 = \mathbf{e}^2) \\
 &= \frac{P(R^1 = r_\alpha^1, \mathbf{E}^1 = \mathbf{e}^1, \mathbf{E}^2 = \mathbf{e}^2)}{P(\mathbf{E}^1 = \mathbf{e}^1, \mathbf{E}^2 = \mathbf{e}^2)} \\
 &= \frac{1}{Z} \left\{ P(R^1 = r_\alpha^1) \sum_{R^2} \{ P(\mathbf{E}^1 = \mathbf{e}^1 | R^1 = r_\alpha^1, R^2) P(R^2) P(\mathbf{E}^2 = \mathbf{e}^2 | R^2) \} \right\}. \quad (2.9)
 \end{aligned}$$

ただし、式中の \sum_{R^2} とは確率変数 R^2 の全ての実現値に関して和をとるという意味である。また、 Z は正規化項であり、式 (2.10) のように表現できる。

$$Z = \sum_{R^1} \sum_{R^2} \{ P(R^1) P(\mathbf{E}^1 = \mathbf{e}^1 | R^1, R^2) P(R^2) P(\mathbf{E}^2 = \mathbf{e}^2 | R^2) \}. \quad (2.10)$$

ここで、事前確率 $P(R^k = r_\kappa^k)$ はサブシステム k で故障 r_κ^k が発生する確率であり、過去のシステム運用に基づく故障発生統計的確率や専門家の先験的知識により決まる値である。また、条件付き確率 $P(E^k | R^1, R^2)$ は、各局所診断器における診断結果を利用して算出することができる。局所診断器は、事前に想定した故障が生じた場合の各サブシステムで生じるイベント間の関係を時間付きマルコフモデルを用いて表現することで設計され、診断時に観測されたイベント系列と事前に想定した故障が生じた場合に観測されるイベント系列とを比較することで局所的な診断を行う。

2.2.3 故障診断手順および故障診断法

Fig. 2.5 は本論文における分散型故障診断の流れを示したフローチャートである。

システム全体を幾つかのサブシステムに分割し、分割した各サブシステム毎にイベント系列を観測する。観測したイベント系列を基に、設計した局所診断器を用いて局所的な故障診断 $P(R | \mathbf{E}^i)$ の計算を行い、その局所的な診断結果から BN の構成要素の一つである条件付き確率を計算する。そして、局所診断器の計算結果と構築したネットワーク構造により、故障候補毎に故障生起確率式 (2.9) を計算する。最後に、算出した故障生起確率の中で最も値が大きいものを選び、その時に想定していた故障を診断結果とする。例えば、サブシステム k に関して、 $\{r_0^k, \dots, r_{n_k}^k\}$ の故障候補が存在する場合、以下の式 (2.11) を満たす故障 r_i^k を診断結果とする。

$$\begin{aligned}
 \text{DiagnosisResult} = \arg \max_{r_i^k} & \left(P(R^k = r_0^k | \mathbf{E}^1 = \mathbf{e}^1, \dots, \mathbf{E}^m = \mathbf{e}^m), \right. \\
 & \left. \dots, P(R^k = r_{n_k}^k | \mathbf{E}^1 = \mathbf{e}^1, \dots, \mathbf{E}^m = \mathbf{e}^m) \right). \quad (2.11)
 \end{aligned}$$

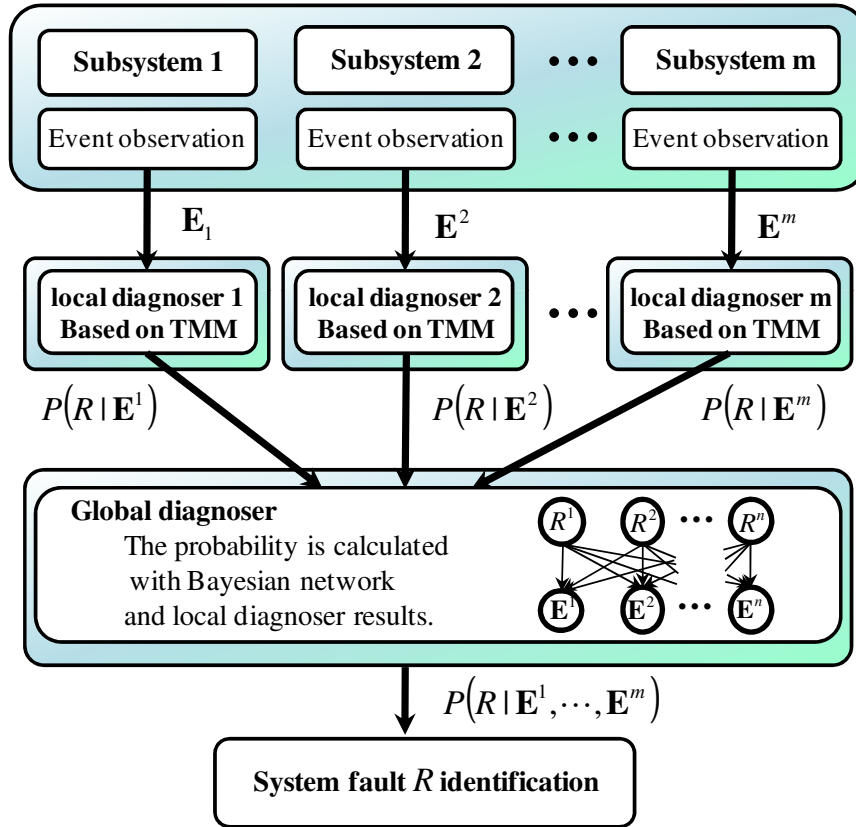


Fig.2.5 Diagram of decentralized fault diagnosis procedure

2.3 準マルコフ過程を用いた故障診断

本節では、準マルコフ過程を用いた故障診断について説明する。準マルコフ過程である時間付きマルコフモデルは状態の滞在時間に応じて、状態遷移確率変動する。時間付きマルコフモデルを用いて時間付きイベント系列を表現するには、イベント間の生起間隔に対する確率密度関数を推定する必要がある。本論文では確率密度関数をヒストグラムを用いて表現し、AICによって最適な帯幅を決定し、ゼロ頻度問題に対応するためにディスカウンティングを行う。

時間付きマルコフモデル (Timed Markov Model: 以下, TMM) とは、状態遷移確率が時間に依存するマルコフモデルである。つまり、ある状態から次の状態へと遷移するまでの滞在時間に応じて、遷移確率が変化する。TMMの詳細については文献 [18] を参照されたい。以下では、TMMに基づいた状態間の関係を表現する方法について説明する。

2.3.1 確率密度関数の定義

各状態の滞在時間 τ は連続量であるため、状態遷移確率は確率密度で表される。

TMM における状態を現時刻から直近に発火したセンサのラベル v_n^k とすることにより、時間付きイベント系列を TMM の挙動として表わすことができる [18]。以下では故障 r における、TMM の性質を示す確率密度を定義する。なお本節では一つのサブシステムについて議論するので、表記の簡単化のためにサブシステムの添え字 k は省略する。

定義 1 時間付き遷移確率 (Timed Transition Probability: TTP) とは、故障が r であり、センサラベル v_n の次に v_{n+1} が観測され、かつその滞在時間が τ_{n+1} であるときの同時確率密度

$$p(V_{n+1} = v_{n+1}, T_{n+1} = \tau_{n+1} | V_n = v_n, R = r) \quad (2.12)$$

である。ただし V_{n+1}, T_{n+1} は v_{n+1}, τ_{n+1} の確率変数である。この TTP は次に遷移するイベントラベル v_{n+1} と滞在時間 τ_{n+1} の関数であり、Fig. 3.7 のような離散変数と連続変数の同時確率となる。

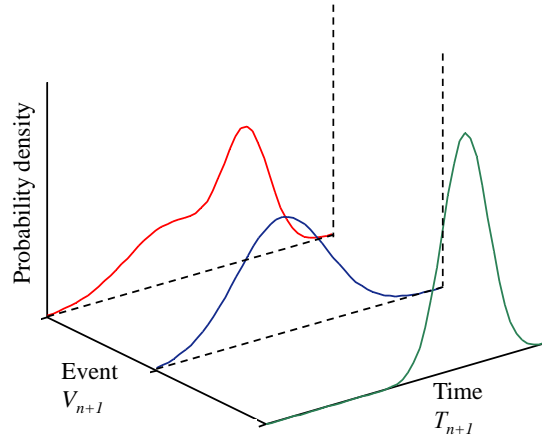


Fig.2.6 The example of timed transition probability

連続して遷移する2つの状態毎に確率密度関数を、全ての状態に関して確率分布関数を作成することで、故障の組が r である状況の下でシステムで生じる状態間の関係を表現することが可能となる。本論文では、学習データを統計的に処理したヒストグラムを用い、さらにゼロ頻度問題の対策としてディスカウンティング (ラプラス法) を行った。ディスカウンティングの詳細については2.3.4節を参照されたい。

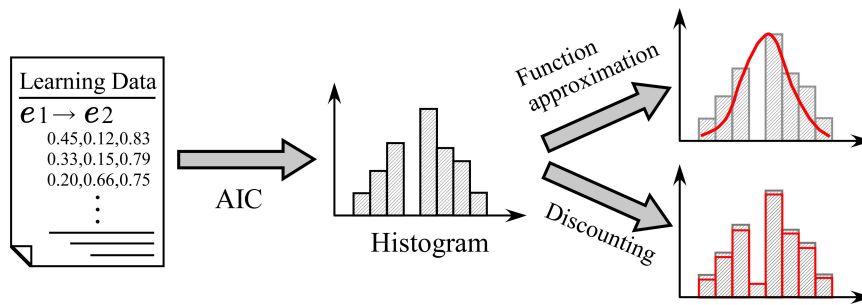


Fig.2.7 Procedure of estimating a probability density function

2.3.2 確率密度関数の推定

前節で示したように、TMM を用いて各サブシステムで生じるイベント間の関係を表現するには、イベント生起間隔に関する確率密度関数を事前に推定する必要がある。その手順を Fig. 2.7 に示す。まず、正常な場合および事前に想定した故障を意図的に起こした状況のもとで、イベント生起間隔のデータ (以下、学習データ) を取得する。ただし、実際の工場などでは、過去に故障が生じた時のデータを蓄積しておくことでこれに対応でき、意図的に故障を起こしてデータを取る必要は無いと考えられる。次に、取得した学習データからヒストグラムを作成する。この時、ヒストグラムの階級幅は AIC を基準として決定したものを用いる。最後にゼロ頻度問題の対策 (関数近似、ディスカウンティングなど) を施し、確率密度関数として局所的な故障診断に利用する。

2.3.3 AIC に基づくヒストグラムの階級幅決定法

ヒストグラム法は最も簡単であるノンパラメトリックな確率密度関数推定手法である。しかし、確率密度関数推定手法としてヒストグラム法を用いる場合、非常に扱いやすい手法である反面、ヒストグラムの階級幅 (Bin 幅) によってその形状が大きく変化するという問題点を有していることに注意しなければならない。本論文では確率密度関数、つまりヒストグラムの形状が局所的な故障診断の精度、しいては大域的な故障診断の精度にも大きく影響を及ぼすため、得られたデータの背後の母集団分布の形状を的確に表現しているヒストグラムを推定することが重要となる。

“Sturges’s Rule”^{*5}が統計学ソフトなどで一般的に採用されている階級数決定則ではあるが、母集団分布が二項分布から遠ざかると推定精度が悪化するという問題点を指摘されているように、その結果はあくまでも参考程度でしかない。そこで、本論文ではモデル選択基準として頻繁に利用されている情報量基準の一つである AIC(Akaike’s

^{*5} (階級数) = $1 + \log_2(\text{観測度数})$
階級幅は階級数から算出可能。

Information Criterion) *6に基づいた手法で階級幅を決定し、ヒストグラムを作成する。

以下、本論文で採用している“AIC に基づくヒストグラムの階級幅決定法 [51]”に関して説明する。この手法では、ヒストグラムを多項分布とみなし、ヒストグラムの階級数、各階級の確率と観測度数 (あるヒストグラムを作成するために得られたデータ数) を用いて、その対数尤度の最尤推定量から AIC の値を計算する。そして、AIC の値が最も小さいものが最適なヒストグラムであると判断する。

まず、階級数をできる限り多くしたヒストグラムを作成し、これを初期ヒストグラムと呼ぶこととする (Fig. 2.8)。また、初期ヒストグラムにおける階級数を c 、各階級の確率を $p(i)$ 、度数を $n(i)$ ($i = 1, \dots, c$) とする。次に、階級数を変更して新たなヒストグラムを作成する (Fig. 2.9)。両端の階級だけを不等間隔、中央部を等間隔にするヒストグラムを作成するので、両端の初期ヒストグラムにおける左端、右端の階級数を q_l, q_r 、中央部では r 個ずつの階級のプーリング*7を試みる。そのヒストグラムに対応する多項分布のモデルを $\text{MODEL}(q_l, r, q_r)$ とすると、 $\text{MODEL}(q_l, r, q_r)$ は式 (2.13) のように表現できる。ただし、 $\theta(j)$ は新たに作成したヒストグラムにおける各階級の確率である。

$$\begin{aligned} p(1) &= p(2) = \dots = p(q_l) = \theta(1), \\ p(q_l + (j-2)r + 1) &= \dots = p(q_l + (j-1)r) = \theta(j) \quad (j = 2, \dots, \frac{c - q_l - q_r}{r} + 1), \\ p(c - q_l + 1) &= \dots = p(c) = \theta\left(\frac{c - q_l - q_r}{r} + 2\right). \end{aligned} \quad (2.13)$$

また、このモデルは $\sum_{i=1}^c p(i) = 1$ より、次の制約を持つ。

$$q\theta(1) + r \sum_j \theta(j) + q\theta\left(\frac{c - q_l - q_r}{r} + 2\right) = 1. \quad (2.14)$$

$\theta(j)$ の最尤推定量を求めることにより、 $\text{MODEL}(q_l, r, q_r)$ の AIC は式 (2.15) のように与えられる。ただし、全ての MODEL に共通な定数部分は省略してある。

$$\begin{aligned} &\text{AIC}(q_l, r, q_r) \\ &= -2 \left\{ \left(\sum_{i=1}^{q_l} n(i) \right) \log \left(\sum_{i=1}^{q_l} \frac{n(i)}{q_l n} \right) + \sum_{j=2}^{(c - q_l - q_r)/r + 1} \left\{ \left(\sum_{i=q_l + (j-2)r + 1}^{q_l + (j-1)r} n(i) \right) \log \left(\sum_{i=q_l + (j-2)r + 1}^{q_l + (j-1)r} \frac{n(i)}{r n} \right) \right\} \right. \\ &\quad \left. + \left(\sum_{i=c - q_l + 1}^c n(i) \right) \log \left(\sum_{i=c - q_l + 1}^c \frac{n(i)}{q_r n} \right) \right\} + 2 \left(\frac{c - q_l - q_r}{r} + 1 \right). \end{aligned} \quad (2.15)$$

*6 AIC は以下の式で定義されている。

$\text{AIC} = -2 \times (\text{モデルの最大対数尤度}) + 2 \times (\text{モデルの自由パラメータ数})$

AIC の詳細に関しては文献 [51] を、モデル選択における AIC の概念に関しては文献 [52] を参照されたい。直感的には、「AIC が最小なモデルとは、データの適合度とモデルの複雑度のバランスが最もとれているモデルである」と考えることができる。

*7 複数の階級を一つに統合すること。つまりこの例において、中央部では初期ヒストグラムの r 個の階級を統合し、新たなヒストグラムの階級の一つとする。

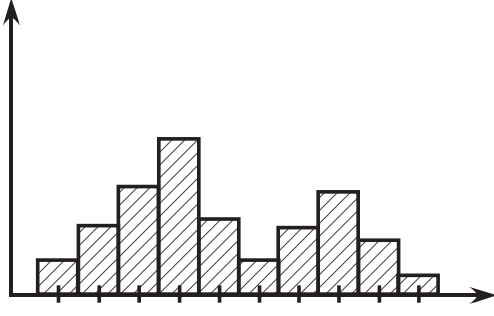


Fig.2.8 Example before integrating histogram

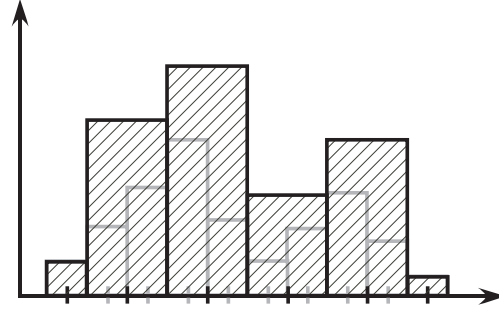


Fig.2.9 Example after integrating histogram

そして、プーリングの結果作られたヒストグラムの度数と階級数をそれぞれ $n'(i)$, c' とすると、 $\text{MODEL}(q_l, r, q_r)$ の AIC は式 (2.16) のように再表現できる.

$$\text{AIC}(q_l, r, q_r) = -2 \left\{ n'(1) \log \frac{n'(1)}{q_l n} + \sum_{j=2}^{c'-1} n'(j) \log \frac{n'(j)}{r n} + n'(c') \log \frac{n'(c')}{q_r n} \right\} + 2(c' - 1). \quad (2.16)$$

式 (2.16) からわかるように、 r と q_l, q_r の値さえ決めれば、プーリング後のヒストグラムだけから AIC の値を求めることができる. よって、 r と q_l, q_r の値を変化させていき、式 (2.16) から求められる AIC の値を比較していくことで、データの背後の母集団分布を最も的確に表現しているヒストグラムの階級数が得られる.

2.3.4 ゼロ頻度問題への対策

前項で説明した方法により確率密度関数 (ヒストグラム) を推定することができるが、一般に、単純に得られた学習データのみから確率密度関数を作成すると、学習データに現れなかった部分の確率値が 0 となってしまう. これは偶然学習データ中に出現しなかっただけか、学習データの背後の母集団においても出現しないのか判断することができない. 局所的故障診断アルゴリズムでは確率密度関数の値の積算を行うことから、確率密度関数の値が 0 となる時にイベントが生じると局所的な診断結果の値も 0 となるため、真の母集団分布においても確率値が 0 か否かといった問題が故障診断の精度に大きく係わる. このような問題をゼロ頻度問題 (Zero Probability Problem) という.

ゼロ頻度問題の代表的な解決手法として関数近似とディスカウンティング (Discounting) が挙げられる. ヒストグラムのように離散の確率密度関数を連続関数で近似すると、確率値が 0 となる箇所および離散値間を補間することができ、結果的にゼロ頻度問題を解決することが可能となる. 関数近似には、パラメトリック回帰や多層パーセプトロン

(Multi Layer Perceptron) を用いた手法, ノンパラメトリックな手法などが挙げられる. 特にノンパラメトリックな手法は事前に関数型を指定する必要が無く, 様々な分布に適用することが可能であることから有効な手法であり, 文献 [17], [53] では最大エントロピー原理 (Maximum Entropy Principle)[54] による確率密度関数推定手法を用いている. 一方, ディスカウンティングとは, データの出現回数を補正することであり, 単純にデータの出現回数を用いるのではなく, 出現回数を補正した値を使うことでゼロ頻度問題を解決する. 本論文では, 計算量の点で有利であるディスカウンティングによってゼロ頻度問題を解決することとし, その際に用いたラプラス法 (Laplas's Law) に関する具体例を以下に示す. 他のディスカウンティング手法に関しては文献 [55] を参照されたい.

ここでは, サブシステム k において, イベント e_i^k の次にイベント e_j^k が生じるまでの時間間隔に関するヒストグラムを例とし, 対象とするヒストグラムのパラメータを次のように定義する. 階級数を $C_{e_j^k e_i^k}$, 階級番号を $c_{e_j^k e_i^k} (\in \{1, \dots, C_{e_j^k e_i^k}\})$, $c_{e_j^k e_i^k}$ 番目の階級に含まれる度数を $n_{e_j^k e_i^k}(c_{e_j^k e_i^k})$, 階級幅を $h_{e_j^k e_i^k}$ とする. その時, 各階級の確率密度 $p_{e_j^k e_i^k}(c_{e_j^k e_i^k})$ は式 (2.17) の様にして計算する.

$$p_{e_j^k e_i^k}(c_{e_j^k e_i^k}) = \frac{n_{e_j^k e_i^k}(c_{e_j^k e_i^k})}{h_{e_j^k e_i^k} N_{e_i^k}}, \quad (2.17)$$

$$N_{e_i^k} = \sum_{e_j^k \in \mathcal{E}^k} \sum_{c_{e_j^k e_i^k}=1}^{C_{e_j^k e_i^k}} n_{e_j^k e_i^k}(c_{e_j^k e_i^k}). \quad (2.18)$$

ただし, 式 (2.17) 中の $N_{e_i^k}$ は式 (2.18) のように表記することができ, イベント e_i^k の次に何らかのイベントが生じた場合の出現総数に相当する.

