

エネルギー関数の形状に着目した
焼きなまし法の適用に関する研究

学位論文

Doctral thesis

2018年12月26日

首都大学東京大学院システムデザイン研究科

博士後期課程

16990503 柴田祐樹

概要

本論文ではエネルギー関数の形状に着目した焼きなまし法の適用方法について提案し, Boltzmann Machine における事後分布推定, 観光経路推薦問題を対象としてその有効性を示す.

Boltzmann Machine や焼きなまし法等, 現在利用されている機械学習や最適化問題の解法は統計物理学に基づくものが多い. これらはいずれも物質を構成する分子の微視的状态を記述する Boltzmann 分布をもとに定式化され, Boltzmann Machine では与えられた分布に適したエネルギー関数を学習により決定する. 得られる分布は有用な潜在的表現を示すことから Neural Networks に対する事前学習手法としてよく利用される. また, Boltzmann 分布に従う系では, エネルギー関数の値が低い状態が高い出現確率を持ち, 温度が低いほどその傾向が強くなる. 焼きなまし法はこの性質を利用し, 最適化問題における目的関数をエネルギー関数とした手法である. 焼きなまし法を用いれば局所最適化法で問題となる局所最適解への収束性を緩和することができ, Traveling Salesman Problem (TSP) 等さまざまな問題に対して適用されている.

焼きなまし法は, 計算量の多さと, 温度変化過程の与え方が自明でないことが短所として指摘されている. さらに, 本論文では, Deep Boltzmann Machine の事後分布推定に対して効果的でないこと, TSP に対しても問題の規模が大きくなるにつれ, Markov Chain Monte Carlo (MCMC) 法における棄却率が高まり, 効率が低下するといった問題点があることも明らかにする. これらの問題に対し, 本論文ではエネルギー関数の形状に着目することで, 焼きなまし法における上記問題の解決を目的とする. 従来手法が MCMC 法の探索方法を決定する提案分布に対し局所最適化法をそのまま用い, エネルギー関数には最適化問題として一般的に定義されていた目的関数をそのまま用いているのに対し, 提案手法では, 問題ごとに適切なエネルギー関数と, 棄却率の低い提案分布の構築を行う. 具体的な例として, Boltzmann Machine と観光経路推薦問題に提案手法を適用し, 有効性を示す.

Boltzmann Machine は階層構造を持たせた Deep Boltzmann Machine として広く用いられ, 事前学習やマルチモーダルデータ (Multimodal Data) に対するモデルとして近年注目を集めている. しかし, 現在一般に用いられている Gibbs Sampling 法を適用しただけでは, 層数の増加に伴い事後分布の推定が難しくなる. また, 本論文で示す通り, 焼きなまし法を適用しても精度を改善することはできない. 提案手法では, 階層構造を反映したエネルギー関数の構築と温度の制御を行うことで事後分布の推定精度を改善する.

観光経路推薦問題への適用においては, TSP に対して MCMC 法における提案分布に適切なものを選ぶことで探索効率を改善できることを示す. また, 観光経路推薦問題を対象として, 訪問スポットの選択も含むように TSP を拡張した選択的 TSP に対しては, 従来節点と辺を用いて定式化

されていたのに対し、辺のみを用い、さらに、自己回帰辺を導入することで次元数を固定とする定式化を提案し、確率場として問題を定義可能とする。また、経路長に対する制約を連続な関数で表現することで焼きなまし法を適用可能とする。これらの工夫により高速に解を探索可能となり、従来手法では固定とされていたスポット間の移動経路をユーザの嗜好に応じて決定する等の柔軟な経路推薦が可能となる。

本論文は5章からなり、1章で序論、2章では本論文の基礎となる関連研究について述べ、3章で Boltzmann Machine に対する焼きなまし法の適用法、4章で観光経路推薦問題に対する焼きなまし法の適用法を提案する。5章では本論文で論じたことをまとめる。

3章では、はじめに Boltzmann Machine により評価値行列から抽出される潜在因子の性質を調査した結果に基づき、事後分布推定における Boltzmann Machine の特性について考察する。その後、Deep Boltzmann Machine のエネルギー関数を層ごとに分割し、各層に対応した関数ごとに温度を割り当てることで温度分布を表現可能とする手法を提案する。これにより、種結晶から金属の単結晶を得る手法である Floating Zone 法を導入し、入力側から温度を低下させることで事後分布を高精度に近似できることを示す。

4章では、観光経路推薦問題の概要、および既存研究における定式化を紹介した後、本論文で提案する定式化について説明する。提案する定式化を用いて、エネルギー関数により経路長に対する制約を表現する方法、棄却率の低い提案分布を用いて MCMC 法を構築する方法を述べる。人工データセットおよび観光客の位置情報から作成されたデータセットを用いて、提案手法の有効性を示す。

本論文で得られた成果は、焼きなまし法を適用するにあたりエネルギー関数の形状を考慮することの重要性を示したものであり、対象とした Boltzmann Machine や、観光経路推薦問題だけでなく、より多様な用途における焼きなまし法の適用可能性の拡大へ貢献するものである。

目次

概要		i
第 1 章	序論	1
1.1	はじめに	1
1.2	本論文の構成	2
第 2 章	関連研究	5
2.1	基本的な数学的表記	5
2.1.1	用いるフォント	5
2.1.2	論理と集合	6
2.1.3	写像と族	7
2.1.4	グラフ	9
2.1.5	確率空間と確率変数	9
2.1.6	添字を用いた確率空間と確率変数の表記	10
2.2	標本抽出法 (Sampling Method)	14
2.3	Boltzmann 分布	16
2.3.1	定義	16
2.3.2	焼きなまし法	17
2.3.3	焼きなまし法における発見的手法の定式化	18
2.3.4	Boltzmann Machine	19
2.4	評価実験で用いる計算環境	20
第 3 章	Boltzmann Machine に対する焼きなまし法の適用	21
3.1	Boltzmann Machine を用いた潜在因子推定	22
3.1.1	はじめに	22
3.1.2	協調フィルタリング	23
3.1.3	本調査における定式化	24
3.1.4	評価実験	26
3.1.5	まとめ	30
3.2	焼きなまし法の結晶化効果	32
3.2.1	はじめに	32

3.2.2	焼きなまし法と Floating Zone 法	33
3.2.3	提案手法	33
3.2.4	評価実験	38
3.2.5	まとめ	43
第 4 章	観光経路推薦問題に対する焼きなまし法の適用	45
4.1	はじめに	45
4.2	辺ベクトルによる観光経路推薦問題の定式化	46
4.2.1	関連研究	46
4.2.2	辺ベクトルと Boltzmann 分布による定式化	47
4.2.3	TSP への適用と 2-opt 法	48
4.2.4	非存在辺の導入	50
4.2.5	焼きなまし法を適用可能な STSP の定式化	52
4.2.6	Boltzmann 定数と提案分布	54
4.2.7	評価実験	55
4.2.8	まとめ	61
4.3	非存在辺と実在辺の対称化による目的関数の改良	62
4.3.1	提案する定式化と対称性に関する考察	63
4.3.2	評価実験	66
4.4	冗長経路の導入によるスポット滞在時間の表現	68
4.4.1	はじめに	68
4.4.2	提案手法	68
4.4.3	評価実験	69
第 5 章	終わりに	73
付録 A	Einstein の縮約記法	75
付録 B	諸式の導出	79
B.1	Metropolis 基準	79
B.2	Boltzmann 分布と経験分布の相対情報量の勾配	80
B.3	温度分布を持つ Boltzmann Machine の条件付き確率	81
B.4	逆写像の共通部分	82
付録 C	Newton-Raphson 法と勾配降下法	83
参考文献		85
発表文献		91
謝辞		91

第 1 章

序論

1.1 はじめに

Boltzmann Machine (BM)[37] や焼きなまし法 [41], Latent Dirichlet Allocation[69] 等, 現在利用されている機械学習 [18] や最適化問題の解法は統計物理学 [7] に基づくものが多い. 焼きなまし法, BM は共に物質の微視的状态を記述する Boltzmann 分布を元に定式化される. BM では与えられた分布に適したエネルギー関数を学習により決定し, 得られる分布は有用な潜在的表現を示すことから Neural Networks [13] に対する事前学習手法としてよく利用される. また, Boltzmann 分布に従う系では, エネルギー関数の値が低い状態が高い出現確率を持ち, 温度が低いほどその傾向が強くなる. 焼きなまし法はこの性質を利用し, 最適化問題における目的関数をエネルギー関数とした手法である. 焼きなまし法を用いれば局所最適化法で問題となる局所最適解への収束性を緩和することができ, Traveling Salesman Problem (TSP) [9] 等さまざまな問題に対して適用されている.

焼きなまし法は, 計算量の多さと, 温度変化過程の与え方が自明でないことが短所として指摘されている [45]. さらに, 本論文では, Deep Boltzmann Machine (DBM) [38] の事後分布推定に対して効果的でないこと, TSP に対しても問題の規模が大きくなるにつれ, Markov Chain Monte Carlo (MCMC) [7] 法における棄却率が高まり, 効率が低下するといった問題点があることも明らかにする. これらの問題に対し, 本論文ではエネルギー関数の形状に着目することで, 焼きなまし法における上記問題の解決を目的とする. 従来手法が MCMC 法の探索方法を決定する提案分布に対し局所最適化法をそのまま用い, エネルギー関数には最適化問題として一般的に定義されていた目的関数をそのまま用いているのに対し, 提案手法では, 問題ごとに適切なエネルギー関数と, 棄却率の低い提案分布の構築を行う. 具体的な例として, BM と観光経路推薦問題 [10][26][27][29] に提案手法を適用し, 有効性を示す.

BM は階層構造を持たせた DBM として広く用いられ, 事前学習やマルチモーダルデータに対するモデルとして近年注目を集めている [52]. しかし, 現在一般に用いられている Gibbs Sampling 法 [44] を適用しただけでは, 層数の増加に伴い事後分布の推定が難しくなる [38]. また, 本論文で示す通り, 焼きなまし法を適用しても精度を改善することはできない. 提案手法では, 階層構造を反映したエネルギー関数の構築と温度の制御を行うことで事後分布の推定精度を改善する. 観光経路推薦問題への適用においては, TSP に対して MCMC 法における提案分布 [24] に適切なものを

選ぶことで探索効率を改善できることを示す。

観光経路推薦問題では、訪問すべきスポットの決定、およびスポット間の経路の両方を決定する必要があり、訪問スポットの選択を含むように TSP を拡張した選択的 TSP (Selective TSP: STSP) が提案されている [26]。STSP では、節点と辺を用いた定式化が行われており [10][27][29] 2つの異なる要素を扱うことから最適化手法の構築が難しくなっている。これに対し本論文では、辺のみを用い、さらに、自己回帰辺を導入することで次元数を固定とする定式化を提案し、確率場として問題を定義可能とする。また、経路長に対する制約を連続な関数で表現することで焼きなまし法を適用可能とする。これらの工夫により高速に解を探索可能となり、従来手法では固定とされていたスポット間の移動経路をユーザの嗜好に応じて決定する等の柔軟な経路推薦が可能となる。

本論文で得られた成果は、焼きなまし法を適用するにあたりエネルギー関数の形状を考慮することの重要性を示したものであり、対象とした BM や、観光経路推薦問題だけでなく、これら以外の確率的モデル、最適化問題等への多様な用途における焼きなまし法の適用可能性の拡大へ貢献するものである。

1.2 本論文の構成

本論文は 5 章からなり、その構成を以下に示す。

第 1 章 序論

第 2 章 関連研究

本論文で用いる数学的表記法、Boltzmann 分布、MCMC 法について説明する。その後 Boltzmann Machine (BM)、焼きなまし法、焼きなまし法を用いた最適化問題の解法と関連研究を紹介する。

第 3 章 Boltzmann Machine に対する焼きなまし法の適用

はじめに BM により評価値行列から抽出される潜在因子の性質を調査した結果に基づき、事後分布推定における BM の特性について考察する。その後、Deep Boltzmann Machine のエネルギー関数を層ごとに分割し、各層に対応した関数ごとに温度を割り当てることで温度分布を表現可能とする手法を提案する。さらに、種結晶から金属の単結晶を得る手法である Floating Zone 法を導入し、入力側から温度を低下させることで事後分布を高精度に近似できることを示す。

第 4 章 観光経路推薦問題に対する焼きなまし法の適用

観光経路推薦問題の概要、および既存研究における定式化を紹介した後、本論文で提案する定式化について説明する。提案する定式化を用いて、エネルギー関数により経路長に対する制約を表現する方法、棄却率の低い提案分布を用いて MCMC 法を構築する方法を述べる。人工データセットおよび観光客の位置情報から作成されたデータセットを用いて、提案手法の有効性を示す。

第 5 章 終わりに

本論で述べたことを総括的にまとめる。

式，図，表に振られる番号には章番号を含む階層的なものを利用している．本論文中で直接記述すれば論述が煩雑となる諸定理について，重要なものについては付録に記載した．

本論文で用いる主要な記号を表 1.1 に記載する．各章で局所的に用いる記号に対してはそれら章のはじめに個別の記号表を載せる．

表 1.1 本論文で用いる記号の表

記号	定義	意味	場所
\emptyset	$\{ \}$	空集合	
\mathbb{N}		自然数の集合	
\mathbb{R}		実数の集合	
\mathcal{N}		節点の集合	2.1.4 節
\mathcal{E}_p		辺の全体集合	2.1.4 節
\mathcal{E}	$\subset \mathcal{E}_p$	実在辺の集合	2.1.4 節
Λ		確率空間全体に対する添字の集合	2.1.6 節
n_{st}	$\mathcal{E}_p \rightarrow \mathcal{N}$	辺から始点の節点への全射	2.1.4 節
n_{ed}	$\mathcal{E}_p \rightarrow \mathcal{N}$	辺から終点の節点への全射	2.1.4 節
d	$\Lambda \rightarrow \mathcal{N}$	BM の状態変数の添字から節点への全単射	式 (2.30)
c	$\Lambda^2 \rightarrow \mathcal{E}$	BM の状態変数の添字対から辺への全射	式 (2.30)
Ω_Λ		Λ で識別される確率空間の標本空間	2.1.6 節
Ω'_M		M ($\subset \Lambda$) 上に定義される確率変数の状態空間	2.1.6 節
\mathbf{x}_M	$\Omega_\Lambda \rightarrow \Omega'_M$	M ($\subset \Lambda$) 上に定義される確率変数	2.1.6 節
f_σ	$x \in \mathbb{R} \mapsto 1/(1 + \exp(-x))$	Sigmoid 関数	
$\arg \min$		最小値を与える引数の集合を返す関数	式 (2.2)
δ_{ij}	$\in \{0, 1\}$	Kronecker Delta	式 (2.1)
δ	$\{T, F\} \rightarrow \{1, 0\}$	Kronecker Delta の一般形	式 (2.1)
ϕ	$\Omega'_\Lambda \rightarrow \mathbb{R}$	エネルギー関数および目的関数	2.3.1 節
Z_p	$\in (0, \infty)$	分配関数	2.3.1 節
k	$\in (0, \infty)$	Boltzmann 定数	2.3.1 節
T	$\in (0, \infty)$	Boltzmann 分布に従う系の温度	2.3.1 節
T_H	$\in (0, \infty)$	焼きなまし法における最高温度	式 (2.47)
T_L	$\in (0, \infty)$	焼きなまし法における最低温度	式 (2.48)
t	$\in [0, \infty)$	時刻, 学習, 計算の進捗度	式 (2.58)
α_{dec}	$\in [0, \infty)$	重み減衰係数	式 (2.59)
α_{mom}	$\in [0, \infty)$	慣性項係数	式 (2.60)
$\epsilon(t)$	$\mathbb{R} \rightarrow [0, \infty)$	学習率	2.3.4 節

第 2 章

関連研究

2.1 基本的な数学的表記

情報科学の分野においては物理や数学におけるモデルほど一般性の高いものばかりが用いられるわけではなく、今後より複雑なモデルが提案されていくことを考えれば、表記法を洗練していく必要があると考える。

本論文で用いる表記法は基本的に国際標準化機構 (ISO: International Organization for Standardization) の指針 (ISO80000-2)[1] に従う。本節では重要なものについて、具体例と関連付けながら説明を行う。また、本論文で扱う量は基本的に無次元量であるとし、物理量との対応がある場合のみ単位を SI 単位系 [2] により記述する。

2.1.1 用いるフォント

諸定数、変数、関数の表記に用いるフォントは以下の基準に従い選ばれる。

- 決して変わることのない定数、よく知られた関数には Roman 体を用いる。たとえば Napier 数 $e = 2.71821\dots$ 、円周率 $\pi = 3.1415\dots$ 、指数関数 \exp 、自然対数 \ln 等である。
- 論文中で新たに定義した関数、定数には Italic 体を用いる。
- 変数には Italic 体を用いる。添字についてはそれが変数の場合には Italic 体を、意味的添字の場合には立体を用いる。例えば利用者 u の持つ要素集合を意味するような変数は、添字も変数であるため C_u と表すが、利用者一般ではなく山田氏 (Yamada) の持つ要素集合であれば C_{Yamada} と表す。注意が必要な例として、座標軸の表し方がある。座標の値を示すとき、それは変数であるので x, y, z と表すが、座標軸に割り当てられる変数は座標軸が意味的添字となるため P_x, P_y, P_z と表記する。これは先程の利用者一般と山田氏が持つ要素集合の例と対応している。
- ISO80000-2 の指針にはないが、変数間の積記号は慣習的に省略され、本論文でも積記号を基本的に省略する。
- 2, 3 の指針から関数名、変数名を Italic 体で表記するが、4 の慣習を併用するため、Italic 体の文字が連続している場合にそれらが異なる変数間の積を意味することにもなり、誤解を招きやすい。従い、変数、関数は 1 文字で表記する。2 文字以上によって表記する場合は添

表 2.1 真理値表

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

字を用いる。たとえば、ある変数 x と y の間の類似度を表す関数に対し、 $Similarity(x, y)$ のような関数名は用いない。よく知られた関数であれば、Roman 体により $Similarity(x, y)$ とすることができるが、論文中で新たに定義した関数については $f_{Similarity}(x, y)$ のように Roman 体で書かれた添字を用い表記する。