データの出現回数に一律に一定の値を加える単純な方法を加算法といい, 特に出現回数に 1 を加える方法がラプラス法と呼ばれている. すなわち, ラプラス法では, 各階級の確率密度 $p_{e_j^k e_i^k}(c_{e_j^k e_i^k})$ を式 (2.19) の様に修正し, 式中の $N'_{e_i^k}$ はイベント e_i^k の次に何らかのイベントが生じた場合の修正後の出現総数に相当する.

$$p'_{e_j^k e_i^k}(c_{e_j^k e_i^k}) = \frac{n_{e_j^k e_i^k}(c_{e_j^k e_i^k}) + 1}{h_{e_j^k e_i^k} (N'_{e_i^k} + V)}, \quad (2.19)$$

$$N'_{e_i^k} = \sum_{e_j^k \in \mathcal{E}^k} \sum_{c_{e_j^k e_i^k}=1}^{C_{e_j^k e_i^k}} (n_{e_j^k e_i^k}(c_{e_j^k e_i^k}) + 1). \quad (2.20)$$

ただし, V は修正前のヒストグラムと矛盾しないことを示す以下の関係を満たすために導入された定数である.

$$\sum_{c_{e_j^k e_i^k}=1}^{C_{e_j^k e_i^k}} p_{e_j^k e_i^k}(c_{e_j^k e_i^k}) = \sum_{c_{e_j^k e_i^k}=1}^{C_{e_j^k e_i^k}} p'_{e_j^k e_i^k}(c_{e_j^k e_i^k}). \quad (2.21)$$

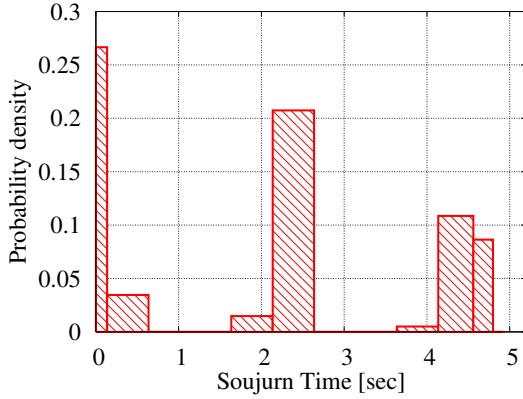


Fig.2.10 Example of discounting result (Laplas's law) before discounting

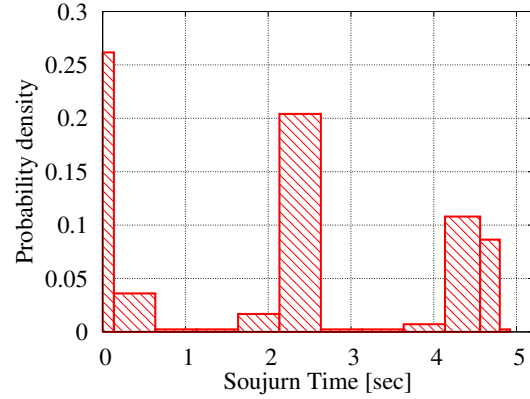


Fig.2.11 Example of discounting result (Laplas's law) after discounting

この操作によって、ヒストグラムを平滑化し、確率値が0の部分に微少な値を割り当てることが可能となる。デイスカウンティング結果の例を Fig. 2.10, 2.11 に示す。デイスカウンティングを行うことで、処理前のヒストグラムにおいて確率値が0の部分に、微少な確率値 (例においては 0.0024) の付与が実現されていることを Fig. 2.11 から確認できる。

2.3.5 推定すべき確率密度関数の数

提案手法における計算量の大半は確率密度関数の数と関連がある。そこで本論文では、故障診断に要する計算量を確率密度関数の数で評価することとし、本節では推定すべき確率密度関数の数について考察する。

本論文で提案する分散型故障診断法 (局所的故障診断) に必要な確率密度関数の数を $N_{pdf_decentralized}$ 、比較対象である文献 [17] で提案した集中型故障診断法に必要な確率密度関数の数を $N_{pdf_centralized}$ とすると確率密度関数の数はそれぞれ故障の組み合わせとイベントの組み合わせの積として表現される。

$$N_{pdf_decentralized} = \sum_{k=1}^n \left\{ (n_{\varepsilon^k})^2 n_{\mathcal{R}^k} \right\}, \quad (2.22)$$

$$N_{pdf_centralized} = (n_{\varepsilon})^2 n_{\mathcal{R}}, \quad \mathcal{E} = \bigcup_{k=1}^n \mathcal{E}^k. \quad (2.23)$$

ただし、 n_{ε^k} はサブシステム k で観測されるイベントの種類数、 $n_{\mathcal{R}^k}$ はサブシステム k で想定した故障の組の総数、 n_{ε} はシステム全体で観測されるイベントの種類数、 $n_{\mathcal{R}}$ はシステム全体で想定した故障の組の総数をそれぞれ表している。集中型故障診断法では各サブシステムの故障の全てに対して組み合わせを考慮する必要がある。一方、分散型故障診断法では各故障はBNにおいて互いに条件付き独立としているため、各サブシステムの観

測イベント系列と因果関係にある故障のみを考慮すればよい。

システム全体をサブシステムに分割することで、一部の積を和に変換することができるため、確率密度関数の数の大幅な削減が可能となる。また、想定する故障の組および観測されるイベントの集合も分割することができ、想定する故障の組数および観測されるイベントの種類数の減少も確率密度関数の数の削減の大きな要因と言える。

2.3.6 局所的故障診断方法

局所診断における目的は観測イベント系列 $\mathbf{e}^k = (e_{1:n}^k, \tau^k)$ に基づいた故障確率

$$\begin{aligned} P(R^k = r^k | \mathbf{E}^k = \mathbf{e}^k) \\ = P(R^k = r^k | V_{1:n}^k = v_{1:n}^k, T_{1:n}^k = \tau_{1:n}^k, [T_{n+1}^k > \tau^k] = \text{True}) \end{aligned} \quad (2.24)$$

を求めることである。ここで、 $[C]$ は関係式 C が成立するか否かを表す確率変数であり、実現値として $\{\text{True}, \text{False}\}$ を取るものとする。これは、時間付きイベント系列 $e_{1:n}^k = (v_{1:n}^k, \tau_{1:n}^k)$ が得られ、かつ τ^k までセンサの発火がないとき、システムの故障状況 r^k である場合の確率である。この確率を計算するために文献 [18] では、次の再帰アルゴリズムを用いて計算を行っている。ただし、本論文の記号等の表記は文献 [18] の表記とは異なっている。また以下では、式の表記の簡略化のために、例えば $V_{1:n}^k = v_{1:n}^k$ を $V_{1:n}^k$ のように、 $[T_{n+1}^k > \tau^k] = \text{True}$ を $[T_{n+1}^k > \tau^k]$ のように、実現値を省略して表現することとする。

時間間隔 τ^k の間センサが発火しない場合を考えると、式 (2.24) はベイズの定理を用いて次のように更新される。

$$\begin{aligned} P(R^k = r^k | \mathbf{E}^k = \mathbf{e}^k) \\ = P(R^k | V_{1:n}^k, T_{1:n}^k, [T_{n+1}^k > \tau^k]) \\ = \frac{P(R^k | V_{1:n}^k, T_{1:n}^k) P([T_{n+1}^k > \tau^k] | V_{1:n}^k, T_{1:n}^k, R^k)}{\sum_{R^k} P(R^k | V_{1:n}^k, T_{1:n}^k) P([T_{n+1}^k > \tau^k] | V_{1:n}^k, T_{1:n}^k, R^k)} \\ = \frac{P(R^k | V_{1:n}^k, T_{1:n}^k) P([T_{n+1}^k > \tau^k] | V_n^k, R^k)}{\sum_{R^k} P(R^k | V_{1:n}^k, T_{1:n}^k) P([T_{n+1}^k > \tau^k] | V_n^k, R^k)}. \end{aligned} \quad (2.25)$$

ただし、時間イベント系列は準マルコフ過程によってモデル化しているので、確率 $P([T_{n+1}^k > \tau^k] | V_{1:n}^k, T_{1:n}^k, R^k)$ は $P([T_{n+1}^k > \tau^k] | V_n^k, R^k)$ となる。ここで、確率 $P([T_{n+1}^k > \tau^k] | V_n^k, R^k)$ は TTP を用いて

$$\begin{aligned} P([T_{n+1}^k > \tau^k] | V_n^k, R^k) \\ = 1 - \sum_{V_{n+1}^k} \int_0^{\tau^k} p(V_{n+1}^k, T_{n+1}^k = \tau' | V_n^k, R^k) d\tau' \end{aligned} \quad (2.26)$$

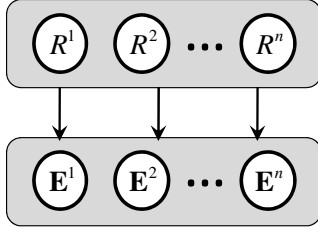


Fig.2.12 Bipartite Bayesian Network for fault diagnosis

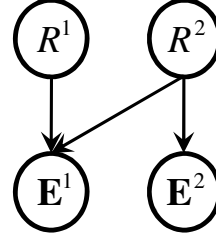


Fig.2.13 Example of Bayesian Network

と計算することができる． $P(R^k)$ は故障の事前確率であり，過去のシステム運用に基づく統計的確率や専門家の先験的知識などから決定される．しかし，本論文ではシステムの故障に関する事前情報はないものとして，事前確率は等確率を割り振る．つまり

$$P(R^k) = \frac{1}{n_k} \quad (2.27)$$

とした．また，運用情報から故障の事前確率が推定できるようになった場合，上記の事前確率を新たに再設定することにより，診断精度を高めることができる．

$T_{n+1}^k = \tau^k$ となったとき，つまり新たに時間付きイベント $(v_{n+1}^k, \tau_{n+1}^k)$ が観測されたとき，式 (2.25) の確率 $P(R^k | V_{1:n}^k, T_{1:n}^k)$ は以下のように更新される．

$$\begin{aligned} & P(R^k | V_{1:n+1}^k, T_{1:n+1}^k) \\ &= \frac{P(R^k | V_{1:n}^k, T_{1:n}^k) p(V_{n+1}^k, T_{n+1}^k | V_n^k, R^k)}{\sum_{R^k} P(R^k | V_{1:n}^k, T_{1:n}^k) p(V_{n+1}^k, T_{n+1}^k | V_n^k, R^k)}. \end{aligned} \quad (2.28)$$

時間付きイベント (v_1^k, τ_1^k) が観測されたときは以下のように更新される．

$$P(R^k | V_1^k, T_1^k) = \frac{P(R^k) p(V_1^k, T_1^k | V_0^k, R^k)}{\sum_{R^k} P(R^k) p(V_1^k, T_1^k | V_0^k, R^k)}. \quad (2.29)$$

v_0^k は初回に発火したセンサラベルであり， v_0^k により TMM における初期状態を決定する．

2.4 ベイジアンネットワークを用いた大域的診断器

BN は複数の確率変数間の定性的な因果関係をネットワーク構造によって表し，個々の変数間の定量的な関係を条件付き確率で表したグラフィカルモデルである．本節では BN を用いた大域故障診断手法 [32] について述べる．

各サブシステムの故障と観測イベント系列の確率変数 R^k, \mathbf{E}^k をそれぞれノードとし，ネットワークの構造を二部グラフとする (Fig. 2.12)．つまり，各サブシステムの故障は互いに独立に発生し，異なるサブシステムの観測イベント系列は故障に関して条件付き独立であるとする [32]．なお，各サブシステムはこれらの独立関係を満たすように分割され

なければならない。BN の構造により診断精度と計算量が大きく左右されるため、適切な BN の構造が必要となる。BN の構造の構築についての詳細は 2.5.1 節で述べる。

2.4.1 ベイジアンネットワークの条件付き確率計算

システム全体の故障確率は局所診断器の出力結果と BN を用いて計算される。BN の構造を陽に表現するために、 \mathbf{E}^k へアークが引かれている故障確率変数を並べたベクトルを $\mathbf{R}^{\mathbf{E}^k}$ と定義する。このベクトルを用いることにより、 $P(\mathbf{E}^k | R^1, \dots, R^m)$ は $P(\mathbf{E}^k | \mathbf{R}^{\mathbf{E}^k})$ と表現できる。なお、Fig. 2.13 の例では $\mathbf{R}^{\mathbf{E}^1} = (R^1, R^2)$, $\mathbf{R}^{\mathbf{E}^2} = (R^2)$ となる。

大域診断器では故障確率

$$P(R^1, \dots, R^m | \mathbf{E}^1, \dots, \mathbf{E}^m) \quad (2.30)$$

を求めることが目的となる。この確率は BN の構造とベイズの定理を用いて次のように計算できる。

$$\begin{aligned} & P(R^1, \dots, R^m | \mathbf{E}^1, \dots, \mathbf{E}^m) \\ &= \frac{P(\mathbf{E}^1, \dots, \mathbf{E}^m | R^1, \dots, R^m) P(R^1, \dots, R^m)}{P(\mathbf{E}^1, \dots, \mathbf{E}^m)} \\ &= \frac{\prod_{k=1}^m P(\mathbf{E}^k | \mathbf{R}^{\mathbf{E}^k}) P(R^k)}{P(\mathbf{E}^1, \dots, \mathbf{E}^m)}. \end{aligned} \quad (2.31)$$

確率 $P(\mathbf{E}^k | \mathbf{R}^{\mathbf{E}^k})$ はベイズの定理から

$$P(\mathbf{E}^k | \mathbf{R}^{\mathbf{E}^k}) = \frac{P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k) P(\mathbf{E}^k)}{P(\mathbf{R}^{\mathbf{E}^k})} \quad (2.32)$$

と計算できる。ここで $P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k)$ は局所診断器の結果 (2.25) そのものである。そして、式 (2.31) は式 (2.32) を用いて

$$\begin{aligned} & P(R^1, \dots, R^m | \mathbf{E}^1, \dots, \mathbf{E}^m) \\ &= \frac{\prod_{k=1}^m P(\mathbf{E}^k)}{P(\mathbf{E}^1, \dots, \mathbf{E}^m)} \times \prod_{k=1}^m \frac{P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k) P(R^k)}{P(\mathbf{R}^{\mathbf{E}^k})} \end{aligned} \quad (2.33)$$

と計算できる。ただし $P(\mathbf{E}^1, \dots, \mathbf{E}^m)$ は以下の正規化項である。

$$\begin{aligned} & P(\mathbf{E}^1, \dots, \mathbf{E}^m) \\ &= \sum_{R^1} \dots \sum_{R^m} P(\mathbf{E}^1, \dots, \mathbf{E}^m | R^1, \dots, R^m) P(R^1, \dots, R^m) \\ &= \sum_{R^1} \dots \sum_{R^m} \prod_{k=1}^m P(\mathbf{E}^k | \mathbf{R}^{\mathbf{E}^k}) P(R^k) \\ &= \prod_{k=1}^m P(\mathbf{E}^k) \times \sum_{R^1} \dots \sum_{R^m} \prod_{k=1}^m \frac{P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k) P(R^k)}{P(\mathbf{R}^{\mathbf{E}^k})}. \end{aligned} \quad (2.34)$$

式 (2.33) に式 (2.34) を代入すると, $\prod_{k=1}^m P(\mathbf{E}^k)$ の項がキャンセルされるため, 計算しなくてもよい. したがって

$$\begin{aligned}
 & P(R^1, \dots, R^m | \mathbf{E}^1, \dots, \mathbf{E}^m) \\
 &= \frac{\prod_{k=1}^m \frac{P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k) P(R^k)}{P(\mathbf{R}^{\mathbf{E}^k})}}{\sum_{R^1} \dots \sum_{R^m} \prod_{k=1}^m \frac{P(\mathbf{R}^{\mathbf{E}^k} | \mathbf{E}^k) P(R^k)}{P(\mathbf{R}^{\mathbf{E}^k})}} \quad (2.35)
 \end{aligned}$$

と計算できる.

2.5 ネットワーク構造学習

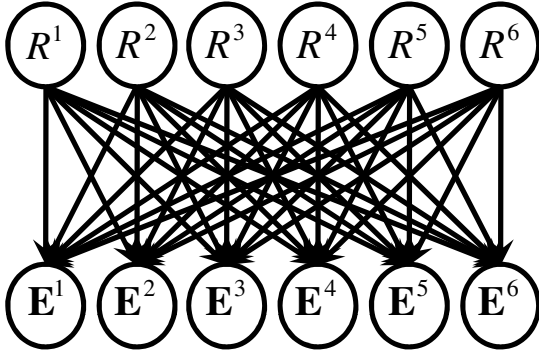


Fig.2.14 Bayesian Network Example 1

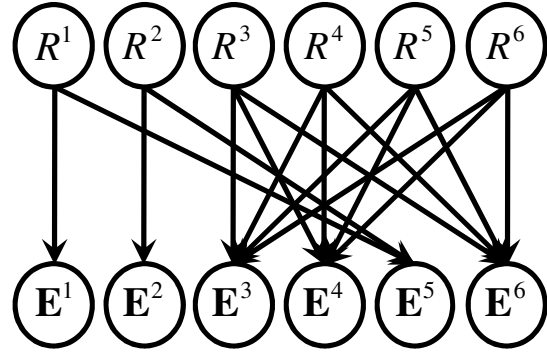


Fig.2.15 Bayesian Network Example 2

本手法における BN は Fig. 2.14, 2.15 のような, 故障状況をラベルとする離散変数 R^k と各サブシステムで観測されたイベント系列 \mathbf{E}^k をノードとする 2 部グラフで表現される. アークは各サブシステム間の故障の因果関係, つまり故障による観測イベントの変化を表している.

BN のアークが多い複雑な構造である程, 多くの因果関係を表していることになるので, Fig. 2.14 のような全ての因果関係を表現した構造が診断精度の良い構造となるが, より多くの因果関係を考慮しなければならないため計算量が増加してしまう. 逆にアークの少ない簡単な構造である場合は, 考慮すべき因果関係が少ないため計算量が少なくて済むが, 因果関係をうまく表現できていない場合は診断精度が落ちてしまう. これらのことから診断精度と計算量はトレードオフの関係であることが分かり, BN の構造によって変化する. またモデルである PDF は想定故障組合せ数だけ必要となるため, アークの数が少ない構造の方が少ない学習データ量で学習できる.

ここで問題となるのは因果関係を表現している重要なアークと因果関係を表現していないアークが存在することである. 重要なアークのみを選択し, 重要でないアークを削除することにより診断精度への影響を減らし, 効率よく計算量を削減できる.

2.5.1 KL 情報量を用いたネットワーク学習法

本節では運用データからアークの重要度を Kullback-Leibler 情報量 (以下 KL 情報量) を用いて評価する手法について述べる.

2.5.2 Kullback-Leibler 情報量

KL 情報量は, 確率論と情報理論における 2 つの確率分布 (真の確率分布 P とモデルの確率分布 Q) の差の尺度であり情報ダイバージェンス, 相対エントロピーなどとも呼ばれる. 離散確率分布の場合, KL 情報量は式 (2.36) のように表わされる.

$$KL(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)} \quad (2.36)$$

また連続確率分布の場合, 式 (2.37) のように表わされる.

$$KL(P||Q) = \int_{-\infty}^{\infty} P(x) \ln \frac{P(x)}{Q(x)} dx \quad (2.37)$$

KL 情報量は常に非負の値を取り, $P = Q$ のとき値が 0 となる. この性質はモデル Q の当てはまりのよさを表す尺度として用いることができ, 当てはまりがよいほど小さな値を持つ. このようなことから直感的に距離と表されることもあるが, 対称性がないため厳密には距離ではない.

2.5.3 アークの重要度評価法

BN は故障事例と観測イベント系列の因果関係を確率的に表現している. そのため, BN の構造は診断精度と計算量に大きな影響を与え, 適切に決める必要がある. 従来の手法ではネットワーク構造を決定するためにはシステムの特性や物理影響などの事前知識や専門家の経験的知識を必要とし, システムが大規模かつ複雑な場合はそれらの知識を正確に把握することは困難である. 本論文ではシステムの特性を表現している確率密度関数の差異に注目し, データに基づいた BN の構築法を提案する. BN の構造は診断精度と計算量に重要な影響を与えるため, 適切に決定する必要がある. BN の構造の学習における制約ベース学習では, 式 (2.32) に相当する, BN の各ノードに割り振られた条件付き確率を用いることが多い. しかしこの条件付きを求めるためには, 観測イベント系列の発生確率 $P(\mathbf{E}^k)$ を計算する必要がある. 観測イベント系列は時間という連続量を含んでおり, さらに系列長は時間付きイベントが得られるたびに増加する. そのため, $P(\mathbf{E}^k)$ を計算することは難しく, 条件付き確率 (2.32) を直接計算することは現実的ではない. そこで本論文では局所診断器のモデルである TMM に着目することにより, 間接的にノード間の独立性を検証する.