6. 演算子には立体を用いる。たとえば p の x による微分は $\frac{dp}{dx}$ 、積分は $\int p dx$ と書く。
7. 確率変数の表記には慣習に従い太字 Roman 体を用いる [18]。たとえば x 軸に分布する確率変数の表記には \mathbf{x} などを用いる。

2.1.2 論理と集合

命題とは T(真), F(偽) のどちらかの値をとる論述もしくは式のことである。それ以上分割できない命題を要素命題と呼び、論理演算子により要素命題を組み合わせて表現したものを複合命題と呼ぶ。論理演算子の真理値表を表 2.1 に示す。なお、論理演算子の結合の優先度は四則演算子よりも低いものとする。また、命題 p が成り立つことを述べる時、「 p が真である」と記述することも考えられるが、本論文では単に「 p である」等の平叙文により記述する。 $p \wedge q$ を単に p, q と書く表現はよく用いられ、本論文でも特に曖昧でない箇所にはこの関係を $p \wedge q \Leftrightarrow p, q$ と定め利用する。「命題 $P(a)$ を満たす要素 a について」という論述は基本的に $a \in P(a)$ と書くが、 $P(a)$ が a を左辺に持つ 2 項演算、たとえば $P(a) \Leftrightarrow a \geq b$ であるような場合、 $a (\geq b)$ とする表記も用いる。まとめると、以下の論述は互いに同値である。

- $x \geq 0$ を満たす実数 x について考える
- 実数 x ($x \geq 0$) について考える
- 実数 x (≥ 0) について考える
- x ($x \geq 0 \wedge x \in \mathbb{R}$) について考える

要素に対する命題 $P(a)$ が与えられた時、 $A := \{a | P(a)\}$ の表記は $P(a) = T$ となるすべての要素 a から集合 A が成ることを示す。加算集合に限っては、 A を n 個の要素 a_1, a_2, \dots, a_n を用いて直接 $A := \{a_1, a_2, \dots, a_n\}$ と表現することも可能である。

$a \in A$ の表記は a が A に含まれることを示す。 $a \notin A$ は $\neg(a \in A)$ と同値であり、 a が A に含まれないことを意味する。また、複合命題に関して、以下の記法を用いる。

- $\forall a \in A, P(a)$: すべての $a (\in A)$ について命題 $P(a)$ が成り立つ。

- $\exists a \in A, P(a)$: 少なくともひとつの a ($\in A$) について命題 $P(a)$ が成り立つ.
- $\nexists a \in A, P(a) \Leftrightarrow \neg(\exists a \in A, P(a))$.

同じ集合に属する 2 変数の命題については $\forall a, b \in A, P(a, b)$ などとも書く.

集合 A と集合 B の和集合を $A \cup B := \{x | x \in A \vee x \in B\}$, 共通部分を $A \cap B := \{x | x \in A \wedge x \in B\}$, 差集合を $A \setminus B := \{x | x \in A \wedge x \notin B\}$ と定義し用いる. 集合 A の要素の数を $|A|$ と表す. ある集合 A の部分集合全体の集合はべき集合とよばれ, これを 2^A と表記する.

実数 a, b で定義される実数の部分集合について, 以下の表記を用いる.

- $[a, b] := \{x \in \mathbb{R} | a \leq x \leq b\}$
- $(a, b] := \{x \in \mathbb{R} | a < x \leq b\}$
- $[a, b) := \{x \in \mathbb{R} | a \leq x < b\}$
- $(a, b) := \{x \in \mathbb{R} | a < x < b\}$

2.1.3 写像と族

$f: A \rightarrow B$ の表記は f が集合 A から集合 B への写像であることを意味する. またこのとき, 以下の表記法を定義する.

- f^{-1} は B から A への f の逆写像を表し, $\forall b \in B, f^{-1}(b) = \{a \in A | f(a) = b\}$ を満たす.
- $f(A)$ は f による A の B に対する像を表し, $f(A) = \{f(a) | a \in A\}$ を満たす.

写像 f が集合 A の要素から集合 B の部分集合への写像であるときは, これを, べき集合を用いて $f: A \rightarrow 2^B$ と表す.

次に, 多変数変数を添え字と写像により定義する族について説明する. ある集合 X に対し, b_1, b_2, \dots, b_n ($\in X$) をこの順番で要素に持つ数列を $b := (b_1, b_2, \dots, b_n)$ あるいは $b := (b_i | i = 1, 2, \dots, n)$ と表す. ここで, 添字集合 $N := \{1, 2, \dots, n\}$ を導入し, 数列 b を添字集合 N から X への写像とみなし, $b: N \rightarrow X$ と定義することもできる. 添字集合が自然数でなく, 一般の集合により与えられたものを族と呼ぶ. 以降一般の添字集合 $\Lambda := \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ について議論を行う. 族は写像であるが, 通常添字を引数にとる $b(\lambda)$ ではなく, b_λ の形で表記される. 例として, b が x - y - z 軸上に定義される 3 次元変数であるとき, $\Lambda = \{x, y, z\}$ を用いて, $b = (b_\lambda | \lambda \in \Lambda)$ または $b = (b_x, b_y, b_z)$ と表すことができる. 族により多次元変数, 系列等の順序関係を持つ変数の組を表現することができる.

各添字 λ ($\in \Lambda$) に対応する集合 X_λ があるとき, $X := (X_\lambda | \lambda \in \Lambda)$ のような集合を要素を持つ族を集合族と呼ぶ. $\forall \lambda \in \Lambda, b_\lambda \in X_\lambda$ を満たす族 b の全体からなる集合を, X の直積とよび, $\prod_{\lambda \in \Lambda} X_\lambda$ と表す. 例として, 3 次元 Euclid 空間の各軸を x, y, z とするとき, この空間に対する添字集合は $\Lambda = \{x, y, z\}$ と与えられ, 集合族は $X = (\mathbb{R}, \mathbb{R}, \mathbb{R})$ となる. このとき, X の直積は $\mathbb{R} \times \mathbb{R} \times \mathbb{R}$ であり, このような同じ集合からなる直積にたいしては, \mathbb{R}^Λ の表記が用いられる.

直積は写像の集合である. つまり, 多次元空間は写像の集合であり, 多次元空間の 1 点を表すのに一つの写像が割り当てられる. もし, 族 b の各要素が媒介変数 $t \in \mathbb{R}$ を引数に取るような

$b(t) = (b_\lambda(t) | \lambda \in \Lambda)$ である場合, b は \mathbb{R} からの写像でもあるし, Λ からの写像でもあるが, 元となる集合の中で添字集合の方は通常省略され, $b: \mathbb{R} \rightarrow \prod_{\lambda \in \Lambda} X_\lambda$ と表記される. この決まりを用いて, 先ほどの Euclid 空間中に時刻 t に依存した軌跡を描く写像 $f(t)$ は $f: \mathbb{R} \rightarrow \mathbb{R}^3$ と表記される.

本来多次元変数を定義するためには各軸を識別する添字を定義しなければならないが, 「 D 次元変数 x は \mathbb{R}^D に属する ($x \in \mathbb{R}^D$)」というような表現をした場合, 添字には $1, 2, \dots, D$ のように自然数を用いる慣習がある. しかしこの表記は用いる軸が概ね限定的である物理学の分野では誤解のないものであるが, 情報科学の分野においては軸として様々な抽象的概念が考えられるため, 添字集合を定義した表記が有効な場面も多い. 例えば 3.1 節で扱う情報推薦の分野ではユーザの集合とアイテムの集合, および対応する評価値行列が用いられるが, ユーザ集合とアイテム集合をそれぞれ添字集合とし, その上に評価値行列が定義されるとすれば整数の添字を利用する必要はない. 3.1 節ではこの表記を採用する. なお, 添字集合 Λ 上に定義された族 $a = (a_i | i \in \Lambda), b = (b_i | i \in \Lambda)$ に関し, $a = b$ であることと $\forall i \in \Lambda, a_i = b_i$ であることは同値である.

命題 P により定義される加算集合 $A = \{a | P(a)\}$ があるとき, 写像 $f: A \rightarrow \mathbb{R}$ の, A の要素全てについての和を $\sum_{P(a)} f(a)$ と表す. 多重積についても同様に $\prod_{P(a)} f(a)$ と表す. この表記により, $i = 1, 2, 4, \dots, 100$ の総和を $\sum_{i \in \mathbb{N} \wedge i \leq 100 \wedge i \neq 3} i$ などと表すことができる.

順序が必要である多項演算の表現には族を用いる. 添字集合 Λ 上に定義される 2 つの族 a, b について, $\lambda \in \Lambda$ に対応する要素同士の積の和を $\sum_{\lambda \in \Lambda} a_\lambda b_\lambda$ と表記し, 多重積についても同様に, $\prod_{\lambda \in \Lambda} a_\lambda b_\lambda$ と表記する.

文献 [6] では, ある命題 p について, $[p]$ の表記は $(p \Rightarrow [p] = 1) \wedge (\neg p \Rightarrow [p] = 0)$ を満たすものとする記法を提案している. この記法は Kronecker Delta の一般形であり, 例えば, $x \in \mathbb{R}$ について, $x = 0$ においてのみ 1, それ以外で 0 を取る関数 $f(x)$ を, $f(x) = [x = 0]$ と表現できる. $[1 > 2] = 0, [0 \in \mathbb{R}] = 1$ を満たすこともわかる. この記法は有用であり, 本論文で用いる多くの条件式を簡潔に表現することができる. 本論文ではこの記法を用いるが, 括弧だけによる表記では曖昧となる場所があるため, 文献 [27] と同様に, 命題からの写像として式 (2.1) を満たすように定義される $\delta: \{T, F\} \rightarrow \{0, 1\}$ を用いる. これにより, Kronecker Delta は $\delta_{ij} = \delta(i = j)$ と定義することができる.

$$(p \Rightarrow \delta(p) = 1) \wedge (\neg p \Rightarrow \delta(p) = 0) \quad (2.1)$$

順序を定義できる集合 A に対して, 以下の 2 つの規則を定義する.

- \max は $\max(A) = \{a \in A | \nexists b \in A, b > a\}$ を満たす, A の部分集合を返す関数とする.
- \min は $\min(A) = \{a \in A | \nexists b \in A, b < a\}$ を満たす, A の部分集合を返す関数とする.

また, A とある集合 B に対し, 写像 $f: B \rightarrow A$ があるとき, f の最小値をあたえる引数の集合 $\arg \min_{c \in C} f(c), C \subset B$ を式 (2.2) により定義する.

$$\arg \min_{c \in C} f(c) := f^{-1}(\min \{f(C)\}) \cap C \quad (2.2)$$

2.1.4 グラフ

与えられたグラフに対し, \mathcal{N} を節点の集合, \mathcal{E}_p を辺の全体集合とし, $\bigcup_{e \in \mathcal{E}_p} (n_{st}(e), n_{ed}(e)) = \mathcal{N}^2$ を満たす2つの写像 $n_{st} : \mathcal{E}_p \rightarrow \mathcal{N}$, $n_{ed} : \mathcal{E}_p \rightarrow \mathcal{N}$ があるとき, ある辺 $e \in \mathcal{E}_p$ に対して, $n_{st}(e)$ を e の始点となる節点, $n_{ed}(e)$ を e の終点となる節点と定義する. また, これらの逆写像 $n_{st}^{-1} : \mathcal{N} \rightarrow \mathcal{E}_p$, $n_{ed}^{-1} : \mathcal{N} \rightarrow \mathcal{E}_p$ が存在し, ある節点 $n \in \mathcal{N}$ に対し, $n_{st}^{-1}(n)$ は n を始点に持つ辺の集合, $n_{ed}^{-1}(n)$ は n を終点に持つ辺の集合を表す. $\mathcal{E} (\subset \mathcal{E}_p)$ は与えられたグラフに含まれる辺の集合とする.

2.1.5 確率空間と確率変数

確率空間は標本空間, 事象の集合, 確率測度からなる [15]. 標本空間を Ω としたとき, $A \subset \Omega$ である A を事象と呼ぶ. 事象の集合を $S = \{A | A \subset \Omega\}$ としたとき, 事象が発生する確率を与える関数 $P : 2^\Omega \rightarrow [0, 1]$ の中で, 式 (2.3) を満たすものを確率測度と呼び, (Ω, S, P) の3組を確率空間と呼ぶ.

$$[\forall A, B \in S, A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)] \wedge P(\Omega) = 1 \quad (2.3)$$

標本空間 Ω からある集合 Ω'_x への全射 $\mathbf{x} : \Omega \rightarrow \Omega'_x$ を確率変数と呼ぶ. 確率変数の終集合 Ω'_x を状態集合, あるいは状態空間と呼ぶ. \mathbf{x} を全射とすることで, \mathbf{x} が取る任意の値 $x \in \Omega'_x$ について元となる事象 $\mathbf{x}^{-1}(x) (\subset \Omega)$ を定めることができ, $\mathbf{x} = x$ となる確率を $P(\mathbf{x}^{-1})$ と表すことができる. また, これと等しい値を $p(\mathbf{x} = x)$ と表記し, このような p を慣習的に \mathbf{x} の従う確率分布と呼ぶ [18]. $x \in \Omega'_x$ のような確率変数の取る値は実現値と呼ばれる. 確率変数に対して, 確率空間の定義から以下の性質が導かれる.

確率空間 (Ω, S, P) において, 2つの確率変数 $\mathbf{x} : \Omega \rightarrow \Omega'_x$, $\mathbf{y} : \Omega \rightarrow \Omega'_y$ と実現値 $x \in \Omega'_x, y \in \Omega'_y$ があるとき, 同時確率 $p(\mathbf{x} = x, \mathbf{y} = y)$ は式 (2.4) と書ける.

$$p(\mathbf{x} = x, \mathbf{y} = y) = P(\mathbf{x}^{-1}(x) \cap \mathbf{y}^{-1}(y)) \quad (2.4)$$

また, 次の式 (2.5) が成り立つ.

$$\begin{aligned} \mathbf{x}^{-1}(x) &= \mathbf{x}^{-1}(x) \cap \Omega \\ &= \mathbf{x}^{-1}(x) \cap \left(\bigcup_{y \in \Omega'_y} \mathbf{y}^{-1}(y) \right) \\ &= \bigcup_{y \in \Omega'_y} (\mathbf{x}^{-1}(x) \cap \mathbf{y}^{-1}(y)) \end{aligned} \quad (2.5)$$

写像の性質から, $\forall y, y' \in \Omega'_y, y \neq y' \Rightarrow \mathbf{y}^{-1}(y) \cap \mathbf{y}^{-1}(y') = \emptyset$ であり (付録 B.4), 式 (2.3), 式 (2.4), 式 (2.5) を用いれば, 式 (2.6) を導ける. これを確率変数 \mathbf{x} の周辺分布と呼ぶ.

$$\begin{aligned} P(\mathbf{x}^{-1}(x)) &= \sum_{y \in \Omega'_y} P(\mathbf{x}^{-1}(x) \cap \mathbf{y}^{-1}(y)) \\ \Leftrightarrow p(\mathbf{x} = x) &= \sum_{y \in \Omega'_y} p(\mathbf{x} = x, \mathbf{y} = y) \end{aligned} \quad (2.6)$$

\mathbf{y} が独立に定まり、 \mathbf{x} はそれに対し従属的に定まる場合を考える。 \mathbf{y} が独立であることから、式 (2.4) は式 (2.7) と表される。

$$\begin{aligned} p(\mathbf{x} = x, \mathbf{y} = y) &= p_{\mathbf{y}}(\mathbf{x} = x)p(\mathbf{y}) \\ &= P_{\mathbf{y}}(\mathbf{x}^{-1}(x))P(\mathbf{y}^{-1}(y)) \end{aligned} \quad (2.7)$$

ここで、右辺における \mathbf{x} は元の確率空間とは異なる $(\Omega, S, P_{\mathbf{y}})$ からなる確率空間の上に定義される確率変数となる。 $p_{\mathbf{y}}$, $P_{\mathbf{y}}$ をそれぞれ、 \mathbf{y} を与えられたもとでの \mathbf{x} の条件付き分布、確率測度と呼び、確率測度は異なるが、慣習的に元の確率空間における確率測度と分布に用いたものと同じ記号を用いて式 (2.8) のように表す。条件付きの記号 $|$ があることから元の分布とは異なるものと判別できるため、この記法を用いても曖昧性は生じない。

$$\begin{aligned} p_{\mathbf{y}}(\mathbf{x} = x) &= p(\mathbf{x} = x | \mathbf{y} = y) \\ &= \frac{p(\mathbf{x} = x, \mathbf{y} = y)}{p(\mathbf{y} = y)} \\ &= P(\mathbf{x}^{-1}(x) | \mathbf{y}^{-1}(y)) \end{aligned} \quad (2.8)$$

確率変数、確率分布、状態空間がわかれば、元の確率空間の構造もわかるため、確率空間が明示的に扱われることは通常ない。また、確率変数だけを用いて、慣習的に同時分布は式 (2.9) の右辺、条件付き分布は式 (2.10) の右辺のように表記される [18][38]。

$$p(\mathbf{x} = x, \mathbf{y} = y) = p(\mathbf{x}, \mathbf{y}) \quad (2.9)$$

$$p(\mathbf{x} = x | \mathbf{y} = y) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \quad (2.10)$$

しかしながら、この表記法では確率変数に用いる記号で確率空間を識別することになり、単純なモデルに対する適用を除いて実用的ではなく、近年の確率モデルの複雑化に対応できないと考えられる。曖昧な表記法を用いたとしても人が処理する上では対処可能な場合もあるが、数式処理システム (Computer Algebra System)[74] を用いた解析においては弊害となりうる。