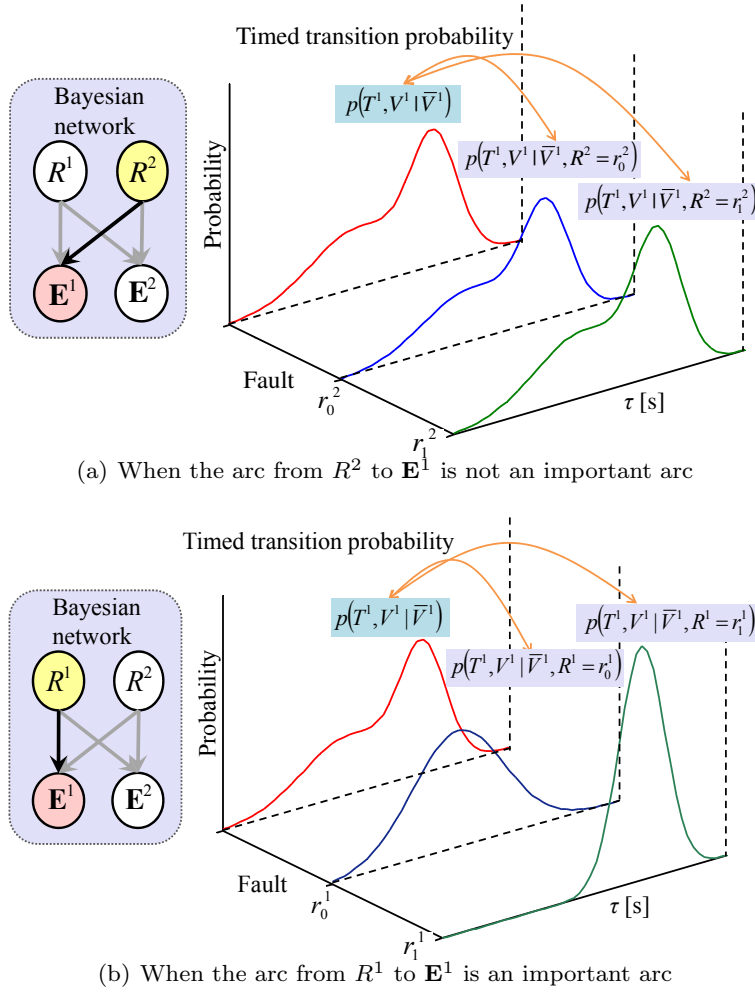


Fig.2.16 Example of the difference between TTPs

本問題設定においては BN の構造を二部グラフとして想定しているの、故障ノード R^i と観測ノード E^j 間の独立性を評価すればよい。そこで、TMM の挙動を表している TTP に着目して独立性を評価する。具体的には独立性を評価するために R^i から E^j へのアークの影響を TTP を用いて評価する。Figure 2.16 の例は故障ノード $R^i (i = 1, 2)$ から E^1 へのアークが存在する場合の TTP $p(V^1, T^1 | \bar{V}^1, R^i)$ と、アークが存在しない場合、つまり R^i について周辺化した TTP $p(V^1, T^1 | \bar{V}^1)$ を示している (ただし、 V^k, T^k はある時間付きイベントのセンサラベルと時間間隔の確率変数、 \bar{V}^k は V^k より 1 つ前に観測されたセンサラベルの確率変数である)。Figure 2.16(a) は R^2 から E^1 へのアークのあるなしに関わらず、TTP の変化が小さい様子を表している。一方、Fig. 2.16(b) の例では R^1 から E^1 へのアークが存在する場合に TTP が大きく変化している様子を示している。このような TTP の分布の差は、故障 R^i で条件付けたときの観測イベント系列 E^j が発生する確率の変化量を表しており、この分布の差が小さいほど R^i と E^j 間は独立であると判断できる。TTP の分布の差は KL 情報量を用いて、以下に示す重要度指標とし

Table 2.1 Set of sensor labels in each subsystem

Subsys.	event labels	Subsys.	event labels
1	$\{v_1, v_2, v_3\}$	4	$\{v_{10}, v_{11}, v_{12}\}$
2	$\{v_4, v_5, v_6\}$	5	$\{v_3, v_7, v_6, v_{10}\}$
3	$\{v_7, v_8, v_9\}$	6	$\{v_9, v_{12}, v_{13}, v_{14}\}$

て評価する.

$$V_{ij} = \max_{\bar{V}^j, R^i} KL(p(T^j, V^j | \bar{V}^j) || p(T^j, V^j | \bar{V}^j, R^i)). \quad (2.38)$$

ただし $KL(p||q)$ は次のとおりである.

$$KL(p||q) \equiv \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx. \quad (2.39)$$

重要度指標 V_{ij} はアークの重要度を表しており, 値が大きくなればなるほど, R^i から \mathbf{E}^j への寄与が大きいことを示す. 全てのアークに対し重要度指標 V_{ij} を計算し, ある閾値以上のアークを選択することにより BN の構造を構築する. これらのアルゴリズムを以下に示す.

Algorithm: Construction of the Bayesian network

Step 1 Set $j = 1, i = 1$.

Step 2 Estimate TTP $p(T^j, V^j | \bar{V}^j)$ from learning data.

Step 3 Estimate TTP $p(T^j, V^j | \bar{V}^j, R^i)$ from learning data.

Step 4 Calculate V_{ij} by using (2.38).

Step 5 If V_{ij} is larger than the threshold, select the arc from R^i to \mathbf{E}^j .

Step 6 If $i < m$, go to Step 3 with $i = i + 1$.

Step 7 Terminate the algorithm if $j = m$, else go to Step 2 with $j = j + 1, i = 1$.

2.6 検証実験

本節では提案した手法を Fig. 2.17 のような自動搬送ラインに適用する. 自動搬送ラインは Fig. 2.18 のような物理シミュレータ PhysX を用いて, CPU: Core2 Duo 3.00GHz, RAM: 4GB, HDD: 250GB の PC でシミュレーションした.

2.6.1 自動搬送ライン

この搬送システムは 6 機のコンベア (長さ 5 m), 2 機のクレーンと 14 個のセンサから構成され, ワークをアンロードステーションに搬送する. 時間付きイベントはワークがセンサを通過した場合に観測され, センサは Fig. 2.17 のように設置されている. この自動搬送ラインを 6 個のサブシステム (Lane1, Lane2, Lane3, Lane4, Crane1, Crane2) に分

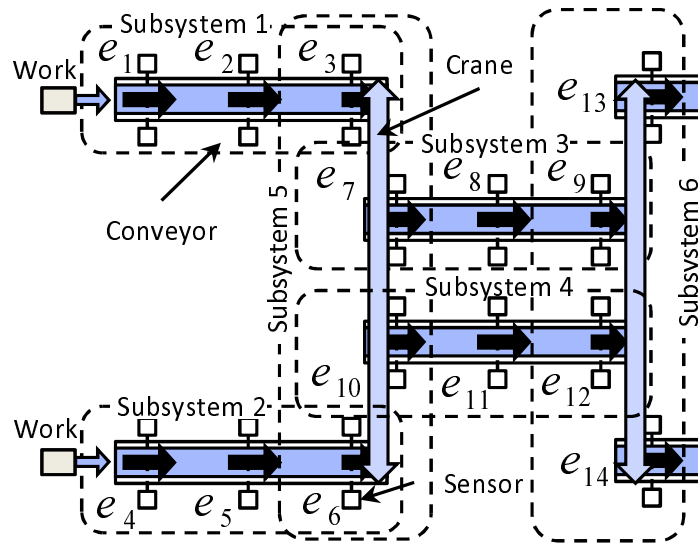


Fig.2.17 Diagram of the transfer line and definition of events

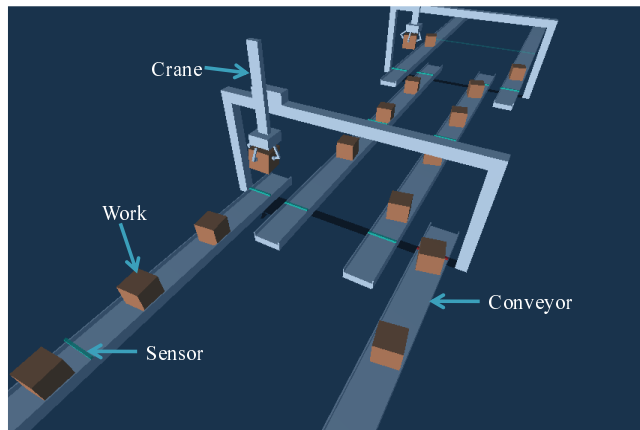


Fig.2.18 Simulator of automatic transfer line

割し，それらのサブシステムで観測される時間付きイベントのセンサラベルを Table 2.1 に示す．なお，サブシステムの分割の仕方は本論文の検討事項でないため既知とする．ワークの投入間隔は平均 1m，分散 0.03m^2 の正規分布とし，この分布に従い投入される．また，1 回の試行はワークを 50 個搬送することとする．

各サブシステムにおける観測イベント系列 \mathbf{E}^k は Table 2.1 のラベルをイベントとした時間付きイベント系列として観測される．例としてサブシステム 1 の実現値を考えると， $\mathbf{e}^k = ((v_1, 0), (v_2, 0.2), (v_1, 0.5), (v_3, 0.6), \dots)$ という時間付きのイベントの系列が観測されることになる．

Table 2.2 Evaluation value V_{ij} of arc from R^i to E^j

V_{ij}	R^1	R^2	R^3	R^4	R^5	R^6
E^1	0.209	0.002	0.002	0.002	0.054	0.002
E^2	0.001	0.205	0.001	0.001	0.054	0.001
E^3	0.004	0.005	2.110	0.264	0.421	0.317
E^4	0.005	0.005	0.234	2.365	0.458	0.357
E^5	0.394	0.379	0.001	0.001	0.043	0.000
E^6	0.072	0.064	0.380	0.441	0.275	0.903

2.6.2 故障候補

本例で扱う故障の候補として正常，速度低下，センサ故障を想定する．各ベルトコンベアの速度が 22cm/sec から 18cm/sec の間である場合を「正常 ($r_0^1, r_0^2, r_0^3, r_0^4$)」とし，18cm/sec から 15cm/sec の間である場合は「速度低下 ($r_1^1, r_1^2, r_1^3, r_1^4$)」であるとする．また Crane1, Crane2 についてはワーク搬送に要する時間が正常時より約 0.3sec 長くなる場合を「速度低下 (r_1^5, r_1^6)」であるとする．センサの故障はコンベアのみ想定し，中間のセンサが 50 % の確率で発火しない場合を「センサ故障 ($r_2^1, r_2^2, r_2^3, r_2^4$)」であるとする．同時にいくつかのサブシステムで発生する場合などを考慮し，システム全体で $3 \times 3 \times 3 \times 3 \times 2 \times 2 = 324$ 通りの故障組み合わせを故障候補として考える．

上記の環境において試行を行い，学習用データとして各故障ごとに 300 試行のデータを用意した．これらの学習データを用いて TMM の TTP を学習した．なお，TTP はヒストグラムとして近似した．

2.6.3 グラフ構造

Table 2.2 は Fig. 2.17 のシステムに対して式 (2.38) を用いて計算した重要度指標 V_{ij} である．アークの重要度指標 V_{ij} に閾値を設け，閾値より高いものは重要なアーク，低いものは重要でないアークとする．閾値はアークの選択基準であるため，重要度指標 V_{ij} のそれぞれの値と閾値の大小関係が変化しない限りにおいては選択されるアーク数は変化しない．それ故，あるアーク数の BN の構造を解析すると，大小関係が変化しない範囲の閾値に対して解析したことになる．実験対象としたシステムにおける，閾値と選択されるアーク数の関係を Fig. 2.20 に示す．閾値が小さいほど選択されるアーク数は増加する．また，選択されるアーク数は重要度指標 V_{ij} の値と閾値の大小関係のみに依存し，アーク数が変化することにより BN の構造が変化する．

ここで閾値を 0.2 とすると Fig. 2.19(a) のような提案構造 1 (Proposed structure 1) が

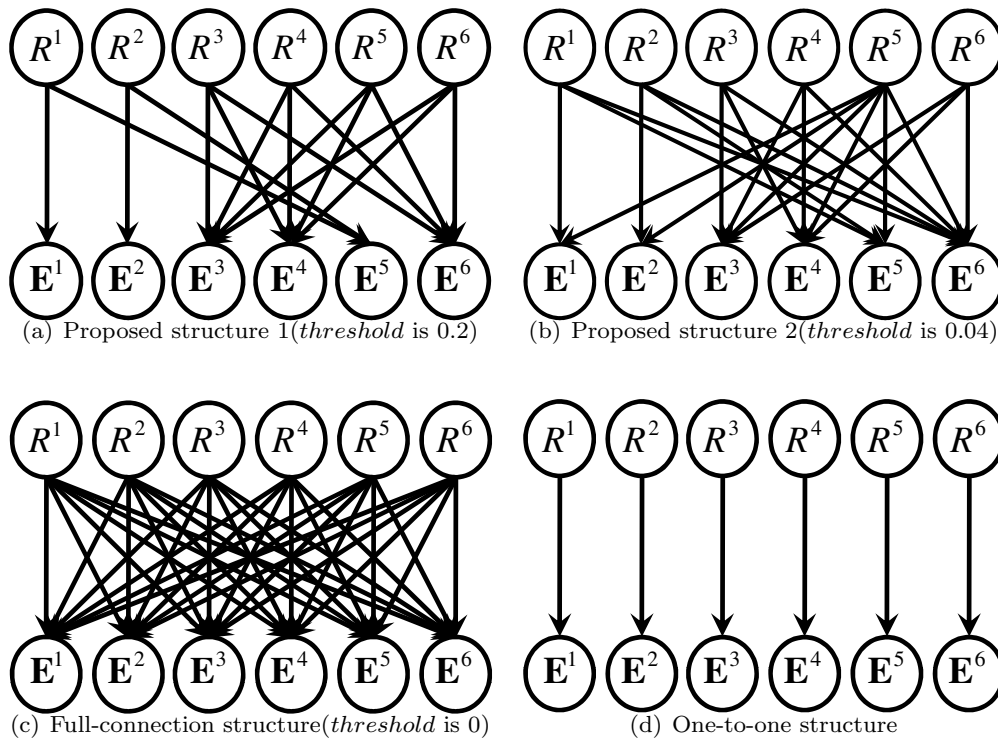


Fig.2.19 Bayesian Network structures

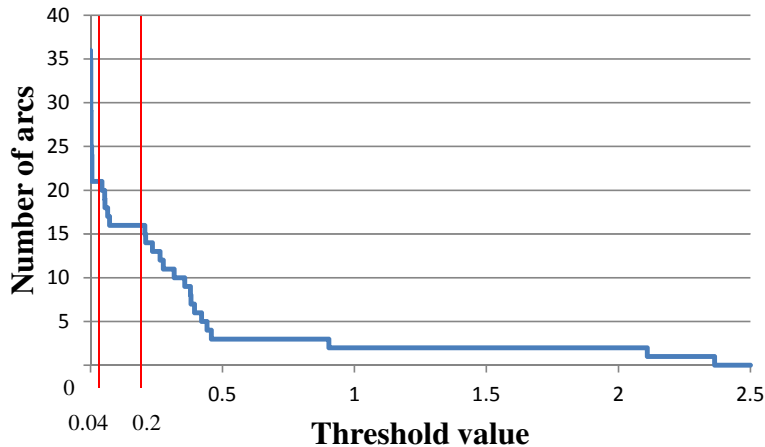


Fig.2.20 Relation between threshold value and the number of arcs

得られる。また、閾値を 0.04 に設定すると Fig. 2.19(b) のような提案構造 2 (Proposed structure 2) が得られる。Figure 2.19(c) の全結合構造 (Full-connection structure) は全てのサブシステム間の因果関係を表している BN であり、提案手法では閾値を 0 にすると得られる構造でもある。また、提案構造との比較のために、Figure 2.19(d) の一対構造 (One-to-one structure) を考える。一対構造はサブシステムの観測をそのサブシステムの故障の診断のみに用いるという構造であり、局所診断器の出力がそのまま大域診断器

Table 2.3 Result of diagnostic method

BN structure	Success rate	Undiagnosed rate	Wrong rate	Number of arcs	Computing time	Number of TTPs
Proposed structure 1	0.823	0.156	0.021	16	0.294 sec	414
Proposed structure 2	0.823	0.156	0.021	21	0.45 sec	1620
Full-connection structure	0.688	0.264	0.047	36	1.598 sec	6480
One-to-one structure	0.233	0.575	0.192	6	0.235 sec	52

の出力となる。

2.6.4 診断結果

診断データは学習データと同じ環境で試行を行い、各故障ごとに 10 試行、計 3240 試行のデータを診断した。また、各 BN の構造に対して同一の診断データを用いて診断した。なお計算にかかる時間を計測するため、オンラインの診断ではなく、オフラインで 1 試行のデータを一度に処理することにより診断結果を計算した。

それぞれの BN を用いて診断を行った診断成功率の結果を Table 2.3 に示す。なお、故障の同定は式 (2.35) の故障確率のエントロピーが 0 に近いと判断したときに同定できる。ただし、確率のエントロピーは以下の式で表され、今回はエントロピーが 0.025 より小さい場合を近いと判断した。

$$H = - \sum_X P(X) \log_2 P(X). \quad (2.40)$$

Table 2.3 における診断成功率 (Success rate) とはサブシステム全ての故障を同定でき、かつ同定された結果が全て正しい場合の割合である。未診断率 (Undiagnosed rate) はサブシステム全ての故障は同定できていないが、同定された結果が全て正しい場合の割合である。誤診断率 (Wrong rate) はサブシステム全ての故障を同定できているか、いないかにかかわらず、同定された結果に間違いを含んだ場合の割合である。計算時間 (Computing time) は 1 試行約 170 秒 (1700setps) のデータに対して、診断にかかる平均の時間であり、Fig. 2.18 の環境をシミュレートした PC と同じ PC で計算した。また TTP の数 (Number of TTPs) は診断に使用する各 BN の TTP の総計であり、アークの数の増加に対し TTP の数は指数的に増加する。また TTP の数 (Number of TTPs) は診断に使用する各 BN の TTP の総計である。なお TTP は各サブシステムごとに割り当てられており、故障と観測イベントラベルを条件部として持つため、TTP の数はそれらの

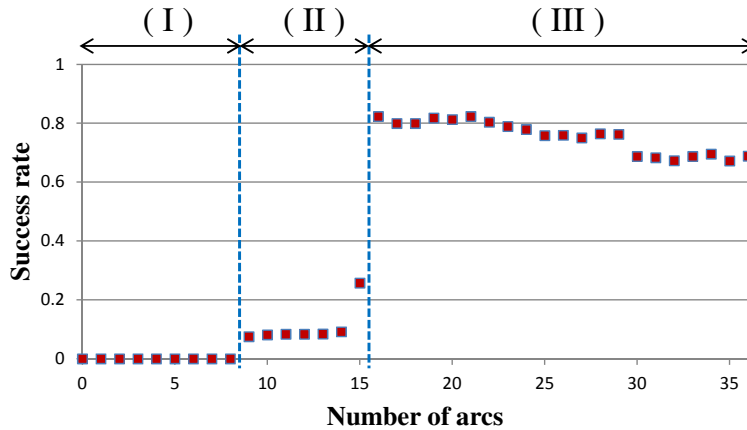


Fig.2.21 Success rate and the number of arcs

実現値の数の積で表現できる.

$$\sum_{i=1}^m n_r^i n_v^i \quad (2.41)$$

ただし, n_r^i はサブシステム i の BN 構造を考慮した故障 $\mathbf{R}^{\mathbf{E}^i}$ の実現値の数, n_v^i はサブシステム i におけるセンサラベルの数である. アークの数が増加すると, $\mathbf{R}^{\mathbf{E}^i}$ の組み合わせ数も増加するため, TTP の数は組み合わせ的に増加する.

計算時間に着目すると提案構造 1 は一対構造とほぼ同等の短さで計算できていることが分かる. 局所, 大域診断器では TTP の数が多ければ多いほど, 計算量が増えてしまう. そのため, 計算量の観点では提案構造 1 や一対構造などのアークが少ない構造の方が優れているといえる.

診断成功率は提案構造 1,2 が最も高くなった. これらのことから提案した手法を用いることにより, 計算量が少なく, かつ計算精度のよい BN を構築することができたことがわかる. 一方でアークが最も多い構造である全結合構造は提案構造より診断成功率は低くなった. アークが多い構造の方がより多くの故障の影響を考慮して診断できるため, 診断成功率は高くなると直感的には思われるが, Table 2.3 の結果では全結合構造の診断成功率が最も高くはなかった. これは学習すべき TTP の数が増加し, 一つの TTP に割り振られる学習データ量が減ってしまったためであると考えられる. 観測イベントは様々な要因から影響を受け, 時間的あるいはイベントの順序関係的にばらつきを含むため, 学習データ量の減少は過学習を生じさせる要因となる. そのため, BN の構造構築において閾値を下げることは, アークの多い詳細な BN の構造を与えるが, それと同時に TTP の数が多い BN の構造になってしまうため, 必ずしも良い結果とならないことを示している.

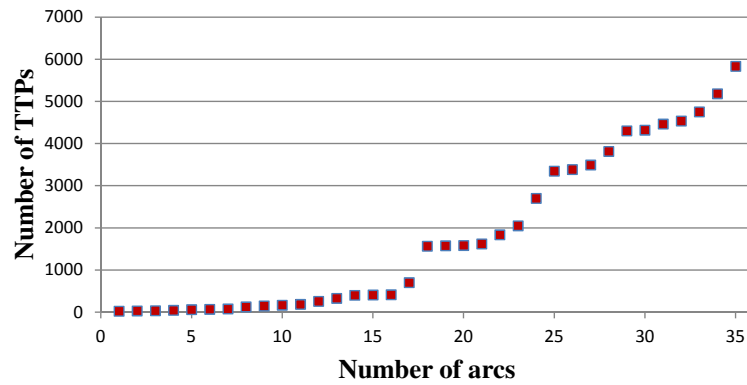


Fig.2.22 The number of TTPs and arcs

2.6.5 閾値による差異

本節では、提案手法において閾値を変化させて得ることができる、全ての BN の構造に関して比較して議論を行う。提案手法に基づき BN の構造を変化、つまり選択されるアークの数を変えたときの診断成功率との関係を Fig. 2.21 に、TTP の数との関係を Fig. 2.22 に示す。ただし、2.6.4 節と同じ条件で診断を行った。Figure 2.21 は (I), (II), (III) の 3 つの領域に分けることができる。(I) の領域はアークが繋がれていない故障ノードが存在する領域であり、アークのない故障ノードは診断されないため、診断成功率は必ず 0 となる。(II) は故障の影響を説明しているアーク、つまり必要なアークが全て選択されていない領域である。必要なアークが全て選択されていないため、診断成功率は低くなる。(III) は必要なアークが全て選択された領域であり、診断成功率は他の領域に比べ高くなる。しかし、必要なアークはすべて選択されているため、アークを増やしても冗長となることがある。

Figure 2.21 の (III) の領域では、診断成功率はなだらかな減少する傾向であったため、過学習が起きていると考えられる。これらの事柄からアークの増加はアークの多い詳細なモデルを得られると同時に、複雑さを増してしまうため必ずしも良い方向に働かないことが分かる。

提案した手法では、閾値が変化することで BN の構造が変化する。閾値を高くしすぎると重要なアークが選択されず、低くしすぎると過学習が起きてしまう。そのためバランスのとれた BN の構造となる閾値を選択する必要があるが、これは (III) の領域の最もアーク数が少ない構造となる閾値を選択すればよい。なお、本システムでは (III) の領域はアーク数 16 から 36 までの領域であり (Fig. 2.21 参照)、アーク数が最小である 16 であったときの BN は Fig. 2.19(a) であった。このような構造では必要なアークのみを選択し、学習すべき TTP の総数も減少するため、少ない学習データ量でも診断できる。

(III) におけるアーク数最小の BN となる閾値は重要度指標 V_{ij} の値だけではなく、シ

システムの故障要素やシステムの構造など様々な要因により変化するため、選択すべき閾値を解析的に求めることは困難である。そこで、事前に得られている学習データを用いて、Fig. 2.21 と同様の、診断成功率とアーク数の関係を示す表を作成して、(III) の領域の境界を求める。具体的には BN の構造学習用のデータを検証データとモデル学習データに分割して、検証データの診断成功率を各アーク数に対して評価することにより Fig. 2.21 を作成できる。

2.7 本章のまとめ

本章では事象駆動型モデルに対して時間付きマルコフモデルを用いてモデル化した。また、システムをいくつかのサブシステムに分割し、Bayesian network を用いてその関係を表現した。さらに、その Bayesian network の構造を KL 情報量を用いた指標により構築する手法を提案した。

提案した Bayesian network の構造構築手法を用いることにより、計算量と計算精度のバランスの良い Bayesian network を選択できることを示した。また、Bayesian network を簡潔な構造で表現できるため、他の構造と比べ少ない学習データ量で診断を行うことができることを示した。このように事象駆動型システムを一つのシステムとしてではなく、いくつかの構造に分割し、その構造間の関係を適切に表現することにより、計算精度を減少させることなく計算量を削減することができる。

しかしながら、本手法ではいくつかの検討課題が存在する。その一つとしてサブシステムの分割法がある。サブシステムの分割は計算精度や計算量に大きく影響を与えるため、サブシステムの分割はシステムの構造や観測イベント系列から適切に決定する必要がある。また、この故障診断手法はシステムの変化に弱く、ワークの投入間隔の変化やセンサなどの位置調整が生じた場合には学習をし直さなければならない。次章ではこの環境の変化を考慮した故障診断手法について説明する。

第 3 章

プロセスの順序性に着目した故障診断

3.1 はじめに

観測されるイベントとその生起時刻の系列を確率的なモデルである時間付きマルコフモデル (Timed Markov Model: TMM) により表現し、その表現に基づいて故障診断を行う手法 [17], [18] が提案されている。このモデルはイベント系列の生起順だけではなく、イベントが生起する時間間隔にも着目したモデルである。そのため、確率オートマトンなどのイベント系列の生起順のみを対象とした確率モデルに比べ、イベントの生起順に変化を及ぼさない故障、例えばアクチュエータの機能低下などの故障を診断できる。そのため、文献 [12] の手法を用いた分散型故障診断手法への適用や、未検討事項であったイベント間の生起時間間隔の確率密度関数の推定手法 [17] などが研究されている。しかしながら、TMM を用いた手法はイベントの生起時間間隔に関してもモデル化するため、多くの学習データを必要とする。また、動作仕様の変更などによりイベントの生起時間間隔の変更があった場合は学習をやり直さなければならず、再学習に伴うデータの収集と計画の負荷が大きいという問題がある。

本研究では、TMM の学習量を削減するため、事象駆動型システムにおいてイベントによって生起する動作に着目し、その動作の組み合わせとしてシステムの挙動を表現することを考える。事象駆動型システムは生起したイベントに対応してシステムの制御が切り替えられる。例えばシーケンス制御では、システムがセンサの活性に連動してアクチュエータを動作させ、システムの一連の動作 (工程) が実現される。このとき、システムの挙動はアクチュエータによる複数の工程の組み合わせとして表現できる。このようにシステムの挙動をいくつかの小さな工程の組み合わせとして表現することにより、学習量の低減が期待できる。

システムの挙動をいくつかの工程に分割して考えることにより、学習の方法にも違いが生じる。従来手法 [17], [18] においてシステム全体の挙動をモデル化する場合には、診断

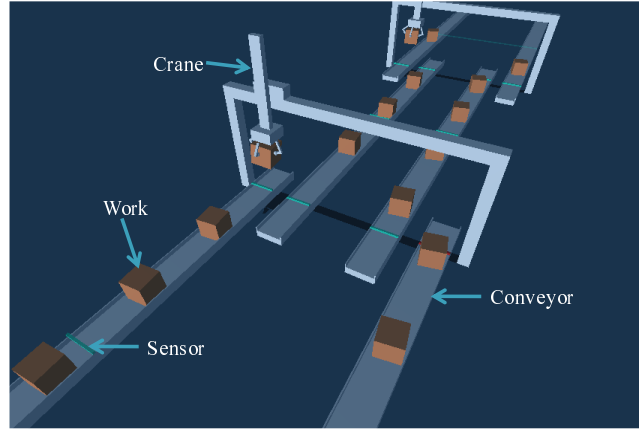


Fig.3.1 Simulated transfer line

時と学習時の状況が同じでなければならない。一方、提案手法ではいくつかの工程に分割して考えているので、一つ一つの動作に分解して学習すればよい。そのため、イベントの生起時間間隔などの変更が生じたとしても、システム全体の挙動を再学習する必要はなく、変更部分のみを学習することにより対応できる。

本章は以降、次のように構成される。3.2 節では本論文の問題設定について述べる。3.3 節では TMM によるモデル化手法、3.4 節では TMM を用いた故障診断手法、3.5 節では TMM の学習方法について述べる。そして、3.6 節において本提案手法を自動搬送ラインのシミュレータに適用した例を示し、その有用性を実証する。

3.2 問題設定

3.2.1 診断対象のシステム

イベントを生起させる物理的主体をトークンと呼び、トークンに対する工程により生起するイベントの系列をプロセスと呼ぶことにする。本論文は、トークンが入力されるとプロセスが生起するシステムを診断の対象とする。システムにおけるイベントのラベルの総数が n_v であるとき、イベントのラベルの集合を $\mathbf{V} = \{v_i | i \in \{0, 1, \dots, n_v\}\}$ とする。ただし、イベント v_0 はトークンにより生じるイベントではなく、トークンが投入されていないことを示すイベントとする。そして、トークンのラベルの集合を $\mathbf{O} = \{o^i | i \in \{1, \dots, L\}\}$ とする。ただし、 L はシステムに入力できるトークンの上限値である。本論文ではシステムの動作をイベントの系列として表現するため、プロセスをイベントの系列として表現する。このとき、プロセスの集合 \mathbf{P} とその要素 p_i を

$$p_i = v'_1 v'_2 \cdots v'_{n_{p_i}} \quad \forall j \in \{1, \dots, n_{p_i}\} \quad (v'_j \in \mathbf{V}), \quad (3.1)$$

$$\mathbf{P} = \{p_i | i \in \{1, \dots, n_p\}\} \quad (3.2)$$

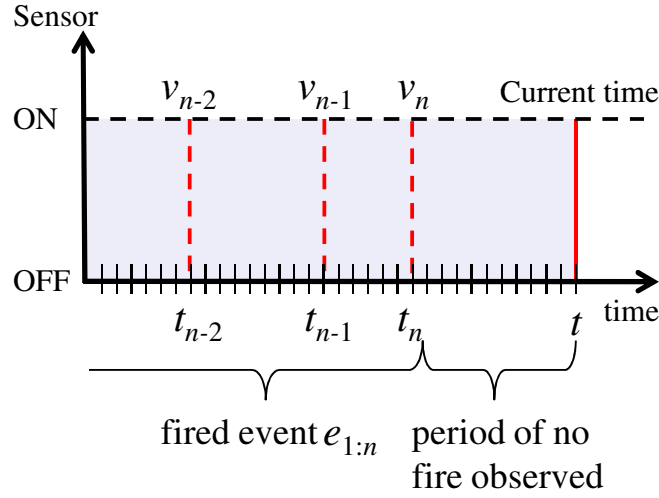


Fig.3.2 Observation of timed events

と定義する．トークンの入力により，式 (3.1) において定められたプロセスが実行される．トークンからプロセスへの写像を $f : \mathbf{O} \rightarrow \mathbf{P}$ とする．システムの状態はトークンがプロセス中に最後に生起させたイベント $s_{o^i} \in \mathbf{V}$ の組 $\mathbf{S} = (s_{o^1}, \dots, s_{o^L})$ とする．そして，システムの初期状態を \mathbf{S}_0 とする．このとき，本論文が対象とするシステムは $\Sigma = (\mathbf{V}, \mathbf{O}, \mathbf{P}, \mathbf{S}, \mathbf{S}_0, f)$ と記述できる．

例として Fig. 3.1 のコンベアとクレーンからなるワークの搬送システムを考える．ここで， v_i はセンサのラベル，トークンは運搬されるワークとなる．このとき，トークンにおけるプロセスはワークの搬送動作であり，ワークが投入されてから搬送終了までに生起するイベントの系列として表現される．また，システムの状態は各ワークの搬送動作中に生起させた直近のイベントの組となり，ワークの搬送状況を表している．

3.2.2 故障診断問題の定式化

イベントの生起時間間隔を考慮するため，イベントは時間付きイベントとして観測されるものとする．時間付きイベントとは発火したセンサのラベルとその生起時刻の組である．システムにおいて n 回目に観測された時間付きイベント e_n を次のように定義する：

$$e_n = (v_n, t_n). \quad (3.3)$$

$v_n \in \mathbf{V}$ は生起したイベントのラベルであり， t_n は v_n が発火した時刻である．Fig. 3.2 はイベント系列とその生起時刻を表しており，観測イベント系列 \mathbf{e} を $e_{1:n} = (e_1, e_2, \dots, e_n)$ と現在時刻 t の組として次のように定義する：

$$\mathbf{e} = (e_{1:n}, t). \quad (3.4)$$

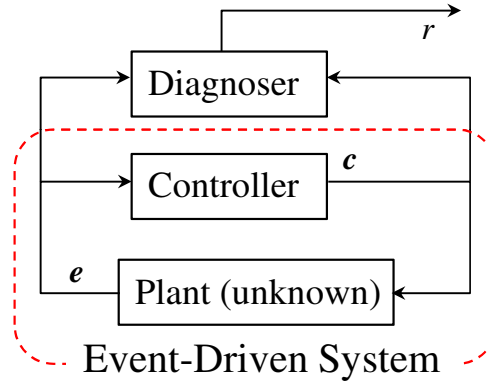


Fig.3.3 Diagnoser and event-driven system

このとき、観測イベント系列 \mathbf{e} は t_n から t までの間にイベントが生起しなかったことを表していることに注意されたい。

事象駆動型システムを Fig. 3.3 のようにプラントとコントローラに分けて考え、コントローラからの出力を観測できるものとする． k 回目におけるアクチュエータ i における制御指令値を c_k^i ，コントローラからの出力値（アクチュエータへの指令値）を $c_k = (c_k^1, c_k^2, \dots, c_k^{n_a})$ ，その時の時刻を t_k^c と定義する．ただし， n_a はシステムのアクチュエータ数である．この時間付きの出力の系列を制御指令系列 \mathbf{c} として定義する：

$$\mathbf{c} = ((c_1, t_1^c), (c_2, t_2^c), \dots, (c_{n_c}, t_{n_c}^c)). \quad (3.5)$$

また，本論文では故障はシステムの動作に影響を与えるシステムの状況であるとし，故障の集合を $\{r_0, r_1, \dots, r_{n_r}\}$ とする．つまり，故障は観測イベント系列 \mathbf{e} を変化させるシステムの状況である．ただし， n_r は想定された故障の総数である．また，故障は正常な状況を含み， r_0 とする． $\mathbf{e}, \mathbf{c}, r$ の確率変数を $\mathbf{E}, \mathbf{C}, R$ とするとき，システムの故障 r^* を推定する故障診断問題は次のような最尤推定問題として書ける．

$$\begin{aligned} \text{Given : Observed event and control state sequences } \mathbf{e}, \mathbf{c}, \\ \text{find : } r^* = \arg \max_r P(R = r | \mathbf{E} = \mathbf{e}, \mathbf{C} = \mathbf{c}). \end{aligned} \quad (3.6)$$

式 (3.6) の確率を計算するためには， \mathbf{E} と \mathbf{C} ，つまり制御指令系列と観測イベントの系列の関係を確率で表現する必要がある．しかし，制御指令系列と観測イベントの系列の関係はシステムや工程に大きく依存し，かつそれぞれが系列で与えられるため，その関係を直接表現することは難しい．そこで，トークンごとのプロセスを明示的に考慮することにより，制御指令系列と観測イベント系列の関係を表現する．

3.2.3 各トークンにおける時間付きイベント

時間付きイベントはセンサのラベルと生起時刻の組として観測されるだけであり，時間付きイベントがどのトークンによって生起したか区別することはできない．しかし，ここ

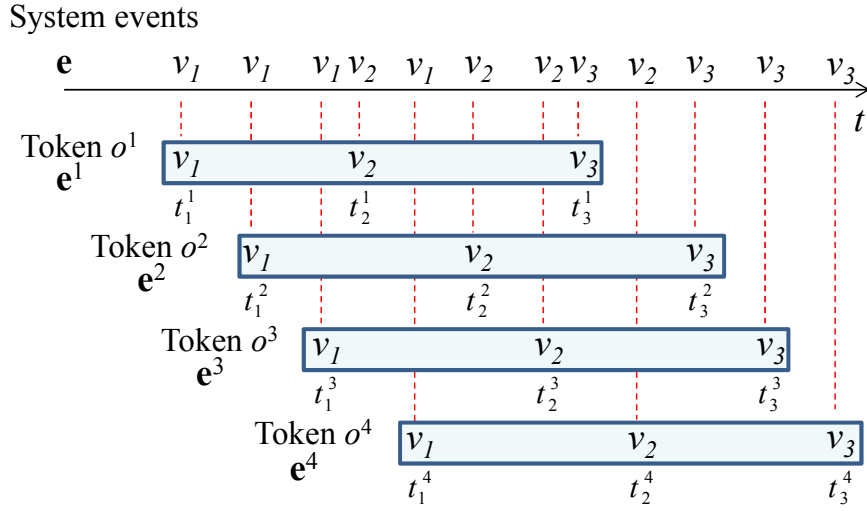


Fig.3.4 The event observation of parallel processes

では仮に時間付きイベントがどのトークンによって生じたかを区別できる場合を考える．このとき，トークン o^i が生起させた n_i 回目のイベント $v_{n_i}^i$ とその時刻 $t_{n_i}^i$ を時間付きイベント $e_{n_i}^i$ として定義する：

$$e_{n_i}^i = \begin{cases} (v_{n_i}^i, t_{n_i}^i), & \text{if } n_i \geq 1, \\ \phi, & \text{otherwise.} \end{cases} \quad (3.7)$$

ただし， ϕ はイベントが生起していないことを示す値である．

トークン o_i が生起させる時間付きイベントの系列 $e_{1:n_i}^i = (e_1^i, \dots, e_{n_i}^i)$ と現在時刻 t の組をトークンイベント系列 \mathbf{e}^i として定義する：

$$\mathbf{e}^i = (e_{1:n_i}^i, t). \quad (3.8)$$

また， \mathbf{e}^i の確率変数を \mathbf{E}^i とする．ただし，繰り返しになるが，本論文で取り扱う故障診断においては，時間付きイベントがどのトークンによって生じたかを区別できない．すなわち， \mathbf{e}^i を直接観測できないことに注意されたい．

3.2.4 各変数間の関係

本研究は対象とするシステムを以下の条件を満たすものとして，故障診断手法を適用する：

- 条件 1 \mathbf{e}^i はプロセスに従う順序性を持ち， \mathbf{e} は各 \mathbf{e}^i の重ね合わせとなる．
- 条件 2 プロセスは他のプロセスから直接影響を受けず，システムの制御およびシステムの故障状況のみから直接影響を受ける．
- 条件 3 プロセスは有限時間内に終了する．

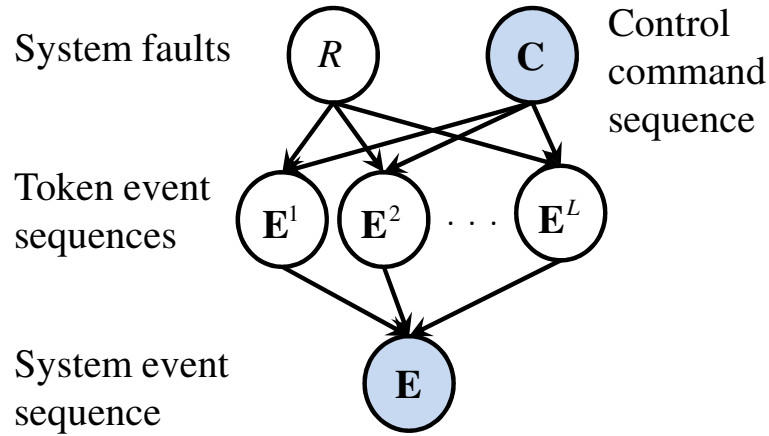


Fig.3.5 Relation between observations and faults: Grey nodes mean observable ones, white nodes non-observable ones

条件 4 時間付きイベントがどのトークンによって生じたか区別できない。

条件 1 では観測されるイベントはトークンによって生起され、プロセスで生起したイベントの重ね合わせとして表すことができることを示している。例として Fig. 3.4 では、プロセスは $p_1 = v_1 v_2 v_3$ であり、全てのトークンはプロセスの順番にイベントを生起させる。イベントはトークンによって生起するため、 \mathbf{e} は \mathbf{e}^i の重ね合わせとして表現される。このとき、各 \mathbf{e}^i を重ね合わせのイベント系列として表現する関数を

$$g : (\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^L) \mapsto \mathbf{e} \quad (3.9)$$

と定義する。条件 2 を Fig. 3.1 の搬送システムの例で考えると、ワークのプロセスは投入されている他のワークの個数や状況に依存せず、システムの制御と故障状況のみに依存して実行されることを意味する。しかし、プロセスは制御により、他のプロセスから間接的に影響を受けることがある。本論文ではこの他のプロセスからの間接的な影響は既知とする（詳細は 3.4.1 節において述べる）。また条件 2 により、提案手法ではプロセスの生起や状況によって他のプロセスへ影響を与えるような故障（ワークの投入に関する故障等）は取り扱うことができない。プロセスが有限時間内に終了しないとすると、システムの稼働時間の増加に従いプロセス数が無限に増加し続けてしまうため、条件 3 においてプロセスは有限時間内に終了するものとする。条件 4 は 3.2.3 節において述べた通りである。

上記の条件を満たすとき、各変数の関係は Fig. 3.5 のベイジアンネットワークにより表現できる。図中の灰色のノードは観測できる変数を、白色はできない変数を表している。ここで、条件 2 より \mathbf{E}^i は R と C に関して条件付き独立であるとしている。また条件 1 より故障 R の影響は \mathbf{E}^i に直接影響を与え、間接的に観測イベント系列 \mathbf{E} に影響を与える。このとき、 \mathbf{E}^i は \mathbf{E} を表現するための、観測できない中間変数であると捉えることができる。