2.1.6 添字を用いた確率空間と確率変数の表記

複雑なモデルを構築する上で数学的表記法は重要であり、歴史的にも数多くの表記法が開発されてきた。例えば Isaac Newton は微分概念とその表記法 [5]、Albert Einstein はテンソル代数の表記を簡便にするために縮約記法 (付録 A) を導入した [75]。用いられるモデルの発展に伴い、表記法は今後も拡張、洗練されていくと考えられる。このような背景の中、2.1.5 節で説明した現状用いられている確率モデルを記述するための表記法は一般性のあるものとは言えず、実際に次に述べるような不便性、曖昧性を含んでおり、改善する必要があると考える。本節では、現在広く持ちいられている表記法の問題点を踏まえ、本論文で用いる確率変数の表記法について解説する。

2.1.6. A 既存文献における表記

既存文献の多くでは確率分布を引数に与えられる確率変数で区別する。この記法を用いるとき、標本空間 Ω と状態空間 Ω'_0 で定義される確率変数 $\mathbf{x} : \Omega \rightarrow \Omega'_0$ と $\mathbf{y} : \Omega \rightarrow \Omega'_0$ に対して式 (2.11) の

命題は一般に成り立たない.

$$p(\mathbf{x} = x) = p(\mathbf{y} = x), \forall x \in \Omega'_0 \quad (2.11)$$

式 (2.11) を成り立たせるためには \mathbf{x} と \mathbf{y} の元となる確率空間とそれぞれの写像としての性質が等しいことを明示しておく必要がある. また, 確率変数の取りうる状態についての総和を, 例えば確率変数 \mathbf{x} については $\sum_{\mathbf{x}}$ のように表すことが行われるが, これについても確率変数が異なれば総和の取り方も異なる. よって, $\mathbf{x}, \mathbf{y} \in \Omega'$ であるとき, 写像 $f: \Omega' \rightarrow \mathbb{R}$ に対して式 (2.12) は一般に成り立たない.

$$\sum_{\mathbf{x}} f(\mathbf{x}) = \sum_{\mathbf{y}} f(\mathbf{y}) \quad (2.12)$$

式 (2.11), 式 (2.12) の制約から確率変数 \mathbf{x} の分散 σ^2 は式 (2.13) のように 2 つの式に分けて書く必要がある. ここで, \mathbf{x} はスカラー変数であることを仮定している.

$$\begin{aligned} \mu &= \sum_{\mathbf{x}} \mathbf{x} p(\mathbf{x}) \\ \sigma^2 &= \sum_{\mathbf{x}} (\mathbf{x} - \mu)^2 p(\mathbf{x}) \end{aligned} \quad (2.13)$$

式 (2.13) を式 (2.14) のように別の記号 \mathbf{y} を用いて入れ子状に表すには, \mathbf{y} と \mathbf{x} は独立同分布に従うと断る必要がある.

$$\sigma^2 = \sum_{\mathbf{x}} \left(\mathbf{x} - \sum_{\mathbf{y}} \mathbf{y} p(\mathbf{y}) \right)^2 p(\mathbf{x}) \quad (2.14)$$

条件付き分布 $p(\mathbf{x}|\mathbf{y})$ を同時分布 $p(\mathbf{x}, \mathbf{y})$ から求める場合にも同様の制約から式 (2.15) のように 2 つの式に分けて書く必要がある. 条件付き分布の解析では分母分子で共通の引数を約分する操作が頻繁に行われるため, この制約による数学的解析上の不利益は大きい.

$$\begin{aligned} p(\mathbf{y}) &= \sum_{\mathbf{x}} p(\mathbf{x}, \mathbf{y}) \\ p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} \end{aligned} \quad (2.15)$$

次に確率変数が多数存在する場合における問題点について考察する. このような確率変数はまとめて多次元確率変数, その分布は多次元確率分布と呼ばれることもある. 多次元確率変数は多くの場合, 1次元のものと同様に \mathbf{x} と書かれ, この表記においてその次元数を判断することができない点がまず問題として挙げられる. 多次元確率分布を近似するために広く用いられる Gibbs Sampling [44][64] では着目する確率変数の条件付き分布が必要になり, 例えば i 番目の変数 \mathbf{x}_i の条件付き分布が式 (2.16) のように書かれることが多い [4][24]. \mathbf{x}_{-i} は \mathbf{x} から i 番目の変数を除外した確率変数を意味する.

$$\begin{aligned} p(\mathbf{x}_{-i}) &= \sum_{\mathbf{x}_i} p(\mathbf{x}) \\ p(\mathbf{x}_i|\mathbf{x}_{-i}) &= \frac{p(\mathbf{x})}{p(\mathbf{x}_{-i})} \end{aligned} \quad (2.16)$$

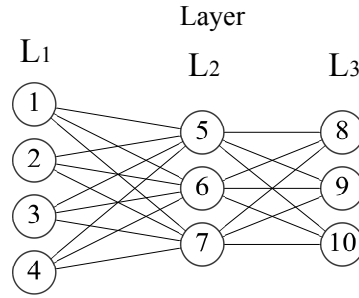


図 2.1 確率分布の説明に用いる Boltzmann Machine (BM)

この表記法は集合論で一般に用いられる表記法 [16] に従っておらず，有用な集合論の諸法則を適用することができない．例えば，集合論の記法において，直積を用いることで 2 つの多次元変数を合わせてより高次の変数を構成することができるが，この記法においてその方法は自明でない．複数の確率変数を合わせて扱うことは BM, Bayesian Network 等の階層型のモデルで行われ，これら現在有用と考えられているモデルに対する解析が容易ではなくなっている．ここで例として，図 2.1 に示す 3 層の BM に対する分布の定義を示す．文献 [38] ではこの BM を構成するユニットの状態を示す分布を式 (2.17) により表している．

$$p(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2) = \frac{1}{Z_p} \exp(-\phi(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2)) \quad (2.17)$$

ここで， $\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2$ はそれぞれ第 1 層，第 2 層，第 3 層の状態を表す多次元確率変数であり， ϕ はエネルギー関数で式 (2.18)， Z_p は分配関数で式 (2.19) により定義される． $\mathbf{W}^1, \mathbf{W}^2$ はそれぞれ 1-2 層間，2-3 層間をつなぐ辺の重み係数である．

$$\phi(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2) = -\mathbf{v}^\top \mathbf{W}^1 \mathbf{h}^1 - \mathbf{h}^{1\top} \mathbf{W}^2 \mathbf{h}^2 \quad (2.18)$$

$$Z_p = \sum_{\mathbf{v}} \sum_{\mathbf{h}^1} \sum_{\mathbf{h}^2} \exp(-\phi(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2)) \quad (2.19)$$

この式で，各層ごとに確率変数が定義されており，また前述したとおりこの変数に別の記号を用いることもできないため，条件付き分布は 1 層目から順番に式 (2.20)，式 (2.21)，式 (2.22) とそれぞれ定義される．

$$p(\mathbf{v}_i = 1 | \mathbf{h}^1) = f_\sigma \left(\sum_j \mathbf{w}_{ij}^1 \mathbf{h}_j^1 \right) \quad (2.20)$$

$$p(\mathbf{h}_j^1 = 1 | \mathbf{v}, \mathbf{h}^1) = f_\sigma \left(\sum_i \mathbf{w}_{ij}^2 \mathbf{v}_i + \sum_m \mathbf{w}_{jm}^2 \mathbf{h}_m^2 \right) \quad (2.21)$$

$$p(\mathbf{h}_m^2 = 1 | \mathbf{h}^1) = f_\sigma \left(\sum_j \mathbf{w}_{jm}^1 \mathbf{h}_j^1 \right) \quad (2.22)$$

この定式化では，式 (2.20)，式 (2.21)，式 (2.22) はほとんど同じ形をしているにもかかわらず，分けて書かれており，冗長である．

これに対し，文献 [12] による定式化では添字を利用している．層を構成する添字の集合を 1 層目から順番に $\mathcal{V}_x = \{1, 2, 3, 4\}$, $\mathcal{V}_y = \{5, 6, 7\}$, $\mathcal{V}_z = \{8, 9, 10\}$ ，確率変数を 1 層目から順番に $\mathbf{x} = \{\mathbf{x}_i | i \in \mathcal{V}_x\}$, $\mathbf{y} = \{\mathbf{x}_i | i \in \mathcal{V}_y\}$, $\mathbf{z} = \{\mathbf{x}_i | i \in \mathcal{V}_z\}$ と定義し，同時分布を式 (2.23) で表している．また層全体の集合を用いた確率変数 $\mathbf{X} = \{\mathbf{x}_i | i \in \mathcal{V}_x \cup \mathcal{V}_y \cup \mathcal{V}_z\}$ を用いて式 (2.24) とも表している． Z_p は分配関数である．

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{1}{Z_p} \exp(-\phi(\mathbf{x}, \mathbf{y}, \mathbf{z})) \quad (2.23)$$

$$p(\mathbf{X}) = p(\mathbf{x}, \mathbf{y}, \mathbf{z}) \quad (2.24)$$

式 (2.24) では集合の演算を利用した汎用性の高い表現となっている．しかしながら，文献 [12] においてもエネルギー関数は一貫して式 (2.25) のように各層の確率変数をそれぞれ用いて表記されている． θ はパラメータの集合， $b \in \mathbb{R}$, $w \in \mathbb{R}^2$ はそれぞれバイアス，結合の重み係数である．また，確率変数を記号で識別している点において他文献と異なりなく，各確率変数の条件付き分布は式 (2.15) と同様に 2 段階に分けて表記されている．

$$\phi(\mathbf{x}, \mathbf{y}, \mathbf{z}, \theta) = - \sum_{i \in \mathcal{V}_x} \mathbf{x}_i b_i - \sum_{i \in \mathcal{V}_y} \mathbf{y}_i b_i - \sum_{i \in \mathcal{V}_z} \mathbf{z}_i b_i - \sum_{i \in \mathcal{V}_x} \sum_{j \in \mathcal{V}_y} w_{ij} \mathbf{x}_i \mathbf{y}_j - \sum_{i \in \mathcal{V}_y} \sum_{j \in \mathcal{V}_z} w_{ij} \mathbf{y}_i \mathbf{z}_j \quad (2.25)$$

2.1.6. B 本論文における表記法

以上の表記上の問題に対し，本論文では確率空間と確率変数を添え字集合により識別する記法を用いる．本記法において，添字集合 Λ に対し， \mathbf{x}_Λ の表記は， \mathbf{x}_Λ が Λ の上に定義される族であるとす．さらに，族の定義を拡張し，添字集合 Λ で添字付けられた確率変数は，どのような記号を用いても同一の確率空間 $(\Omega_\Lambda, S_\Lambda, P_\Lambda)$ の標本空間 Ω_Λ から状態空間 Ω'_Λ への全射 $\mathbf{F}_\Lambda : \Omega_\Lambda \rightarrow \Omega'_\Lambda$ に等しいとする．ただし，分布に用いられる記号が異なれば確率測度だけ異なるものとする．これにより，確率変数 $\mathbf{x}_\Lambda, \mathbf{y}_\Lambda$ について，式 (2.26) の命題が成り立つ．これは式 (2.11) に対応する．ただし， $p = q$ の表記は， p, q に用いられる記号が等しいことを意味することに注意されたい．

$$(p = q) \Rightarrow p(\mathbf{x}_\Lambda = x) = q(\mathbf{y}_\Lambda = x), \forall x \in \Omega'_\Lambda \quad (2.26)$$

多次元確率変数に対して添字を用いることは一般に行われており [12][38]，各要素ではなく全体を指定するときに集合を添え字に与えたとしても表記上の負担は増えない．また，この記法の利点の一つとして，確率変数の次元は添字集合の大きさに等しいため，自明となることが挙げられる．さらに，同じ添字を持つ確率変数同士の状態空間は等しいため，式 (2.12) に対応する式 (2.27) が，本記法では成り立つ．

$$\sum_{\mathbf{x}_\Lambda} f(\mathbf{x}_\Lambda) = \sum_{\mathbf{y}_\Lambda} f(\mathbf{y}_\Lambda) \quad (2.27)$$

部分集合 $M \in \Lambda$ もまた適切な確率空間と確率変数を定義するとすれば， \mathbf{x}_M の $\mathbf{x}_{\Lambda \setminus M}$ で条件付けられた分布は式 (2.28) と表すことができ，式 (2.13) のように式を 2 つに分ける必要はない．付録 B.3 等の例に示すとおり，この表記を用いることで，条件付き分布を求めるときに，分母と分子で約分される因数を調べることが容易になる．

$$p(\mathbf{x}_M | \mathbf{x}_{\Lambda \setminus M}) = \frac{p(\mathbf{x}_\Lambda)}{\sum_{\mathbf{y}_M} p(\mathbf{y}_M, \mathbf{x}_{\Lambda \setminus M})} \quad (2.28)$$

本記法により Boltzmann 分布を表記すれば式 (2.29) と書ける.

$$p(\mathbf{x}_\Lambda) = \frac{1}{Z_p} \exp(-\phi(\mathbf{x}_\Lambda)) \quad (2.29)$$

本記法では, 添字により確率変数を定義するとしているため, パラメータの添字に対し確率変数へ用いたものと同じものは用いることができない. したがって, 本論文における BM の定式化には, 重み係数に対する添字に辺の集合 \mathcal{E} の要素, バイアスの添字に節点集合 \mathcal{N} の要素を用いることとする. 確率変数の添字と対応付けるために, 写像 $d: \Lambda \rightarrow \mathcal{N}, c: \Lambda^2 \rightarrow \mathcal{E}, \forall i, j \in \Lambda, c(i, j) = c(j, i)$ をそれぞれ定義する. 図 2.1 の例では, 各層の添字集合は $\mathcal{V}_1 := \{1, 2, 3, 4\}, \mathcal{V}_2 := \{5, 6, 7\}, \mathcal{V}_3 := \{8, 9, 10\}, \mathcal{V}_4 := \emptyset$ となり, 式 (2.25) に対応するエネルギー関数は式 (2.30) と書ける.

$$\phi(\mathbf{x}_\Lambda) = - \sum_{k=1}^3 \sum_{i \in \mathcal{V}_k} \left\{ \mathbf{x}_i b_{d(i)} + \sum_{j \in \mathcal{V}_{k+1}} \mathbf{x}_i w_{c(i,j)} \mathbf{x}_j \right\} \quad (2.30)$$

$\mathbf{x}_i = 1, i \in \mathcal{V}_k$ の $\mathbf{x}_{\Lambda \setminus \{i\}}$ で条件つけられた分布は式 (2.31) と書ける (付録 B.3).

$$p(\mathbf{x}_i = 1 | \mathbf{x}_{\Lambda \setminus \{i\}}) = f_\sigma \left(\sum_{j \in V(i)} w_{c(i,j)} \mathbf{x}_j \right) \quad (2.31)$$

ただし, $V(i)$ は i 番目の要素に結合を持つ要素の集合であり, ここでの例では $V(i) = \mathcal{V}_{k-1} \cup \mathcal{V}_{k+1}$ を満たす. 式 (2.20), 式 (2.21), 式 (2.22) による表記と異なり, 本記法において式 (2.31) はどのようなネットワーク構造を持つ BM においても同様に書ける.

2.2 標本抽出法 (Sampling Method)

標本抽出法は確率分布から標本を得る手法であり, 得られた標本を用いて各種統計量を近似することができる. 本手法にはいくつかの分類があり, 精度の高さで降順, 汎用性の高さで昇順とすれば, 累積分布関数を用いた抽出法, 近似分布を用いた棄却抽出法, Markov 連鎖を用いた抽出法がある [24]. Markov Chain Monte Carlo (MCMC) 法 [7] は定常な分布が系列から生成されるとモデル化し, 反復により元の定常分布を近似する手法である.

定常な分布 $p(\mathbf{x}_\Lambda)$ があるとする. Λ は確率空間と確率変数を定義する添字集合である. これを系列分布で近似するため, 系列の適当な長さ $N (\in \mathbb{N})$ と系列に対する添字集合 $\Lambda(1), \Lambda(2), \dots, \Lambda(N)$ を用意し, 式 (2.32) で与えられる分布を考える. ただし, すべての確率変数 $\mathbf{x}_{\Lambda(1)}, \mathbf{x}_{\Lambda(2)}, \dots, \mathbf{x}_{\Lambda(N)}$ の状態空間については, $\forall i \in \{1, 2, \dots, N\}, \Omega'_{\Lambda(i)} = \Omega'_\Lambda$ であるとし, 系列全体への添字を $R = \bigcup_{i=1}^N \Lambda(i)$ とする.

$$p(\mathbf{x}_R) = p(\mathbf{x}_{\Lambda(1)}) \prod_{i=2}^N p(\mathbf{x}_{\Lambda(i)} | \mathbf{x}_{\Lambda(i-1)}) \quad (2.32)$$

$l, k \in \{1, 2, \dots, N\}$ に対して, $R(l, k) = \bigcup_{l > i \geq k} \Lambda(i)$ としたとき, 式 (2.32) から得られる条件付き分布 $p(\mathbf{x}_{\Lambda(n)} | \mathbf{x}_{R(n,1)})$ は式 (2.33) を満たす.

$$p(\mathbf{x}_{\Lambda(n)} | \mathbf{x}_{R(n,1)}) = p(\mathbf{x}_{\Lambda(n)} | \mathbf{x}_{\Lambda(n-1)}), n = 2, 3, \dots, N \quad (2.33)$$

このようなある要素が一つ前の要素にのみ依存する系列分布を 1 次 Markov 連鎖と呼ぶ. $p(\mathbf{x}_{\Lambda(n)}|\mathbf{x}_{\Lambda(n-1)})$ のことを遷移確率分布とも呼ぶ. この Markov 連鎖から得られるそれぞれの確率変数が元の分布 $p(\mathbf{x}_{\Lambda})$ に従うように遷移確率分布を選ぶ必要がある. ここで, 式 (2.34) が成り立つ場合を考える.

$$p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) p(\mathbf{y}_{\Lambda}) = p(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}) p(\mathbf{x}_{\Lambda}), \forall \mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda} \quad (2.34)$$

このとき, 周辺分布 $p(\mathbf{x}_{\Lambda(n)})$ は式 (2.34) の結果から元の分布 $p(\mathbf{x}_{\Lambda})$ に従うことがわかる.