以降では、式の表記の簡略化のために、例えば $E_{1:n_i}^i = e_{1:n_i}^i$ を $E_{1:n_i}^i$ のように、実現値を省略して確率変数を表現する。

3.3 システムの故障診断

本節では式 (3.6) の故障確率 $P(R|\mathbf{E}, \mathbf{C})$ の計算法について述べる。故障確率 $P(R|\mathbf{E}, \mathbf{C})$ は Fig. 3.5 のベイジアンネットワークから、

$$\begin{aligned} P(R|\mathbf{E}, \mathbf{C}) \\ = \sum_{\mathbf{E}^1, \dots, \mathbf{E}^L} P(R|\mathbf{E}^1, \dots, \mathbf{E}^L, \mathbf{C}) P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{E}, \mathbf{C}) \end{aligned} \quad (3.10)$$

と表現できる。ここで、 $P(R|\mathbf{E}^1, \dots, \mathbf{E}^L, \mathbf{C})$ は各トークンイベント系列 \mathbf{E}^i と制御指令系列 \mathbf{C} が与えられたときの故障確率であり、 $P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{E}, \mathbf{C})$ は観測イベント系列 \mathbf{E} と制御指令系列 \mathbf{C} が与えられたときの、 \mathbf{E}^i の生起確率である。つまり、観測イベント系列 \mathbf{E} と制御指令系列 \mathbf{C} に基づき各トークンイベント系列 \mathbf{E}^i の生起確率を計算し、各 \mathbf{E}^i が得られたときの故障確率を計算している。

3.3.1 トークンに着目した故障確率の計算

式 (3.10) の確率 $P(R|\mathbf{E}^1, \dots, \mathbf{E}^L, \mathbf{C})$ は Fig. 3.5 のベイジアンネットワークとベイズの定理より

$$\begin{aligned} P(R|\mathbf{E}^1, \dots, \mathbf{E}^L, \mathbf{C}) &= \frac{P(\mathbf{E}^1, \dots, \mathbf{E}^L|R, \mathbf{C})P(R)}{\sum_R P(\mathbf{E}^1, \dots, \mathbf{E}^L|R, \mathbf{C})P(R)} \\ &= \frac{P(R) \prod_{i=1}^L P(\mathbf{E}^i|R, \mathbf{C})}{\sum_R P(R) \prod_{i=1}^L P(\mathbf{E}^i|R, \mathbf{C})} \end{aligned} \quad (3.11)$$

と計算できる。 $P(R)$ は故障の事前確率であり、過去のシステム運用に基づく統計的確率や専門家の先験的知識などから決定される。また、 $P(\mathbf{E}^i|R, \mathbf{C})$ はトークンイベント系列の生起確率である。

3.3.2 観測に基づくトークンイベント系列の生起確率計算

式 (3.10) の確率 $P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{E}, \mathbf{C})$ はベイズの定理により次のように表現できる。

$$\begin{aligned} P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{E}, \mathbf{C}) \\ = \frac{P(\mathbf{E}|\mathbf{E}^1, \dots, \mathbf{E}^L)P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{C})}{\sum_{\mathbf{E}^1, \dots, \mathbf{E}^L} P(\mathbf{E}|\mathbf{E}^1, \dots, \mathbf{E}^L)P(\mathbf{E}^1, \dots, \mathbf{E}^L|\mathbf{C})}. \end{aligned} \quad (3.12)$$

このとき、 $P(\mathbf{E}^1, \dots, \mathbf{E}^L | \mathbf{C})$ は Fig. 3.5 のベイジアンネットワークから次のように分解できる.

$$P(\mathbf{E}^1, \dots, \mathbf{E}^L | \mathbf{C}) = \sum_R P(R) \prod_{i=1}^L P(\mathbf{E}^i | R, \mathbf{C}). \quad (3.13)$$

3.3.3 故障確率の計算

故障確率 $P(R | \mathbf{E}, \mathbf{C})$ は式 (3.10), (3.11), (3.12), (3.13) から

$$\begin{aligned} P(R | \mathbf{E}, \mathbf{C}) &= \frac{1}{Z} \sum_{\mathbf{E}^1, \dots, \mathbf{E}^L} \left\{ P(R) \prod_{i=1}^L \{P(\mathbf{E}^i | R, \mathbf{C})\} \right. \\ &\quad \left. \times P(\mathbf{E} | \mathbf{E}^1, \dots, \mathbf{E}^L) \sum_R \{P(R) \prod_{j=1}^L P(\mathbf{E}^j | R, \mathbf{C})\} \right\} \end{aligned} \quad (3.14)$$

と計算できる. ただし, Z は正規化項である. 式 (3.14) では各トークンイベント系列 \mathbf{E}^j の生起確率 $P(\mathbf{E}^j | R, \mathbf{C})$ を求め, 故障確率を計算している. この計算では各 \mathbf{E}^j に対して周辺化計算しなければならないため, 生起し得る \mathbf{E}^j の全ての候補に対して確率を計算する必要がある. このとき, \mathbf{E}^j の候補は生起し得るイベントの組み合わせを考慮しなければならない. しかし, \mathbf{E}^j はプロセスのイベントの順序通りに生起するため, イベントの組み合わせを考慮する必要はなく, プロセスの順序に従って計算すればよい. 確率 $P(\mathbf{E} | \mathbf{E}^1, \dots, \mathbf{E}^L)$ は各トークンイベント系列 \mathbf{E}^i が与えられたときの観測イベント系列 \mathbf{E} の生起確率である. \mathbf{E} は \mathbf{E}^i の重ね合わせとして表現されるため, 各 \mathbf{E}^i が与えられる場合は \mathbf{E} を一意的に定めることができる. つまり,

$$P(\mathbf{E} | \mathbf{E}^1, \dots, \mathbf{E}^L) = \begin{cases} 1 & \text{if } \mathbf{e} = g(\mathbf{e}^1, \dots, \mathbf{e}^L) \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

となる. したがって, 式 (3.14) の故障確率を計算するためには故障確率 $P(R)$, トークンイベント系列の生起確率 $P(\mathbf{E}^j | R, \mathbf{C})$ の値が必要である. 次節では TMM を用いて $P(\mathbf{E}^j | R, \mathbf{C})$ を求める方法について説明する.

3.4 時間付きマルコフモデル

Timed Markov Model (TMM) は準マルコフ過程モデルの一つであり, 状態遷移確率が現在の状態における滞在時間 τ に依存するモデルである. 時間付きイベント系列を TMM で表現するために, TMM の状態を生起したイベントのラベルとする [18]. このとき, TMM の状態遷移確率は生起したイベントと次のイベントが生起するまでの時間 τ の同時確率として表現できる. なお, イベントの生起時間間隔 τ は離散時間であるとし, 状態遷移確率は同時確率分布により表される.

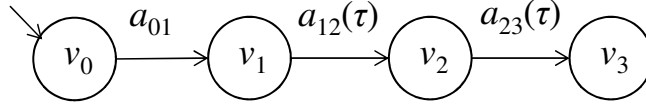


Fig.3.6 The example of Left-to-Right TMM representing a process

以下に故障 r においてトークン o_i があるプロセス中に生成させるトークンイベント系列を表現する TMM を考える.

定義 2 時間付き遷移確率 (Timed Transition Probability: TTP) とは, 故障が r であり, イベント $v_{n_i}^i$ が時刻 $t_{n_i}^i$ に生起し, そして, $\tau = t_{n_{i+1}}^i - t_{n_i}^i$ 時間後に $v_{n_{i+1}}^i$ が発火するときの同時確率分布 $P(E_{n_{i+1}}^i = e_{n_{i+1}}^i | E_{n_i}^i = e_{n_i}^i, R = r)$ である. ここで, 時間付き遷移確率はイベントが生起する絶対時刻に依らないと仮定すると,

$$\begin{aligned}
 & P(E_{n_{i+1}}^i = e_{n_{i+1}}^i | E_{n_i}^i = e_{n_i}^i, R = r) \\
 &= P(V_{n_{i+1}}^i = v_{n_{i+1}}^i, T_{n_{i+1}}^i = t_{n_i}^i + \tau | V_{n_i}^i = v_{n_i}^i, T_{n_i}^i = t_{n_i}^i, R = r) \\
 &= P(V_{n_{i+1}}^i = v_{n_{i+1}}^i, T = \tau | V_{n_i}^i = v_{n_i}^i, R = r) \\
 &\stackrel{\text{def}}{=} a_{v_{n_i}^i v_{n_{i+1}}^i}^r(\tau)
 \end{aligned} \tag{3.16}$$

となる. ただし $V_{n_{i+1}}^i, T_{n_{i+1}}^i, T$ は $v_{n_{i+1}}^i, t_{n_{i+1}}^i, \tau$ の確率変数である.

各トークンのイベントの生起時間を時刻ではなく, 時間間隔として表現することにより, トークンの投入時刻に依らずトークンイベント系列を表現することができる.

TMM はトークンがあるプロセス中に生起させる時間付きイベントの生成過程を表現している. システムにおいて各トークンは区別できないため, 同一プロセスでは各トークンの間に差はないものとする. つまり, TMM はプロセスのみに依存する. また, 各プロセスにおいて生起するイベントのラベルは順序性を持つ. 例として Fig. 3.4 のシステムを考える. このとき, 全てのトークンはプロセスに従い v_1, v_2, v_3 と順番にイベント生起するため, TMM は Fig. 3.6 の Left-to-Right 型のモデルとして表現できる. また, Fig. 3.6 の TMM の状態遷移は限られているため, 少ない TTP で表現することができる. このように, システムで観測されるイベント系列 \mathbf{E} ではなく, 各トークンのイベント系列 \mathbf{E}^i に着目することにより, TTP は全てのイベントの組み合わせを考慮する必要はなく, 限られた状態遷移で表現できる. よって, TTP の数は $O(n_v^2)$ から $O(n_v)$ へ減少し, 学習量の削減を期待できるのである.

TMM の初期状態を v_0 とし, この状態はトークンが投入されていない待機状態を表す. トークンにおいて初回に観測されるイベント v_1 が観測されたとき, TMM は初期状態から状態 v_1 に確率 1 で遷移するものとする.

3.4.1 制御指令系列による影響

トークンイベント系列 \mathbf{E}^i は制御指令系列 \mathbf{C} の影響を受ける． \mathbf{E}^i が \mathbf{C} から受ける影響は制御指令値の変更によって生じるため，制御手法に着目するとその影響が分かる場合がある．その例として一時停止制御と速度変化制御について考え， \mathbf{C} の影響を考慮した TPP を次のように定義する．

定義 3 故障が r であり，制御指令系列 \mathbf{C} と時間付きイベント $E_{n_i}^i$ が観測され，時間付きイベントが生起する絶対時刻に依らないと仮定したとき，次の時間付きイベント $E_{n_i+1}^i$ が生起する TPP を

$$f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, \mathbf{c}) \stackrel{\text{def}}{=} P(E_{n_i+1}^i | E_{n_i}^i, R, \mathbf{C}) \quad (3.17)$$

とする．

一時停止制御

一時停止制御はリソースの競合や例外処理などによりアクチュエータを一時的に停止させる制御であり，TMM の一時停止処理に相当する．一時停止制御が t_a から t_b まで動作しているときの TPP は次のようになる．

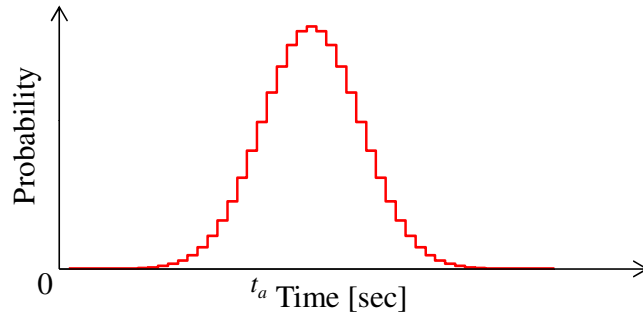
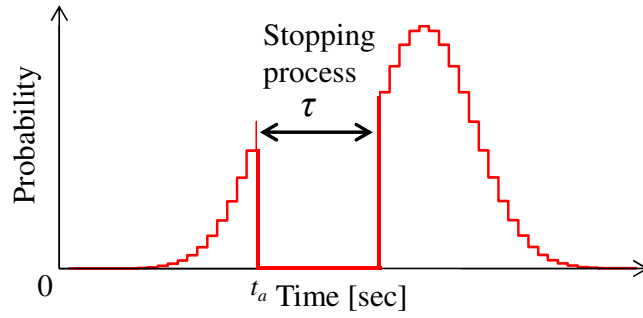
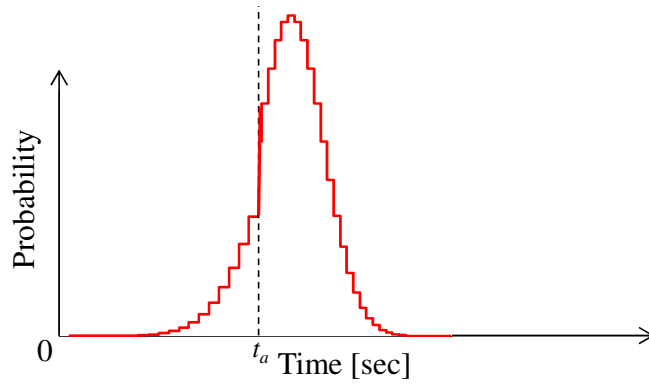
$$\begin{aligned} & f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, \mathbf{c}) \\ &= \begin{cases} a_{v_{n_i}^i v_{n_i+1}^i}^r(\tau) & (0 \leq \tau \leq t_a) \\ 0 & (t_a < \tau \leq t_b) \\ a_{v_{n_i}^i v_{n_i+1}^i}^r(\tau - (t_b - t_a)) & (t_b < \tau) \end{cases} \end{aligned} \quad (3.18)$$

Fig. 3.7 は制御指令系列 \mathbf{C} の影響がないときの TPP を示している．Fig. 3.8 は一時停止制御の影響がある場合の TPP であり，停止期間に 0 の確率を挿入することにより表現できる．

速度変化動作

次にアクチュエータの動作速度が変化する場合の影響について考える．アクチュエータの動作速度が α 倍に変化するとき，トークンのイベントの生起確率は α 倍に，アクチュエータの動作時間は $1/\alpha$ 倍に変化すると考えられる．時刻 t_a から動作速度が変化するときの TPP は

$$\begin{aligned} & f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, \mathbf{c}) \\ &= \begin{cases} a_{v_{n_i}^i v_{n_i+1}^i}^r(\tau) & (0 \leq \tau \leq t_a) \\ \alpha a_{v_{n_i}^i v_{n_i+1}^i}^r\left(\frac{1}{\alpha}(\tau - t_a) + t_a\right) & (t_a < \tau) \end{cases} \end{aligned} \quad (3.19)$$

Fig.3.7 A example of $a_{v_{n_i}^i v_{n_i+1}^i}^r(\tau)$ Fig.3.8 $f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, c)$ after a stop action controlFig.3.9 $f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, c)$ after a speed change control

となる. Fig. 3.9 は時刻 t_1 において実行速度が変化したときの TTP であり, TTP の確率は時間的に伸縮したように表現できる.

3.4.2 トークンイベント系列の生起確率の計算

故障状況が r であるときの、トークンイベント系列の生起確率は TMM より

$$\begin{aligned} P(\mathbf{E}^i = \mathbf{e}^i | R, \mathbf{C}) \\ = P(E_{1:n_i}^i = e_{1:n_i}^i, [T_{n_i+1}^i > t] = \text{True} | R, \mathbf{C}) \end{aligned} \quad (3.20)$$

と表現できる．ここで、 $[X]$ は論理式 X が成立するか否かを表す確率変数であり、実現値として $\{\text{True}, \text{False}\}$ を取るものとする．式 (3.20) は、故障状況が r であり、制御指令系列が \mathbf{C} の場合のとき、 $e_{1:n_i}^i$ が得られ、かつ現在時刻 t までセンサの発火がないときの確率である．

過去直近のイベント生起時刻 $t_{n_i}^i$ から現在時刻 t までの間センサが発火しないため、式 (3.20) は次のように計算される．

$$\begin{aligned} P(\mathbf{E}^i = \mathbf{e}^i | R, \mathbf{C}) \\ = P(E_{1:n_i}^i, [T_{n_i+1}^i > t] | R, \mathbf{C}) \\ = P([T_{n_i+1}^i > t] | E_{1:n_i}^i, R, \mathbf{C}) P(E_{1:n_i}^i | R, \mathbf{C}). \end{aligned} \quad (3.21)$$

ここで、式 (3.21) 右辺の確率 $P([T_{n_i+1}^i > t] | E_{1:n_i}^i, R, \mathbf{C})$ は直近のイベントから現在時刻までにイベントが生起しない確率であり、TMM のマルコフ性と TTP を用いて

$$\begin{aligned} P([T_{n_i+1}^i > t] | E_{1:n_i}^i, R, \mathbf{C}) \\ = P([T_{n_i+1}^i > t] | E_{n_i}^i, R, \mathbf{C}) \\ = 1 - \sum_{v_{n_i+1}^i \in \mathbf{V}} \int_{t_{n_i}}^t P(V_{n_i+1}^i = v_{n_i+1}^i, T_{n_i+1}^i = \tau | E_{n_i}^i, R, \mathbf{C}) d\tau \\ = 1 - \sum_{v_{n_i+1}^i \in \mathbf{V}} \int_{t_{n_i}}^t f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, \mathbf{c}) d\tau \end{aligned} \quad (3.22)$$

と計算することができる．

また、 $T_{n_i+1}^i = t$ となったとき、つまり新たに時間付きイベントが生起したとき、式 (3.22) の n_i は n_{i+1} に更新される．このとき、確率 $P(E_{1:n_{i+1}}^i | R, \mathbf{C})$ は以下のように計算される．

$$\begin{aligned} P(E_{1:n_{i+1}}^i | R, \mathbf{C}) &= P(E_{n_i+1}^i | E_{1:n_i}^i, R, \mathbf{C}) P(E_{1:n_i}^i | R, \mathbf{C}) \\ &= P(E_{n_i+1}^i | E_{n_i}^i, R, \mathbf{C}) P(E_{1:n_i}^i | R, \mathbf{C}) \\ &= f_{v_{n_i}^i v_{n_i+1}^i}^r(\tau, \mathbf{c}) P(E_{1:n_i}^i | R, \mathbf{C}). \end{aligned} \quad (3.23)$$

各時刻において、イベントが観測されないときは式 (3.22) を用いて、観測されたときは式 (3.23) を用いて更新することにより、各時刻の $P(\mathbf{E}^i = \mathbf{e}^i | R, \mathbf{C})$ が得られる．

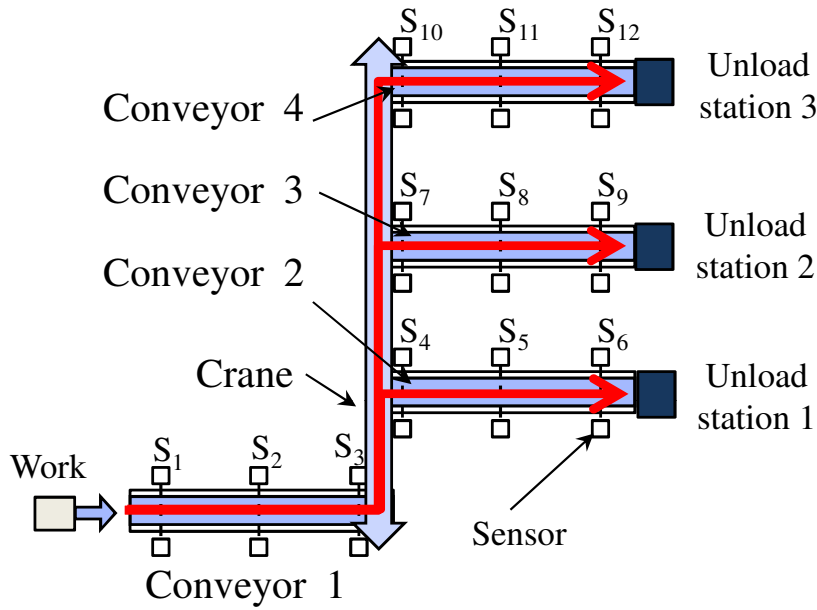


Fig.3.10 Diagram of simulated transfer line

3.5 TTP の学習方法

本節では TMM のパラメータである TTP の学習について述べる．TTP は、一つのトークンがある一つのプロセス中に生起させる、時間付きイベントの生起確率を表現している．一つのプロセス中の TTP を学習するため、学習データを集める場合においては、投入されているトークン数が常に一つである状況とする．この状況下において、各故障状況ごとに集められた学習データからプロセス中に生起する時間付きイベント $e_{n_i}^i$ の生起確率、つまり $a_{v_{n_i}^i v_{n_{i+1}}^i}^r(\tau)$ を求める．そして、TMM における各 TTP を学習することにより、式 (3.10) の確率を計算する．