$$\begin{aligned} p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}) &= \sum_{\mathbf{y}_{\Lambda}} p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) p(\mathbf{y}_{\Lambda}) \\ &= p(\mathbf{x}_{\Lambda}) \sum_{\mathbf{y}_{\Lambda}} p(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}) \\ &= p(\mathbf{x}_{\Lambda}) \end{aligned} \quad (2.35)$$

式 (2.34) は詳細釣り合い条件と呼ばれる. この条件を満たすことが, 遷移確率分布が元の分布を近似するための必要条件であり, その中でも近似の効率が良いものを選ばれる. 式 (2.34) に加え, 遷移確率分布が式 (2.36) で示される全ての状態への変更可能性 (全探索性) を満たすとき, 系列は元の分布を近似する.

$$\forall x, y \in \Omega'_{\Lambda}, \exists j, p(\mathbf{x}_{\Lambda(j)} = y|\mathbf{x}_{\Lambda(1)} = x) \neq 0 \quad (2.36)$$

系列が元の分布を近似する性質は, Ergodic 性と呼ばれる [34]. 文献 [7] は式 (2.37) により詳細釣り合い条件を満たす遷移確率分布を提案している.

$$p(\mathbf{x}_{\Lambda(n)}|\mathbf{x}_{\Lambda(n-1)}) = \frac{1}{Z_p(\mathbf{x}_{\Lambda(n-1)})} A_M(\mathbf{x}_{\Lambda(n)}, \mathbf{x}_{\Lambda(n-1)}) q(\mathbf{x}_{\Lambda(n)}|\mathbf{x}_{\Lambda(n-1)}) \quad (2.37)$$

A_M は Metropolis 基準と呼ばれており [24], 式 (2.38) で定義される.

$$A_M(\mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda}) = \min \left\{ 1, \frac{p(\mathbf{x}_{\Lambda})}{p(\mathbf{y}_{\Lambda})} \right\} \quad (2.38)$$

$Z_p : \Omega'_{\Lambda} \rightarrow \mathbb{R}$ は正規化関数であり, ここでは式 (2.39) により与えられる.

$$Z_p(\mathbf{y}_{\Lambda}) = \sum_{\mathbf{x}_{\Lambda}} A_M(\mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda}) q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) \quad (2.39)$$

$q(\mathbf{x}_{\Lambda(n)}|\mathbf{x}_{\Lambda(n-1)})$ は提案分布であり, 式 (2.40) に示す対称性の条件と, 全探索性を満たすものとする.

$$q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) = q(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda}|\mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}), \forall \mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda} \quad (2.40)$$

式 (2.38), 式 (2.40) により定まる式 (2.37) は詳細釣り合い条件を満たす (付録 B.1). Metropolis の基準 A_M と対称な提案分布により任意の分布を近似することができる. A_M により標本が棄却される可能性があることで, どのような提案分布を用いたとしても詳細釣り合い条件が成り立つことが保証され, 式 (2.35) で示した周辺分布は元の分布に従う. しかしながら, 実計算においては式 (2.35) を近似的に計算できる資源が限られているため, 棄却率が高くなる提案分布を用いることはできず, 元の分布に適したものをを用いる必要がある.

対称でない提案分布を用いた場合でも詳細釣り合い条件を満たすものとして、式 (2.41) により A_M を与え、遷移確率分布を定義する Metropolis Hastings 法が提案されている [73]。ただし、 $\forall \mathbf{x}_{\Lambda(n)}, \mathbf{x}_{\Lambda(n-1)}, q(\mathbf{x}_{\Lambda(n)}|\mathbf{x}_{\Lambda(n-1)}) \neq 0$ であるとする。

$$A_M(\mathbf{x}_\Lambda, \mathbf{y}_\Lambda) = \min \left\{ 1, \frac{p(\mathbf{x}_\Lambda) q(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_\Lambda | \mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_\Lambda)}{p(\mathbf{y}_\Lambda) q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_\Lambda | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_\Lambda)} \right\} \quad (2.41)$$

Gibbs Sampling は式 (2.41) の A_M と式 (2.42) により定義される提案分布により遷移確率分布を定義する [24]。

$$q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_\Lambda | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_\Lambda) = \sum_{\mathbf{G}_g} \delta(\mathbf{x}_{\Lambda \setminus \mathbf{G}_g} = \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{x}_{\mathbf{G}_g} | \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{G}_g) \quad (2.42)$$

ここで、 \mathbf{G}_g は添字 g の上に定義される Λ の部分集合を要素にもつ確率変数であり、その状態集合は $\bigcup_{G \in \Omega'_g} G = \Lambda$ を満たす。

Gibbs Sampling において、 A_M を以下のように表し変形することが可能であり、棄却率は 0 になることがわかる。

$$A_M(\mathbf{x}_\Lambda, \mathbf{y}_\Lambda) = \min \left\{ 1, \frac{p(\mathbf{x}_\Lambda) \sum_{\mathbf{G}_g} \delta(\mathbf{y}_{\Lambda \setminus \mathbf{G}_g} = \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{y}_{\mathbf{G}_g} | \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{G}_g)}{p(\mathbf{y}_\Lambda) \sum_{\mathbf{G}_g} \delta(\mathbf{x}_{\Lambda \setminus \mathbf{G}_g} = \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{x}_{\mathbf{G}_g} | \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{G}_g)} \right\}$$

ここで、 $p(\mathbf{z}_\Lambda) = p(\mathbf{z}_{\mathbf{G}_g} | \mathbf{z}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{z}_{\Lambda \setminus \mathbf{G}_g})$ の関係があり、また、任意の集合 A と関数 $f: A \rightarrow \mathbb{R}$ に対して、 $\forall x, y \in A, f(x) \delta(x = y) = f(y) \delta(y = x)$ が成り立つため、上式は式 (2.43) と書ける。

$$\begin{aligned} A_M(\mathbf{x}_\Lambda, \mathbf{y}_\Lambda) &= \min \left\{ 1, \frac{\sum_{\mathbf{G}_g} p(\mathbf{x}_{\mathbf{G}_g} | \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) \delta(\mathbf{y}_{\Lambda \setminus \mathbf{G}_g} = \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{y}_{\mathbf{G}_g} | \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{G}_g)}{\sum_{\mathbf{G}_g} p(\mathbf{y}_{\mathbf{G}_g} | \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) \delta(\mathbf{x}_{\Lambda \setminus \mathbf{G}_g} = \mathbf{y}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{x}_{\mathbf{G}_g} | \mathbf{x}_{\Lambda \setminus \mathbf{G}_g}) p(\mathbf{G}_g)} \right\} \\ &= 1 \end{aligned} \quad (2.43)$$

この結果から、Gibbs Sampling における遷移確率分布は式 (2.42) の q に等しく、また、 q は詳細釣り合い条件を満たすことがわかる。状態空間 Ω'_g の選択には任意性があるが、標本抽出を容易にするために通常は独立集合系が選ばれる。独立集合系のうち最も単純なものに $\Omega'_g = \{\{i\} | i \in \Lambda\}$ とする方法があり、Boltzmann Machine (BM) の分布を近似する手法としてよく用いられる [37]。Restricted Boltzmann Machine 等のモデルには Ω'_g の要素として大きな独立集合を与えることができ、並列化効率を高めた実装が可能である [38][57]。

Gibbs Sampling は棄却率が 0 となるため、標本化の効率は高く、提案分布を構築することが難しい場合にも用いることができる。しかしながら、 Ω'_g および $p(\mathbf{G}_g)$ の設定によっては、得られた提案分布が同じ状態を保つ確率が高いものとなる場合に、棄却率が 0 となる利点が失われる場面もあり、特に、4 章で扱う観光経路推薦問題において $\Omega'_g, p(\mathbf{G}_g)$ の設定が標本抽出効率に大きな影響を及ぼす。

2.3 Boltzmann 分布

2.3.1 定義

Boltzmann 分布は分子のミクロな状態を記述するために導入された。粒子を識別するための添字集合を Λ とし、状態を表す確率変数を \mathbf{x}_Λ 、系の温度を T 、Boltzmann 定数を k として、Boltzmann

分布は式 (2.44) により定義される.

$$p(\mathbf{x}_A) = \frac{1}{Z_p} \exp\left(-\frac{1}{T_k} \phi(\mathbf{x}_A)\right) \quad (2.44)$$

$\phi(\mathbf{x}_A)$ はエネルギー関数であり, 添字集合 A を位置に対するもの M と運動量に対するもの $A \setminus M$ に分割するならば, 系に存在する保存力 $E(\mathbf{x}_M)$ と粒子の運動エネルギー $K(\mathbf{x}_{A \setminus M})$ により, 式 (2.45) と表すことができる.

$$\phi(\mathbf{x}_A) = E(\mathbf{x}_M) + K(\mathbf{x}_{A \setminus M}) \quad (2.45)$$

最適化問題や生成モデルへの応用においては, 位置変数のみを考慮する. しかしながら, 状態変数が実数値を取る確率変数である場合には, 運動量変数を考慮することで高速な標本抽出法 (Hybrid Monte Carlo 法) を構築できることが示されている [23].