3.6 実験環境

本節では提案した手法を Fig. 3.10 の自動搬送ラインに適用し、その性能を調べる．なお、システムは Fig. 3.1 のような物理シミュレータを用いてシミュレーションした．

3.6.1 自動搬送ライン

このシステムは 4 機のコンベア 1, 2, 3, 4 (長さ 5 m) と 1 機のクレーンから構成され、ワークをアンロードステーション 1, 2, 3 に搬送する．各コンベアには 3 つのセンサが取り付けられており、Fig. 3.10 の位置 $S_1 \sim S_{12}$ に設置されている．

Table 3.1 Event labels in the transfer line

Event	Object	Action
v_1	S_1	Enter sensor
v_2	S_2	Enter sensor
v_3	S_3	Enter sensor (Stop Conveyor 1)
v_4	S_3	Leave sensor (Start Conveyor 1)
v_5	S_4	Enter sensor (Stop Conveyor 2)
v_6	Conveyor 2	Start Conveyor 2
v_7	S_5	Enter sensor
v_8	S_6	Enter sensor
v_9	S_7	Enter sensor (Stop Conveyor 3)
v_{10}	Conveyor 3	Start Conveyor 3
v_{11}	S_8	Enter sensor
v_{12}	S_9	Enter sensor
v_{13}	S_{10}	Enter sensor (Stop Conveyor 4)
v_{14}	Conveyor 4	Start Conveyor 4
v_{15}	S_{11}	Enter sensor
v_{16}	S_{12}	Enter sensor

コンベア 1 の制御はセンサ S_3 にワークが侵入するとアクチュエータが一時停止し、離れると始動する。クレーンはセンサ S_3 にワークが侵入すると始動し、ワークから搬送先 (コンベア 2, 3, 4) を読み取り、その搬送先へワークを運搬する。コンベア 2, 3, 4 は安全に搬送するために S_4, S_7, S_{10} にワークが侵入すると一時停止し、タイマによって再始動する (始動時にイベント v_6, v_{10}, v_{14} を発火する)。各イベントは Table 3.1 に記された通りに発火する。Table 3.2 はシステムのプロセスを示しており、各ワークのプロセスは予め搬送先に応じて決められている。また、Table 3.3 は各アクチュエータへの指令値 c_k^i の集合を表しており、制御指令系 c_k はこれらの組み合わせとなり、制御指令系列 \mathbf{c} は c_k の時間付きの系列として与えられる。コンベアの停止は他のワークに影響を与える制御であるため、この影響は 3.4.1 節の手法を用いて計算した。

ワークの投入間隔の確率は平均 1.8m, 分散 0.4m^2 の正規分布とし、この分布に従い投入する。また、1 回の試行はワークを 30 個搬送することとする。

故障状況

本論文で想定する故障を Table 3.4 に示す。各ベルトコンベアの速度が 22cm/sec から 18cm/sec の間である場合を「正常 (r_0)」とし、18cm/sec から 15cm/sec の間である場合は「速度低下 (r_1, r_2, r_3, r_4)」であるとする。クレーンにおいてワークの搬送に要する時

Table 3.2 Fired events of processes for each destination

Process	Fired events	Destination
p_1	$v_1v_2v_3v_4v_5v_6v_7v_8$	Conveyor 2
p_2	$v_1v_2v_3v_4v_9v_{10}v_{11}v_{12}$	Conveyor 3
p_3	$v_1v_2v_3v_4v_{13}v_{14}v_{15}v_{16}$	Conveyor 4

Table 3.3 Command set for each actuator

Label	Object	Command set
c_k^1	Conveyor 1	{Start, Stop }
c_k^2	Conveyor 2	{Start, Stop }
c_k^3	Conveyor 3	{Start, Stop }
c_k^4	Conveyor 4	{Start, Stop }
c_k^5	Crane	{Move a work, Wait }

Table 3.4 Candidates of faults

Symbol	Detail of fault
r_0	The system is normal.
r_1	Speed of Conveyor 1 is reduced.
r_2	Speed of Conveyor 2 is reduced.
r_3	Speed of Conveyor 3 is reduced.
r_4	Speed of Conveyor 4 is reduced.
r_5	Speed of Crane is reduced.

間が正常時より 0.3sec 多くなる時を「速度低下 (r_5)」であるとする。また、システムの故障に関する事前情報はないものとして、事前確率 $P(R)$ には等確率を割り振る。

3.6.2 実験結果

故障診断における検証データは各故障ごとに 100 試行、計 600 試行のデータを使用した。この検証データに提案手法を適用した故障診断結果を Table 3.5 に示す。また、比較対象としてシステムの観測イベント系列 \mathbf{E} をそのまま TMM によりモデル化する手法 [18] を従来手法として用いる。従来手法ではプロセスやトークンを考慮しないため、学習データは検証データと同一の状況で収集された。提案手法では 3.5 節でも述べた通り、常にワークが 1 つのみ投入されている状況において学習データを収集した。

従来手法による故障診断結果を Table 3.6 に示す。また、平均診断成功率と学習データ

Table 3.5 Success rate of diagnosis with proposed method

# of data	r_0	r_1	r_2	r_3	r_4	r_5	Average
10 trials	0.84	0.87	0.92	0.82	1.00	0.84	0.88
20 trials	0.91	0.88	0.96	0.87	1.00	0.96	0.93
50 trials	0.92	0.9	0.97	0.96	1.00	0.97	0.95
100 trials	0.95	0.97	0.99	0.99	1.00	0.98	0.98
200 trials	0.95	0.97	0.99	0.99	1.00	0.97	0.98
300 trials	0.96	0.99	0.93	0.98	1.00	0.98	0.97
400 trials	0.97	0.99	0.98	0.99	1.00	1.00	0.98
500 trials	0.96	0.99	1.00	0.99	1.00	1.00	0.99
1000 trials	0.96	0.99	1.00	0.96	1.00	1.00	0.99

Table 3.6 Success rate of diagnosis with conventional method

# of data	r_0	r_1	r_2	r_3	r_4	r_5	Average
10 trials	0.37	0.10	0.71	0.78	0.87	0.17	0.50
20 trials	0.55	0.41	0.78	0.77	0.86	0.24	0.60
50 trials	0.46	0.69	0.74	0.84	0.98	0.38	0.68
100 trials	0.45	0.76	0.84	0.89	0.97	0.48	0.73
200 trials	0.34	0.76	0.9	0.92	0.99	0.68	0.77
300 trials	0.44	0.89	0.95	0.96	0.98	0.58	0.80
400 trials	0.51	0.82	0.93	0.97	0.98	0.55	0.79
500 trials	0.53	0.81	0.94	0.91	0.99	0.56	0.79
1000 trials	0.53	0.85	0.95	0.94	0.99	0.56	0.80

Table 3.7 The number of TTP in each method

Conventional method	Proposed method
1536	90

量の関係を示したグラフを Fig. 3.11 に示し、各手法における TTP の数を Table 3.7 に示す。なお、故障の同定は次式の故障確率のエントロピーが 0.025 を下回る場合と判断した。

$$H = - \sum_R P(R|\mathbf{E}, \mathbf{C}) \log_2 P(R|\mathbf{E}, \mathbf{C}). \quad (3.24)$$

提案手法の診断結果である Table 3.5 では、学習データ量が増加すると診断成功率も高くなる傾向があった。また、提案手法は 100 試行の学習データでも高い診断成功率を示した。一方、従来手法の診断結果である Table 3.6 では、提案手法と比較すると全体的に診

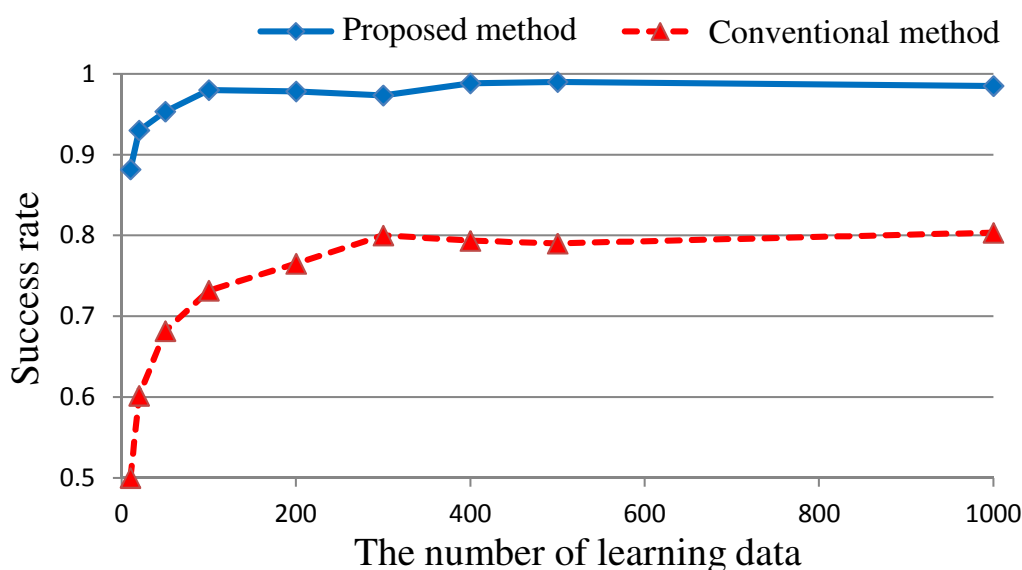


Fig.3.11 The relation between average success rate and learning data

断成功率は低くなった。また、300 試行の学習データを用いたとき、平均の診断成功率が最も高くなった。

以上の結果より提案手法の方が少ない学習データ量でも正しく診断できていることがわかる。この差はモデルとする対象の違いにより生ずるものと考えられる。提案手法ではシステムの観測イベント系列 \mathbf{E} をトークンイベント系列 \mathbf{E}^i に分割して考えている。一方、従来手法では \mathbf{E} をそのままモデル化しているため、トークン間の組み合わせを考慮しなければならない。そのため、Table 3.7 のように多くの TTP を学習しなければならない。学習データの量が同じ場合、学習すべき TTP の数が少ない方が、1 つの TPP の学習量に割り当てられる学習データの量が増えるので、提案手法では従来手法より精度が高い TPP を得ることができたと考えられる。また、Tables 3.5, 3.6 では平均診断成功率に約 20% の差が生じている。これは、トークンイベント系列が生起するイベントのラベルに関して順序性を持つため、観測イベント系列 \mathbf{E} を TMM によって表現しやすくなったためであると考えられる。

次にワーク投入間隔を変化させた場合において実験を行った。システムへのワークの投入間隔は平均 4.0m, 分散 0.4m² の正規分布に変更した。なお、学習データはワークの投入間隔を変更する前のデータを用いる。各故障ごとに 100 試行、計 600 試行のデータを検証データとし、学習データ量は 1000 試行とした。

各手法における診断結果を Table 3.8 に示す。提案手法はワークの投入間隔を変更前とほぼ変わらない診断成功率であったのに対し、従来手法では診断成功率が大きく減少した。提案手法では観測イベント系列を各トークンのプロセスごとに分割して考えているため、トークンの入力時間に依存せず診断できている。一方、従来手法ではトークンイベント系列を考慮していないため、ワークの投入間隔の変更により正しく診断できなかった。

Table 3.8 Success rate of diagnosis when setting is changed

Method	r_0	r_1	r_2	r_3	r_4	r_5	Average
Proposed method	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Previous method	0.00	0.32	0.22	0.55	0.76	0.00	0.31

トークンの入力時刻はシステム外部の変更の影響により変化することがある。その場合において、提案手法では学習データの収集をやり直す必要がないため、学習データの収集の負荷を減らすことができる。

従来手法では、事象駆動型システムにおける連続して生起するイベントの順序を観測されたデータから判断していた。それゆえ、故障診断器を作成するためには多くの学習データを必要とした。しかしながら、事象駆動型システムではプロセスなどのシステムの設計時に定められた動作をするものが多い。このような場合では、連続して生起するイベントの順序はデータから判断するのではなく、プロセスとして事前知識として予め与えることができる。上記の点を踏まえ、提案手法ではイベントの生起順序をプロセスとして事前に与え、故障診断モデルを作成した。そして、イベントの生起確率や時間分布など、事前に設計することが困難なパラメータを観測されたデータから同定した。このように、システムの動作いくつかのプロセスに分割して考えることにより、学習データ量を削減することができる。

3.7 本章のまとめ

本章では事象駆動システムにおけるプロセスに着目し、システム全体で得られる観測イベントを各プロセスごとに分割して、Timed Markov Model によりモデル化を行った。ベイジアンネットワークを用いて故障と生起するイベント系列の関係性を記述し、Timed Markov Model を用いた確率的故障診断法を提案した。そして、自動搬送ラインに提案手法を適用し、その性能を調べた。

提案手法はシステムにおける観測イベント系列を各プロセスごとに分割し、各プロセスに着目して学習を行っている。そのため、少ない学習量で診断でき、より高い精度で診断できた。また、トークンの投入時刻に依らず故障を診断できることを示した。

このようにシステムの構造ではなく、システムの動作をいくつかの部分に分割することにより、計算量を削減することができる。

第 4 章

動的計画法を用いた車の走行予測

4.1 はじめに

電気自動車 (Electric Vehicle:EV) やプラグインハイブリッドカー (Plug-in Hybrid Vehicle:PHV) を高度交通システム (Intelligent Transportation System:ITS) とエネルギーマネジメントシステム (Energy Management System:EMS) の両方のシステムにおいて最大限有効活用するためには、各車の走行傾向に合わせた車の到着と出発のパターン (Profile of the Departure and Travel Time: PDTT) の予測は重要な問題である。車の使用状況は車を乗り物 (ITS) として使用している状態と蓄電池 (Energy Management System: EMS) として使用している状態の二状態として表現でき、PDTT はそれらのバイナリ変数の系列として表現できる。しかしながら、PDTT はバイナリ変数の組み合わせとして表現されるため、その状態空間はとて大きくってしまう (例として一日を 48 ステップに離散化した場合、実現数は 2.8×10^{14} 通りとなる)。そのため、リアルタイムで PDTT を予測するためには計算量を削減する工夫が必要となる。

本章では、車の PDTT の予測問題を最尤問題として定式化し、観測情報と過去の出発と到着の統計データ (Statistics of Departure and Travel Time: SDTT) から予測する問題について説明する。最尤問題の大域的最適解をリアルタイムで計算するため、PDTT に対して準マルコフ過程を適用し、マルコフモデルとして表現する。そして、得られたモデルに対して動的計画法を適用することにより、最適解を導出する。

4.2 PDTT と SDTT の表現

車の利用は蓄電池として利用できる状態とできない状態の二状態として表現でき、PDTT はその系列として表現できる。ここで、‘available’ を蓄電池として利用できる状態、‘unavailable’ を蓄電池として利用できない状態 (乗り物として利用している状態) として定義する。ここで、unavailable であるときの状態を車が家庭から出発した時刻と帰宅した時刻で表現することとする。Fig. 4.1 では available と unavailable を用いて表現

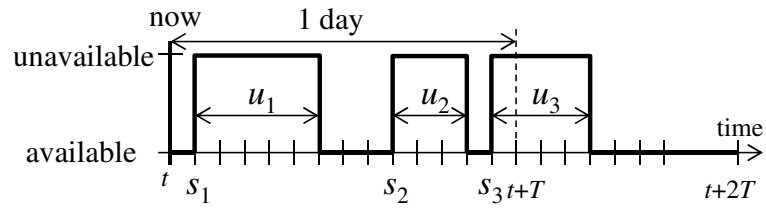


Fig.4.1 Profile of departure and travel time (PDTT) over 1 day from the present time t

Table 4.1 Statistics of departure and travel time (SDTT)

		Departure time						
		6:00	6:30	7:00	7:30	...	5:00	5:30
		1	2	3	4	...	$T-1$	T
Travel time	0	91	86	72	50	...	95	92
	1	3	1	2	4	...	1	0
	2	2	1	1	6	...	0	1
	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	$T-1$	1	3	1	2	...	1	2
	T	0	1	0	1	...	1	0

された PDTT の例を示している．なお，ここでは一日を T ステップに分割する．

現在時刻から i 番目の出発時刻を s_i ，そのときの走行時間を u_i とする．Fig. 4.1 では，時刻 s_1, s_2, s_3 において車を乗り物として使用し， u_1, u_2, u_3 の間それぞれ使用し続けている場合を示している．

SDTT は Table 4.1 のような度数分布表として表現できる．度数分布表は日々の車の PDTT の蓄積されたデータから作成される．列は出発時刻を，行はその出発時刻における走行時間を表している．例として Table 4.1 を考えると，3 行 2 列目の要素が 1 であるため，車が時刻 3 において出発し，かつ 2 ステップの間使用した回数が 1 回であったことを示している．

本手法では日々の生活サイクルを考慮して，列の数を一日である T とする*1．なお， T ステップ以上走行する場合は考慮せず，度数分布表に反映させないものとする．それ故，行の数は $T+1$ となる．このとき，最初の行は走行時間が '0' となり，その時刻において車が使用されなかったことを意味する．この度数分布表として表現される SDTT に基づき，PDTT は予測される．

*1 一日以上の期間の PDTT を予測したい場合は Table 4.1 の表の大きさを変更することにより対応できる

4.3 PDTT の最尤推定問題

本節では SDTT と動的計画法を用いた車の PDTT の最尤推定問題について説明する。

4.3.1 最尤推定問題の定式化

$\gamma(t) \in \{0, 1\}$ を時刻 t における車の使用状況を表す変数とする。ただし、0 は車が家庭にあり、蓄電池として利用できる状態 (available) とし、1 を家庭から出発し、蓄電池として使用できない状態 (unavailable) とする。また、 $s_0 \in \{1, 2, \dots, T\}$ を過去直近の車の出発時刻とする。このとき、各時刻において $\gamma(t), s_0$ が観測できるものとする。そして、PDTT を予測する問題は次のように表現できる。

A. Prediction of the PDTT

Given : $s_0, \gamma(t), \text{SDTT},$

find : $(s_1, u_1), \dots, (s_k, u_k),$

which maximize :

$$J = P((S_1, U_1) = (s_1, u_1), \dots, (S_k, U_k) = (s_k, u_k)), \quad (4.1)$$

subject to :

$$\forall i \in \{1, 2, \dots, k\} (t \leq s_i \leq t + T, 1 \leq u_i \leq T), \quad (4.2)$$

$$\forall j \in \{1, 2, \dots, k\} ([s_i, s_i + u_i] \cap [s_j, s_j + u_j] = \phi). \quad (4.3)$$

ただし、 $k \in \{1, 2, \dots, T\}$ を 24 時間先までの車の走行回数とし、 S_i, U_i を s_i, u_i の確率変数とする。この問題は現在時刻から 24 時間先までの PDTT を予測をしている。そのため、条件 (4.2) において出発時刻 s_i は現在時刻から 24 時間先まで、走行時間 u_i は 24 時間までに限定される。条件 (4.3) は二つの車の走行において時間的な重複を防ぐ条件である。この条件により、一日の k 回目までの車の走行を予測することができる。

4.3.2 準マルコフ過程を用いた PDTT のモデル化

評価関数 (4.1) は $2k$ 個の確率変数の同時確率として表現できる。そのため、評価関数 (4.1) の実現値は k の増加に従い、指数的に増加するため、SDTT から各実現値ごとに確率を同定することは困難である。この複雑さを避けるため、車の PDTT の動作を準マルコフ過程を用いて表現する。このモデリングにより、PDTT の最尤推定問題に動的計画法を適用することができ、大域的最適解を求めることができる。

ここで、 $x_t \in \{0, 1, \dots, T\}$ を時刻 t における帰宅するまでの走行時間とする。なお、走行していない状況 (車が家庭にある状況) の場合は x_t は 0 となる。一方、車が出発したときに x_t は u_i となり、走行を続けている状況では帰宅するまで走行時間が減少する。こ

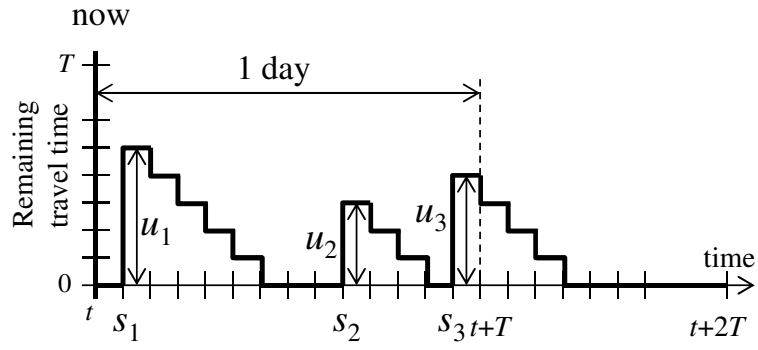


Fig.4.2 PDTT represented by the state of the remaining travel time to return home

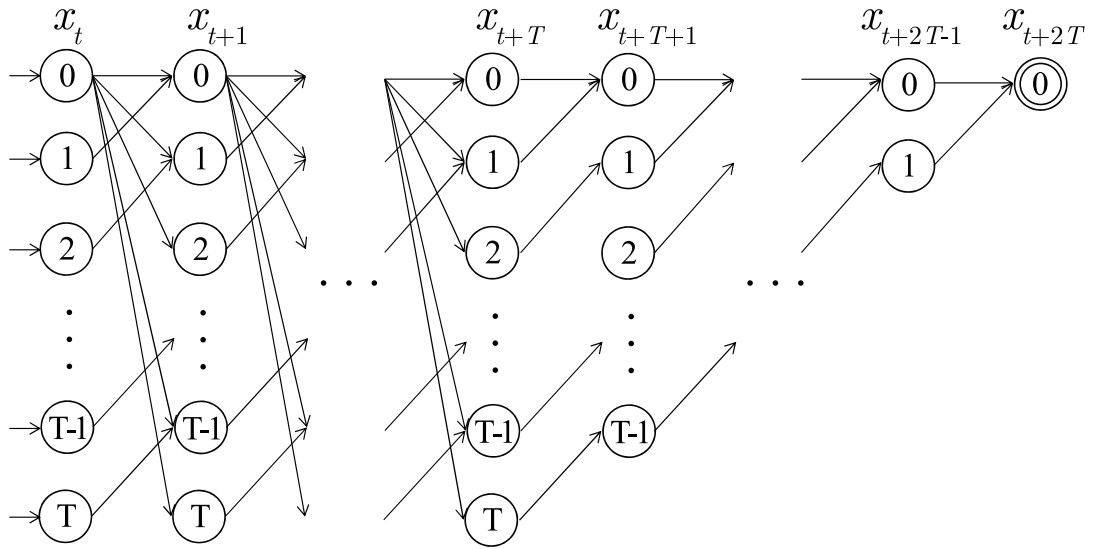


Fig.4.3 Left-to-Right Markov model of a PDTT

の x_t を用いて PDTT を表現すると Fig. 4.1 は Fig. 4.2 として表現される。

x_t の振る舞いは Fig. 4.3 のような、帰宅までの残りの走行時間を表す Left-to-Right 型のマルコフモデルとして表現できる。ここで、時刻 $t+T$ において T 時間車を走行し続ける可能性があるので、マルコフモデルの状態は $t+2T$ までを考慮する。しかし、出発時間は現在時刻から 24 時間先までの範囲を対象としているので、 $t+T$ 以降の状態は出発しない (状態 0 から次の時刻の状態 0 以外に遷移しない) ものとする。マルコフモデルの初期状態は現在の走行状態 $\gamma(t)$ と過去直近の出発時刻 s_0 に依存する (4.3.3 節参照)。このマルコフモデルでは時刻 $t+2T$ における状態 0 を最終状態としており、このようなモデルにした理由は二つある。一つの理由は全ての PDTT において時刻 $t+2T$ では必ず状態 0 となることが保証できることである。もう一つの理由は、全ての PDTT を公平に比較するためには、同じ系列長の PDTT を比較する必要があることである。

Fig. 4.3 のマルコフモデルを考慮して、4.3.1 の ‘A. Prediction of the PDTT’ は次の

ように書き換えることができる.

A'. Prediction of the PDTT

$$\begin{aligned}
 &\textbf{Given} : s_0, \gamma(t), \text{SDTT}, \\
 &\textbf{find} : x_t, \dots, x_{t+T}, \dots, x_{t+2T}, \\
 &\textbf{which maximize} : J = P(X_t = x_t, X_{t+1} = x_{t+1}, \dots, \\
 &\quad X_{t+T} = x_{t+T}, \dots, X_{t+2T} = x_{t+2T}), \\
 &\textbf{subject to} : \text{Markov model of Fig. 4.3.}
 \end{aligned} \tag{4.4}$$

4.3.3 マルコフモデルの状態遷移確率

時刻 t において状態 j から状態 i へ遷移する確率を

$$a_{ij}(t) = P(X_{t+1} = i | X_t = j) \tag{4.5}$$

と表現する. ただし, X_t は x_t の確率変数である. また, 準マルコフ過程を考えているため, 状態遷移確率は時変となる. この状態遷移確率は SDTT から同定される.

ここで, SDTT における n 列 m 行目の成分 (Table 4.1 参照) を c_{nm} とする. このとき, 状態遷移確率 $a_{i0}(t)$ は SDTT を用いて

$$\begin{aligned}
 a_{i0}(t) &= \frac{c_{\tau i}}{\sum_{i=0} c_{\tau i}}, \\
 &\quad \tau = t \bmod T
 \end{aligned} \tag{4.6}$$

と表現できる. ただし, τ は Table 4.1 における一日の時刻を表す変数である. $i > 0$ のとき, つまり車が走行状態であるときは残りの走行時間は時間の経過とともに減少する. それ故, この場合の状態遷移確率 $a_{ij}(t)$ ($i > 0, j > 0$) は

$$a_{ij}(t) = \begin{cases} 1 & i = j - 1 \\ 0 & i \neq j - 1. \end{cases} \tag{4.7}$$

と表現できる. よって, 状態遷移確率 $a_{ij}(t)$ は式 (4.6) と (4.7) を用いて表現される.

マルコフモデルの初期状態確率については Fig. 4.3 のモデルと観測情報 $\gamma(t), s_0$ (現在の車の走行状態と過去直近の出発時刻) に基づき計算される. Fig. 4.4 は $\gamma(t) = 1$ である場合, つまり車が走行状態にある場合の初期状態確率の計算例を示している. このとき, 車は時刻 $s_0 - 1$ と s_0 ($s_0 < t$) の間に出発している. この例では, 時刻 $s_0 - 1$ において状態 0 であり, 時刻 s_0 から t までの間状態が 0 でないことが観測情報からわかる. そのため, 実行可能な状態遷移のパスは Fig. 4.4 のように限定される. この状態遷移のパスの尤度を計算することにより, 初期状態確率を決定する.

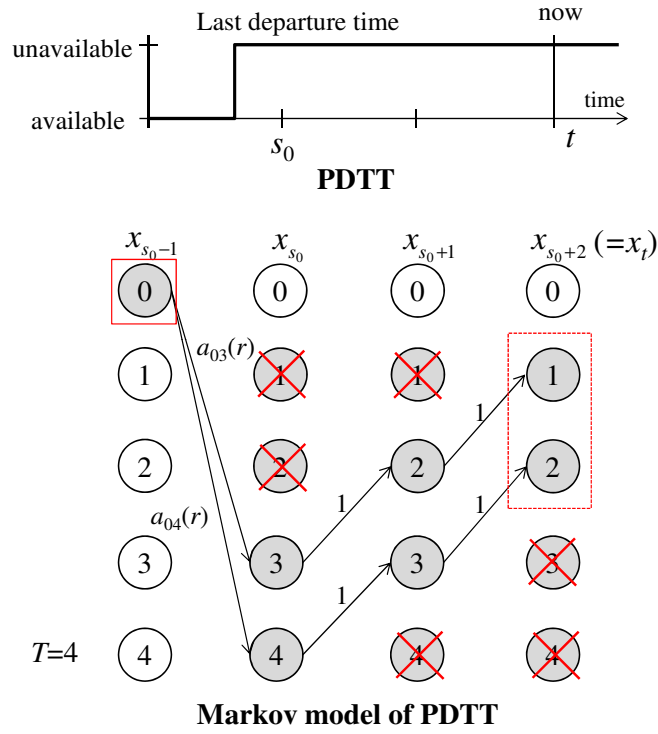


Fig.4.4 Initial state probabilities based on the observation

状態 $x_t \in \{0, 1, \dots, T\}$ における初期状態確率を π_{x_t} と表記する．車が走行状態にある場合 ($\gamma(t) = 1$)，初期状態確率 π_{x_t} は

$$\begin{aligned} \pi_{x_t} &= P(X_{s_0-1} = 0, X_{s_0} \neq 0, X_{s_0+1} \neq 0, \dots, X_t \neq 0) \\ &= \begin{cases} \frac{a_{(i+t-s_0)0}(r)}{T}, & 0 \leq i \leq T-t+s_0 \\ \sum_{j=t-s_0} a_{j0}(r) & \\ 0, & T-t+s_0 < i \leq T \end{cases} \end{aligned} \quad (4.8)$$

となる．一方，車が家庭にある場合 ($\gamma(t) = 0$) は残りの走行時間は 0 となる．それ故，初期状態確率 π_{x_t} は次のようになる．

$$\pi_0 = 1. \quad (4.9)$$

時刻 $t+T+1$ から $t+2T$ までの状態遷移について考える．これら状態遷移のパスは分岐が存在しないため，時刻 $t+T$ の状態によってユニークに決定される． $x_{t+T} = i \in [0, T]$ のとき，状態 x_{t+T} から状態 x_{t+2T} までのパスの同時生起確率は

$$\epsilon_i(t+T) = \begin{cases} \prod_{\tau=t+T+i}^{t+2T-1} a_{00}(\tau) & 0 \leq i \leq T-1 \\ 0 & i = T \end{cases} \quad (4.10)$$

と計算できる．

4.3.4 動的計画法に基づく最尤推定

この節では前節のマルコフモデルと動的計画法に基づく最尤推定問題について述べる.

Fig. 4.3 の Left-to-Right 型のマルコフモデルを考えると式 (4.4) の目的関数は

$$J = \pi_{x_t} a_{x_{t+1}x_t}(t) a_{x_{t+2}x_{t+1}}(t+1) \cdots a_{x_{t+T}x_{t+T-1}}(t+T-1) \epsilon_{x_{t+T}}(t+T), \quad (4.11)$$

となり, 最尤値 J^* は

$$J^* = \max_{x_t, \dots, x_{t+T}} [\pi_{x_t} a_{x_{t+1}x_t}(t) a_{x_{t+2}x_{t+1}}(t+1) \cdots a_{x_{t+T}x_{t+T-1}}(t+T-1) \epsilon_{x_{t+T}}(t+T)]. \quad (4.12)$$

と表現できる. この最尤値をマルコフモデルの構造を用いて考えると

$$J^* = \max_{x_{t+T}} a_{x_{t+T}x_{t+T-1}}(t+T-1) \epsilon_{x_{t+T}}(t+T) \left\{ \max_{x_t, \dots, x_{t+T-1}} [\pi_{x_t} a_{x_{t+1}x_t}(t) a_{x_{t+2}x_{t+1}}(t+1) \cdots a_{x_{t+T-1}x_{t+T-2}}(t+T-2)] \right\}. \quad (4.13)$$

のように最大化演算を分割して表現できる. ここで, 次の中間変数 $\delta_\tau(x_\tau)$ を導入する.

$$\delta_\tau(x_\tau) = \begin{cases} \pi_{x_t} & \tau = t \\ \max_{x_{\tau-1}} [\delta_{\tau-1}(x_{\tau-1}) a_{x_\tau x_{\tau-1}}(\tau-1)] & \tau \geq t+1, \end{cases} \quad (4.14)$$

また, 次の変数

$$\psi_\tau(x_\tau) = \arg \max_{x_{\tau-1}} [\delta_{\tau-1}(x_{\tau-1}) a_{x_\tau x_{\tau-1}}(\tau-1)]. \quad (4.15)$$

も導入する. ただし, $\tau \geq t+1$ である. ここで式 (4.13) に式 (4.14) を代入すると, 最尤値は

$$J^* = \max_{x_{t+T}} [a_{x_{t+T}x_{t+T-1}}(t+T-1) \epsilon_{x_{t+T}}(t+T) \delta_{t+T-1}(x_{t+T-1})]. \quad (4.16)$$

と表現できる. これは式 (4.14) を $\tau = t$ から $\tau = t+T$ まで計算することにより, 最尤値 J^* を計算できることを示している. 最終状態 x_{t+T}^* は J^* を求めるための再帰計算により導出できる.

$$x_{t+T}^* = \arg \max_{x_{t+T}} [a_{x_{t+T}x_{t+T-1}}(t+T-1) \epsilon_{x_{t+T}}(t+T) \delta_{t+T-1}(x_{t+T-1})]. \quad (4.17)$$

最終状態以降の系列 ($\tau \geq t+T$) は分岐を含まないため,

$$x_{\tau+1}^* = \begin{cases} x_\tau^* - 1 & x_\tau^* \neq 0 \\ 0 & x_\tau^* = 0 \end{cases} \quad (4.18)$$

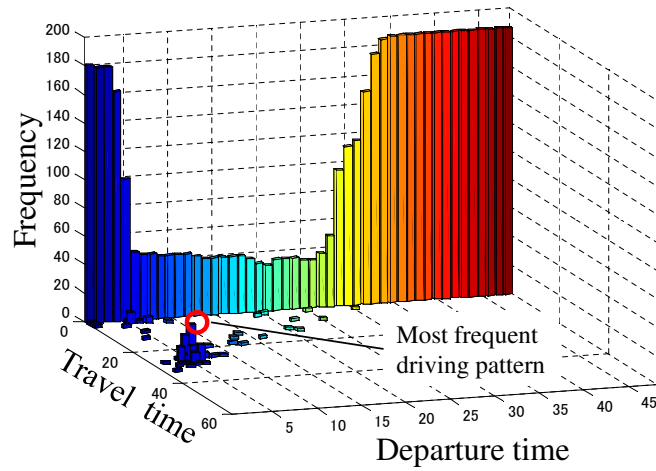


Fig.4.5 SDTT of the examinee for 188 days

となる．時刻 $t \leq \tau \leq t + T - 1$ の系列は式 (4.17) と (4.18) を用いて

$$x_{\tau}^* = \psi_{\tau+1}(x_{\tau+1}^*) \quad t \leq \tau \leq t + T - 1. \quad (4.19)$$

と計算できる．この計算により，4.3.2 節における PDTT の大域的最適解を求めることができる．

4.4 実験

提案手法の有用性を確認するため，本節では実験を通してその有用性を確認する．

4.4.1 実験条件

40 歳代の男性に対して 188 日間 SDTT のデータを取得した．被験者は主に通勤のために車を使用した．被験者の車一台に位置情報と時間が計測できる GPS ロガーを取り付けた．Fig. 4.5 は $T = 48$ としたときの，被験者の SDTT を表している．

4.4.2 実験結果

PDTT の予測について検討するため，Fig. 4.6 の三つのパターンを検証用データとして用いた．なお，車として利用している状態 (available) を 1 として表現している．パターン 1 は被験者の SDTT の最頻のパターンである (頻度は 11)．また，パターン 2 は準最頻のパターンである (頻度は 8)．パターン 3 はほとんど観測されなかったパターンである (頻度は 1)．

パターン 1 を観測として用いた場合の予測プロセスの結果 ($t=1, 2, \dots, 9$) を Fig. 4.7 に示す．Fig. 4.7 では検証用データを実線，予測結果を破線で表示しており，すべての時

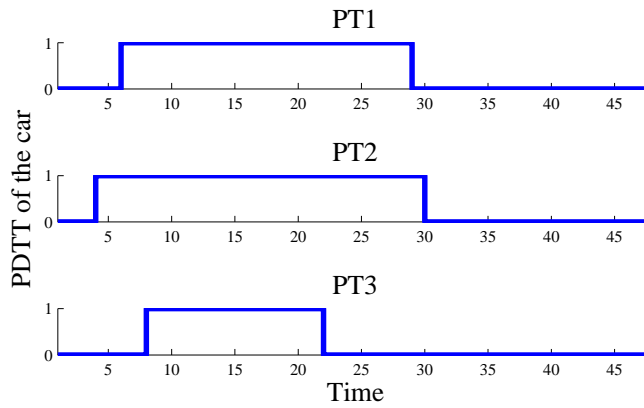


Fig.4.6 Typical patterns of the PDTT

Table 4.2 The average agreement rate for one day

Pattern of PDTT	Pattern 1	Pattern 2	Pattern 3
Agreement rate	1	0.96	0.76

Table 4.3 Calculation time of proposed method and full search method

Searching method	Full search method	Proposed method
Calculation time	7.9×10^4 sec	8.0×10^{-3} sec

間帯において完全に一致していることがわかる。

Fig. 4.8 はパターン 2 における予測結果を表している．時刻 $t = 1$ から $t = 3$ までの時間では検証用データは異なった予測をしている．一方，時刻 $t = 4$ では現在の新たな観測 $\gamma(4)$ と s_0 が得られたため，PDTT が再予測され，検証用データと同じ予測がなされている．Fig. 4.9 ではパターン 3 における予測結果を表しており，パターン 1, 2 の場合の結果と比較すると検証用データと異なる予測がなされていることがわかる．本手法では最尤法に基づき予測を行っているため，過去に観測される頻度が少ないパターンに対しては，正しく予測することは難しい．

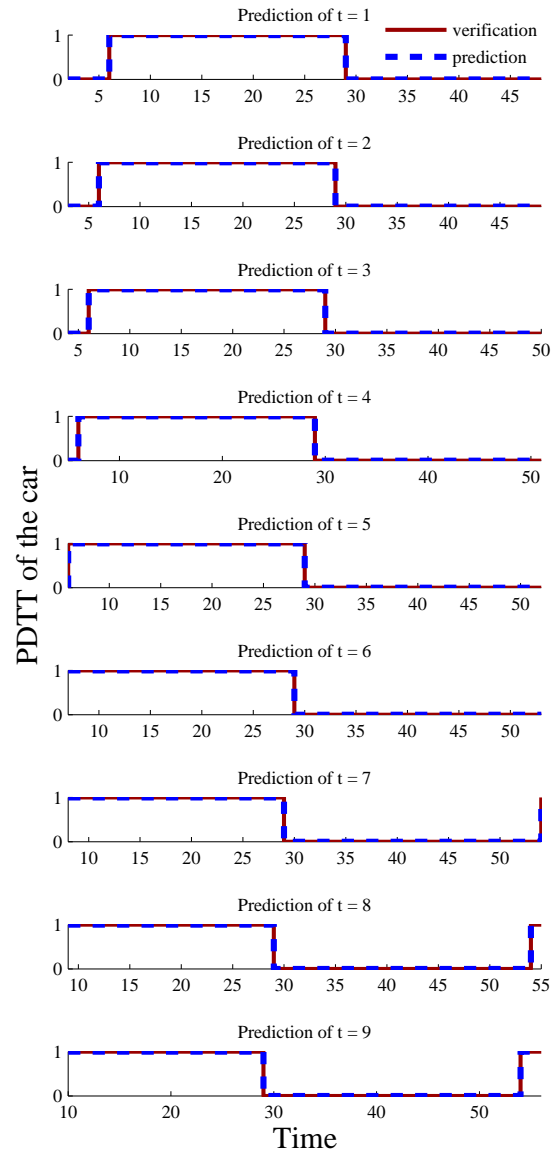
検証用データと予測結果の一致率を $\delta(t)$ とし，次のように計算した．

$$\delta(t) = \frac{\# \text{ of steps the verif. and pred. value match in one day}}{\text{Total \# of steps of one day}(= T)}, \quad (4.20)$$

また，一日の各時刻 $t = 0 \sim T$ において PDTT を予測しているため， $\delta(t)$ の時間平均を

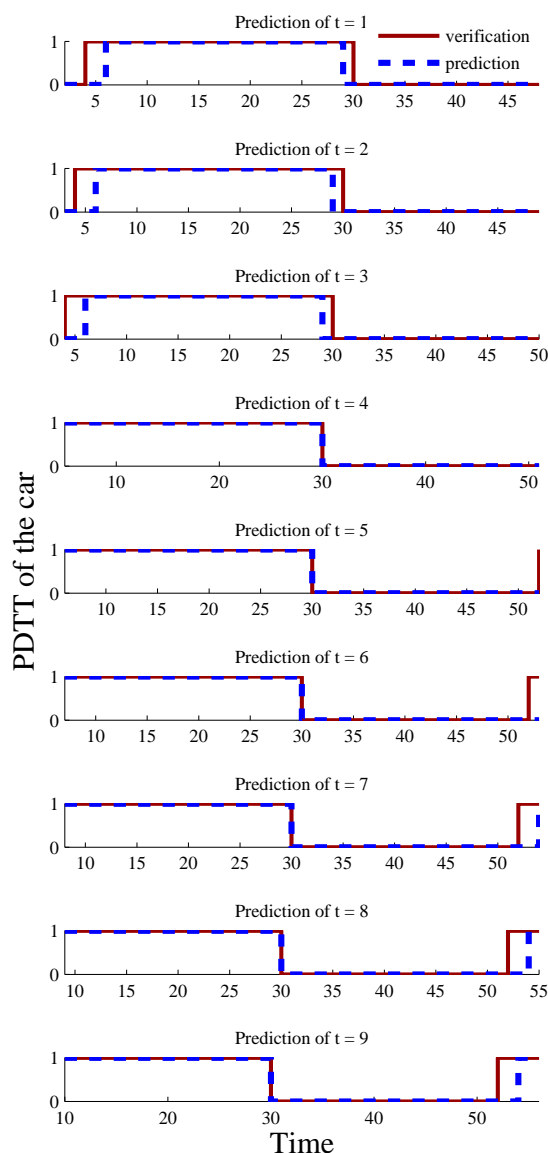
$$\Delta = \frac{1}{T} \sum_{t=1}^t \delta(t). \quad (4.21)$$

と計算した．Table 4.2 はパターン 1, 2, 3 の Δ を示している．この結果により，観測頻度の高い PDTT である程，予測の精度が高くなる傾向がある．


 Fig.4.7 Prediction and verification of Pattern 1 from $t = 1$ to 9

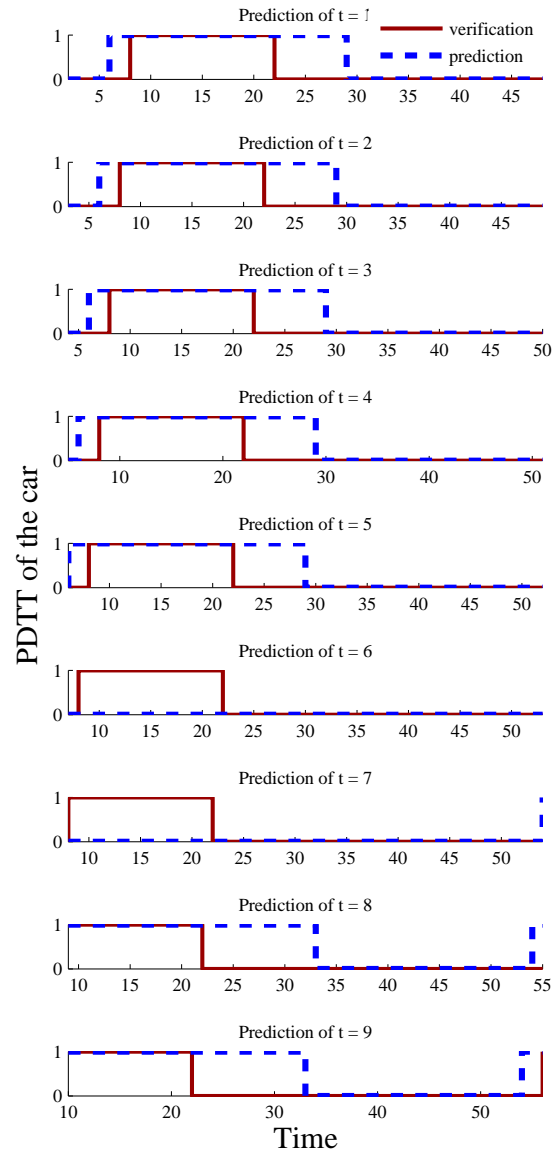
また Table 4.3 は提案手法を用いた場合の計算時間と全探索手法を用いた場合の計算時間を表している．全探索手法では，全ての実現値に対して条件式 (4.3) を考慮しながら評価関数 (4.1) を最大化するパターンを探索するため，多くの時間を要することがわかる．一方，提案手法では動的計画法を用いているため，少ない計算時間で済むことがわかる．

これらの結果から SDTT と観測情報をもとに PDTT を予測できることがわかる．また，動的計画法を用いることにより，リアルタイムに計算することが十分可能な時間で計算できることがわかる (今回の実験では 1 ステップを 30 分とした)．

Fig.4.8 Prediction and verification of Pattern 2 from $t = 1$ to 9

4.5 本章のまとめ

評価関数 (4.1) に対する最尤推定問題では、組み合わせ的な問題を持つため、評価関数の状態空間がとても大きい。そのため、評価関数の学習や最尤推定の計算は容易ではない。上記の点を踏まえ、本章では PDTT(Profile of Departure and Travel Time) の傾向を SDTT(Statistics of Departure and Travel Time) として表現した。この表現により、統計データを小さな状態空間上で表現することができた (一日を 48 ステップに分割して


 Fig.4.9 Prediction and verification of Pattern 3 from $t = 1$ to 9

表現したとき、状態空間の大きさは 2^{48} から 48×49 の大きさまで削減することができる). また、PDTT に対して準マルコフ過程を用いてモデル化することにより、SDTT から評価関数を計算することが可能となり、動的計画法を適用することができた. 動的計画法を適用することにより、評価関数 (4.1) の最尤推定の計算をリアルタイムに計算することができた.

本章では、過去蓄積されたデータから SDTT を作成し、SDTT を用いてリアルタイムに最尤となる PDTT を予測する手法を提案した. しかしながら、この予測手法を EMS(Energy Management System) に適用した場合にどのような結果が得られるか評価

されていない．そのため，この提案手法を EMS に適用し，予測手法を評価することが今後の課題として挙げられる．

第 5 章

結論

5.1 本研究の成果

本研究は事象駆動型システムに対して準マルコフ過程を適用し、そのモデルを用いて故障診断や未来の走行予測など確率推論を行った。

準マルコフ過程は生起するイベントの順序だけではなく、その生起時間間隔を表現できるため、他のモデルでは表現できない性質を検討することができる。例えば、2 章や 3 章における故障診断では故障としてアクチュエータなどの速度低下を取り扱っている。これは、生起するイベント順序列では判別できない故障であり、ペトリネットやオートマトンなどのモデルでは診断することができない。このように、準マルコフ過程ではその表現能力の高さゆえ、多くの対象の問題を取り扱うことができる。

しかしながら、その表現能力の高さはモデルの複雑さの起因となっており、計算量や学習量の多さが問題となる場合が多い。計算量や学習量の多さが問題となる場合は分散型手法が有用なアプローチである。本研究では、システムの構造やモデルの構造などをいくつかの部分問題に分割し、部分問題を用いて本問題を記述することにより、計算量や学習量の多さの解決を計った。

本研究では以下の三点を中心とした議論を展開し、事象駆動型システムを準マルコフ過程として表現することの有用性について検証した。

1. 確率理論を用いた事象駆動型システムの状況の推定
2. 準マルコフ過程を用いた予測器の設計
3. 観測データに基づく分散型手法の適用

5.1.1 確率理論を用いた事象駆動型システムの状況の推定

2 章、3 章において確率理論を用いた事象駆動型システムの故障の推定を行った。

2章では事象駆動型システムをいくつかのサブシステムから構成されるシステムであるとみなし、故障の推論問題を定式化した。システム全体を小さなサブシステムとして分割できるため、計算量や学習量の削減が期待できる。しかし、サブシステム間の関係を記述しているベイジアンネットワークは診断精度や学習量に大きな影響を与えてしまう。そこで、観測されたデータに基づき適切なネットワークを構築することにより、計算精度を落とすことなく計算量を削減できることを示した。

3章では事象駆動型システムの動作の順序性に着目し、システムの動作をいくつかのプロセスに分割して表現した。この表現を用いることにより、生起するイベントの順序性を陽に表現することができ、準マルコフ過程におけるイベントの組み合わせ問題を軽減することが期待できる。また、順序性をモデルに組み込んだことにより、モデルの複雑さを軽減することができ、診断精度を向上させる効果があることがわかった。

5.1.2 準マルコフ過程を用いた予測器の設計

4章において準マルコフ過程を用いた予測器の設計を行った。

4章では予測器としての準マルコフ過程を検証した。まず、車の走行状態を駐車状態と走行状態の二状態に分け、車の走行パターンをそれらのバイナリ系列として表現した。この表現を用いて、車の走行予測問題を最尤推定問題として定式化した。しかし、この問題は組み合わせ問題の性質を持っているため、計算量が問題となる。計算量を削減するため、車の走行動作に対して準マルコフ過程を適用し、マルコフモデルを用いて表現した。そして、問題をいくつかの部分問題に分割する動的計画法を用いて計算することにより、リアルタイムに車の予測をすることができることを示した。

5.1.3 観測データに基づく分散型手法の適用

2章、3章、4章において観測データに基づく分散型手法を適用した。

2章ではシステムの構造をいくつかのサブシステムに分割することにより、計算量や学習量の削減を計った。3章ではシステムの動作に着目し、その動作をいくつかのプロセスとして分割して表現することにより、計算量や学習量の削減を計った。4章では予測問題における最適化問題をいくつかの部分問題に分割し、動的計画法を適用した。これらの分散構造の因果関係はベイジアンネットワークなどを用いて適切に表現する必要がある。

以上のような結果から、事象駆動型システムに対して準マルコフ過程を適用することにより、多くの問題を取り扱うことができ、また他のモデルでは検証できない性質を確率推論を用いて検証できることを示した。また、準マルコフ過程において問題となる学習量や計算量の多さに対処するために分散型手法を適用した。そして、その分散構造を観測データに基づき適切に表現することにより、現実的な問題として計算できることを示した。

5.2 今後の課題

本研究では離散事象システムの動作を事象駆動型システムとして捉え、準マルコフモデルを用いて確率推論を行った。しかしながら、離散事象システムの分野は幅広いため、いくつかの課題が残されている。

1. **人間行動への適用** 人間の行動も離散的な意図をもつ場合があり、離散事象システムとして捉えることができる。また、人間の行動は機械の動作に比べてばらつきが大きく、決まった動作をしないことが多い。このような順序的あるいは時間的にばらつきのある対象は確率推論により議論できることが多いため、準マルコフ過程に適している。しかし、準マルコフ過程では確率の同定に多くの学習データを必要とする。そのため、学習データの収集には、データのサンプリングに大きな負担がかかる人間の場合では難しいことが多い。この学習データに関する問題を解決することにより、適用できる対象が今後は大きく広がると考えられる。
2. **他のシステムへの適用** 本研究では、故障診断の実験対象として自動搬送ライン、予測問題の対象として車の走行に適用した。しかしながら、故障診断手法では時間つきイベントを観測できる対象であれば適用でき、人間の意図推定、事象駆動型システムの隠れ状態推定などにも適用できる。また、予測問題手法では状態が二状態であるシステムに対しては適用できるため、部屋などの使用行動に対して適用できる。このような対象では観測されるデータが異なるため、異なった知見が得られることが予測される。
3. **より大規模なシステムへの適用** 本研究で用いたシステムは小さなシステムであるため、計算量や学習量に比較的余裕がある。そのため、大規模・複雑なシステムに対して準マルコフ過程を適用することにより、本研究では問題とならなかった箇所が問題となることが予想される。そこで、新たな問題に対して分散型手法を適用し、現実的な問題として解くことができるかを検証することが必要である。

謝辞

本研究を進めるにあたり、多大な御指導と有益なる御教示を賜りました名古屋大学大学院 工学研究科 機械理工学専攻 鈴木 達也 教授、同 稲垣 伸吉 講師、田崎 勇一 助教、また 情報科学研究科 奥田 裕之 研究員に心から御礼申し上げます。先生方には、研究の方針、学会発表など、あらゆる面において貴重な御助言をいただきました。

本論文をまとめるにあたり、数々の御助言を賜りました、名古屋大学大学院工学研究科 機械理工学専攻 早川 義一 教授、大阪大学大学院工学研究科電気電子情報工学専攻 高井 重昌 教授に、厚くお礼申し訳上げます。

加えて、4 章に関しましては、株式会社デンソー 金森 淳一郎 様、藤田 充 様、伊藤 章 様 は研究において尽力をいただきました。

そして、研究生生活を充実したものにしてくださった研究室の皆様に感謝いたします。

参考文献

- [1] Christos G. Cassandras and Stephane Laforune. *Introduction to Discrete Event Systems*. Springer Science+Business Media, LLC, Boston, MA, 2008.
- [2] 児玉慎三. 離散事象システム研究の動向と課題. 計測と制御, Vol. 31, No. 1, pp. 208–213, January 1992.
- [3] 守屋悦朗. 形式言語とオートマトン. Information science & engineering. サイエンス社, 2001.
- [4] 小倉久和. 形式言語と有限オートマトン入門: 例題を中心とした情報の離散数学. コロナ社, 1996.
- [5] 熊谷貞俊, 薦田憲久. ペトリネットによる離散事象システム論. コロナ社, 1995.
- [6] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541–580, April 1989.
- [7] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, Vol. 25, No. 1, pp. 206–230, January 1987.
- [8] 離散事象システム研究専門委員会. ペトリネットとその応用. 計測自動制御学会, 1992.
- [9] 羽鳥裕久, 森俊夫. 有限マルコフ連鎖. 培風館, 1982.
- [10] Rinald B. Schinazi. マルコフ連鎖から格子確率モデルへ: 現代確率論の基礎と応用. シュプリンガーフェアラーク東京, 2001.
- [11] 鈴木達也. Programmable logic controller に対するシステム論的考察—ハイブリッド系からのアプローチ. システム/制御/情報, Vol. 46, No. 3, pp. 144–149, 2002.
- [12] Frank Schiller, Jochen Schröder, and Jan Lunze. Diagnosis of transient fault in quantised systems. *Engineering Applications of Artificial Intelligence*, Vol. 14, No. 1-2, pp. 519–536, 2001.
- [13] Adnan Darwiche and Gregory Provan. Exploiting system structure in model-based diagnosis of discrete-event systems, 1996.
- [14] L. A. M. Riascos, L. A. Moscato, and P. E. Miyagi. Detection and treatment of faults in manufacturing systems based on petri nets. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 26, No. 3, pp. 280–289,

- 2004.
- [15] Meer Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, Vol. 40, No. 9, pp. 1555–1575, Sep 1995.
 - [16] Antonio Ramirez-Trevino, Elvia Ruiz-Beltran, Israel Rivera-Rangel, and Ernesto Lopez-Mellado. Online fault diagnosis of discrete event systems. a petri net-based approach. *IEEE T. Automation Science and Engineering*, Vol. 4, No. 1, pp. 31–39, 2007.
 - [17] 齋藤光生, 鈴木達也, 稲垣伸吉, 青木猛. 最大エントロピー原理を用いた時間付きマルコフモデルによる事象駆動系の故障診断. 計測自動制御学会論文集, Vol. 42, No. 9, pp. 1067–1075, Sep 2006.
 - [18] Jan.Lunze. "diagnosis of quantized system based on a timed discrete-event model". Vol. 30, pp. 322–335. IEEE, 2000.
 - [19] Fuchun Liu, Daowen Qiu, Hongyan Xing, and Zhujun Fan. Decentralized diagnosis of stochastic discrete event systems. *IEEE Trans. Automat. Contr.*, Vol. 53, No. 2, pp. 535–546, 2008.
 - [20] ビショップ CM. パターン認識と機械学習: 上. 第 3 巻. シュプリンガー・ジャパン, 2008.
 - [21] ビショップ CM. パターン認識と機械学習: 下. 第 3 巻. シュプリンガー・ジャパン, 2008.
 - [22] 岸本圭史, 小栗宏次. 直前の一定期間の運転行動を考慮した ar-hmm に基づく停止行動予測 (高度交通システム (its)). 電子情報通信学会論文誌. A, 基礎・境界, Vol. 92, No. 9, pp. 624–632, sep 2009.
 - [23] Ron Begleiter, Ran El-yaniv, and Golan Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, Vol. 22, pp. 385–421, 2004.
 - [24] Shiyao Chen and Lang Tong. iems for large scale charging of electric vehicles: Architecture and optimal online scheduling. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pp. 629–634, 2012.
 - [25] Michael C. Caramanis and Justin M. Foster. Management of electric vehicle charging to mitigate renewable generation intermittency and distribution network congestion. In *CDC*, pp. 4717–4722. IEEE, 2009.
 - [26] Soheil Shahidinejad, Eric Bibeau, and Shaahin Filizadeh. Statistical development of a duty cycle for plug-in vehicles in a north american urban setting using fleet information. *IEEE T. Vehicular Technology*, Vol. 59, No. 8, pp. 3710–3719, 2010.
 - [27] Ali Ashtari, Eric Bibeau, Soheil Shahidinejad, and Tom Molinski. Pev charging profile prediction and analysis based on vehicle usage data. *IEEE Trans. Smart*

-
- Grid*, Vol. 3, No. 1, pp. 341–350, 2012.
- [28] Mei Chen and Steven I. J. Chien. Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1768, pp. 157–161, 2001.
- [29] Chun hsin Wu, Jan ming Ho, and D. T. Lee. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, pp. 276–281, 2004.
- [30] Dick Ettema and Harry Timmermans. Modeling departure time choice in the context of activity scheduling behavior. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1831, pp. 39–46, 2003.
- [31] Dick Ettema, Fabian Bastin, John Polak, and Olu Ashiru. Modelling the joint choice of activity timing and duration. *Transportation Research Part A: Policy and Practice*, Vol. 41, No. 9, pp. 827 – 841, 2007. Selected Papers on Applications of Discrete Choice Models Presented at the European Regional Science Conference, Amsterdam, August 2005.
- [32] 齋藤光生, 稲垣伸吉, 鈴木達也, 青木猛. Bayesian network を用いた事象駆動型システムの分散型故障診断. 計測自動制御学会論文集, Vol. 43, No. 11, pp. 1065–1073, nov 2007.
- [33] O. Contant, S. Lafortune, and D. Teneketzis. Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems: Theory and Applications*, Vol. 16, No. 1, pp. 9–37, 2006.
- [34] Gregory F. Cooper and Edward Herskovits. A bayesian method for constructing bayesian belief networks from databases. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, UAI’91, pp. 86–94, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.
- [35] David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, Vol. 3, pp. 507–554, March 2003.
- [36] Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12*, pp. 505–511. MIT Press.
- [37] Jie Cheng, David A. Bell, and Weiru Liu. An algorithm for bayesian belief network construction from data. IN PROCEEDINGS OF AI & STAT’ 97, pp. 83–90, 1997.
- [38] Finn V. Jensen and Thomas D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, Incorporated, 2nd edition, 2007.
- [39] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Monographs in Computer Science. Springer, Jan

- 1997.
- [40] 繁桝算男, 植野真臣, 本村陽一. ベイジアンネットワーク概説. 培風館, 第3版, Jul 2006.
 - [41] 渡辺澄夫, 荻原克幸, 赤穂昭太郎, 本村陽一, 福永健次, 岡田真人, 青柳美輝. 学習システムの理論と実現, pp. 75–97. 森北出版, 2005.
 - [42] 本村陽一. ベイジアンネットワークによる確率的推論技術. 計測と制御, Vol. 42, No. 8, pp. 649–654, 2003.
 - [43] 田中和之. 確率的情報処理と統計力学. 臨時別冊・数理科学. サイエンス社, Sep 2006.
 - [44] W. A. J. J. Wiegerinck, H. J. Kappen, E. W. M. T. ter Braak, W. J. P. P. ter Burg, M. J. Nijman, Y. L. O, and J. P. Neijt. Approximate inference for medical diagnosis. *Pattern Recognition Letters*, Vol. 20, No. 11-13, pp. 1231–1239, 1999.
 - [45] F. Jensen, U. Kjaeruluf, B. Kristiansen, H. Langseth, C. Skaanning, J. Vomlel, and M. Vomlelova. The sacso methodology for troubleshooting complex systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, Vol. 15, pp. 321–333, 2004.
 - [46] John Mark Agosta and Thomas Gardos. Bayes network "smart" diagnostics. *Intel Technology Journal*, Vol. 8, No. 4, pp. 361–372, 2004.
 - [47] 河原吉伸, 矢入健久, 町田和雄. Dynamic bayesian networks を用いた宇宙機異常診断法. 人工知能学会論文誌, Vol. 21, No. 1, pp. 45–54, 2006.
 - [48] S. C. W. Ong, Surendra Ranganath, and Y. V. Venkathesh. Understanding gestures with systematic variations in movement dynamics. *Pattern Recognition*, Vol. 39, No. 9, pp. 1633–1648, 2006.
 - [49] Tomoyoshi Akiba and Hozumi Tanaka. A bayesian approach for user modeling in dialogue systems. In *the International Conference on Computational Linguistics*, pp. 1212–1218, Aug 1994.
 - [50] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Péér. Using bayesian network to analyze expression data. *Computational Biology*, Vol. 7, pp. 601–620, 2000.
 - [51] 坂元慶幸, 石黒真木夫, 北川源四郎. 情報量統計学, pp. 65–91. 情報科学講座, No. A.5.4. 共立出版, 1983.
 - [52] 村田昇. 情報理論の基礎, 第5章, pp. 67–90. 臨時別冊・数理科学. サイエンス社, Jan 2005.
 - [53] M. Srikanth, H. K. Kesavan, and P. H. Roe. Probability density function estimation using the minmax measure. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, Vol. 30, No. 1, pp. 77–83, 2000.
 - [54] J. N. Kapur, G. Baciú, and H. K. Kesavan. The minmax information measure. *International journal of systems science*, Vol. 26, No. 1, pp. 1–12, 1995.
 - [55] 辻井純一, 北研二. 確率的言語モデル. 言語と計算, No. 4. 東京大学出版会, 第3版,

Nov 1999.

ページ設定調整のための dummy ページ

Dummy ページ

このページは表紙ページを作るための, Dummy です.

LateX の使い方

デフォルトの設定パラメータの表示

この論文の設定パラメータは以下の通りです。

```
lzw = 9.24687pt
headsep = 17.07164pt
baselineskip = 16.0pt
textwidth = 369.87488pt
textheight = 602.1pt
headsep = 17.07164pt
evensidemargin=41.0pt
oddsidemargin=3.0pt
voffset=-20.0pt
vsize=602.1pt
```

このように、`\the` 命令で長さを表す変数の設定値を取り出します。