2.3.2 焼きなまし法

式 (2.44) の Boltzmann 分布はエネルギー関数 $\phi(\mathbf{x}_A)$ の値が小さいほど, その状態を取る確率が高くなる. 温度が小さくなるほどこの傾向は強くなり, $\phi(\mathbf{x}_A)$ の最低値を与える \mathbf{x}_A の集合を Ω_A^* としたとき, 温度 $T \rightarrow 0$ の極限で式 (2.44) は式 (2.46) に収束する.

$$p(\mathbf{x}_A) = \frac{1}{|\Omega_A^*|} \delta(\mathbf{x}_A \in \Omega_A^*) \quad (2.46)$$

式 (2.46) を用いて最適解を抽出することはできないため, MCMC 法を用いながら温度 T を徐々に下げ, この極限分布を近似する. 温度を時刻 $t \in \mathbb{R}$ に対する関数 $T(t)$ とし, 式 (2.47) に従い温度を低下させていけば最適解が得られることが示されている [42]. T_H は初期温度である.

$$T(t) = \frac{T_H}{\ln(t+e)}, t \in [0, \infty) \quad (2.47)$$

しかしながら式 (2.47) を用いた場合, 温度 T が 0 へ近づくまでにあまりにも多くの標本抽出が必要となり, 現実的ではない.

線形に温度を減少させる式 (2.47) はたびたび用いられるが [45], この温度関数では終了時間が定まっており, 温度設定が最適なものからずれていて, 解の収束性が十分でなくなるような設定の場合に, 追加で計算を続ける等の処理ができないという欠点がある.

$$T(t) = (T_L - T_H)t + T_H, t \in \left[0, \frac{T_H}{T_H - T_L}\right) \quad (2.48)$$

これに対し, 式 (2.47) は終了予定時刻を過ぎたあとも計算を継続することができ, 式 (2.48) よりも多くの設定で良い解を与える [41][42] ことから広く用いられている. 本論文における提案手法では式 (2.49) を基本とした温度関数を用いる. この式において $T(1) = T_L$ となるように定式化されており, $t = 0$ を計算開始時刻, $t = 1$ をおおよその計算終了時刻として用いることが可能である.

$$T(t) = T_H \exp\left(-t \ln\left(\frac{T_H}{T_L}\right)\right), t \in [0, \infty) \quad (2.49)$$

上記以外にも、異なる温度を持つ状態を多数用意し、それらの間に相互作用を定義して焼きなまし法を構築する手法が提案されており、高い最適化性能を実現している [45]。また、同様に多数の状態を用いる手法として量子焼きなまし法が提案されており [43]、近年注目を集めている。

$\phi(\mathbf{x}_A)$ を式 (2.50) のように設定することで、任意の分布 $q(\mathbf{x}_A)$ の最大確率を与える状態を得るために SA を適用することができる。

$$\phi(\mathbf{x}_A) = -\ln q(\mathbf{x}_A) \quad (2.50)$$

2.3.3 焼きなまし法における発見的手法の定式化

最適化問題とはある制約のもと与えられた目的関数を最小化する問題であり、関連する手法として数理計画法、線形計画法がある。目的関数が連続変数からなる場合に使える手法として、Newton Raphson 法は古くから知られその性能も高いが、極値や変曲点が多い関数においては局所最適解へ収束しやすい問題点がある [17]。また、Newton Raphson 法では目的関数の微分値が求められる必要があり、状態変数が離散的であるものには対応できない。焼きなまし法は離散変数からなる問題に対しても適用可能な手法である。

変数が属する状態空間を S_p とし、目的関数 $\phi: S \rightarrow \mathbb{R}$ を最小化することを考える。ここで、説明変数の取りうる状態のうち、問題の制約を満たす状態空間を $S (\in S_p)$ とする。ある状態 $a (\in S)$ から別の状態への変化を与える操作を $f: S \rightarrow 2^S$ とする。 $f(a) \subset S$ であり、この集合から選ばれた $b (\in f(a))$ もまた S に属する。山登り法では $B \left(= \arg \min_{x \in f(a)} \phi(x) \right)$ の中から、適当に $b (\in B)$ を選び $a = b$ として状態を更新する。事前に定めた回数この手続きを行うか、 $B = \emptyset$ となった場合に b を解候補とする。

これに対し、焼きなまし法による実装では、上述した f による状態変化を遷移確率分布により表し、状態空間を探索する。文献 [41] で提案された手法に対応する提案分布は、 f を用いて式 (2.51) で与えられる。

$$q(\mathbf{x}_{A(n)} = \mathbf{x}_A | \mathbf{x}_{A(n-1)} = \mathbf{y}_A) = \frac{1}{|f(\mathbf{y}_A)|} \sum_{x \in f(\mathbf{y}_A)} \delta(x = \mathbf{x}_A) \quad (2.51)$$

$\forall \mathbf{x}_A, \mathbf{y}_A \in S, \mathbf{x}_A \in f(\mathbf{y}_A) \Leftrightarrow \mathbf{y}_A \in f(\mathbf{x}_A)$ ならば式 (2.51) で与えられる提案分布は対称となり、これに対し Metropolis 基準を適用した遷移確率分布は式 (2.35) の詳細釣り合い条件を満たす。提案分布は全探索性を満たす必要があるため、状態変化が方向付けられることが一般的な、Tree Search [32][35] にこの定式化をそのまま適用することはできない。 p には 2.3.2 節で説明したとおり、目的関数をエネルギー関数とした Boltzmann 分布を用いる。式 (2.51) の提案分布は目的関数を増加させる方向にも状態を変更させるものであり、温度 T が大きければ、棄却率は 0 へ近づき、どの状態変化の発生頻度も互いに近くなる。これにより山登り法で問題となる、局所最適解への収束を高い温度において防ぐことができる。

式 (2.51) の定式化では、探索範囲が山登り法に用いられていたものと同様であり、焼きなまし法に適したものとなっていない。これに対し、4.1 節では適切な提案分布を設計することで、焼きなまし法を効率よく動作させる手法を提案する。

2.3.4 Boltzmann Machine

本来の Boltzmann 分布は物質を構成する粒子の状態を記述したものであり、そのエネルギー関数は物理的背景により定められている。これに対し、Boltzmann Machine (BM) では、式 (2.52) の形に制限されたエネルギー関数の中から、与えられた分布に合うものを学習により求めて用いる。

$$\phi(\mathbf{x}_A) = - \sum_{(i,j) \in \mathcal{E}} \mathbf{x}_i w_{c(i,j)} \mathbf{x}_j - \sum_{i \in A} \mathbf{x}_i b_{d(i)} \quad (2.52)$$

パラメータ $w_{c(i,j)}$ を学習するとき、温度 T と Boltzmann 定数 k は $w_{c(i,j)} = Tkw'_{c(i,j)}$ などと定めることから学習中に $w_{c(i,j)}$ へ吸収されてしまうため、これらを 1 とした式 (2.53) の Boltzmann 分布を通常用いる。

$$p(\mathbf{x}_A) = \frac{1}{Z_p} \exp(-\phi(\mathbf{x}_A)) \quad (2.53)$$

観測可能な要素の集合 \mathbf{x}_M に対する学習データの経験分布 $q(\mathbf{x}_M)$ を $p(\mathbf{x}_A)$ で近似することを考える。近似度合いを図る尺度として、相対情報量 [8] と Expectation-Maximization (EM) 法 [22] における完全データ対数尤度関数を用いるものがあり、それぞれ式 (2.54)、式 (2.55) と定義される。EM 法では完全データの経験分布を $q(\mathbf{x}_A) = q(\mathbf{x}_M)p(\mathbf{x}_{A \setminus M})$ と近似する。 $l(\theta)$ は相対情報量であり、 θ はパラメータの集合である。

$$l(\theta) = - \sum_{\mathbf{x}_M} q(\mathbf{x}_M) \ln \left(\sum_{\mathbf{x}_{A \setminus M}} p(\mathbf{x}_A) \right) \quad (2.54)$$

$$l(\theta) = - \sum_{\mathbf{x}_A} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) \ln p(\mathbf{x}_A) \quad (2.55)$$

BM の学習には勾配降下法が広く用いられ、 l の微分値は式 (2.56) により与えられる [2][37][38](付録 B.2)。この結果は EM 法、相対情報量の微分どちらを用いても同じである。

$$\frac{\partial l}{\partial u} = \sum_{\mathbf{x}_A} \left\{ \frac{\partial \phi(\mathbf{x}_A)}{\partial u} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M}) - \frac{\partial \phi(\mathbf{x}_A)}{\partial u} p(\mathbf{x}_A) \right\} \quad (2.56)$$

式 (2.56) において $u = w_{c(i,j)}$ とすれば重み係数についての勾配、 $u = b_{d(i)}$ とすればバイアスについての勾配が得られる。式 (2.57) は $u = w_{c(i,j)}$ を代入した結果である。

$$\frac{\partial l}{\partial w_{c(i,j)}} = - \sum_{\mathbf{x}_A} \{ \mathbf{x}_i \mathbf{x}_j q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M}) - \mathbf{x}_i \mathbf{x}_j p(\mathbf{x}_A) \} \quad (2.57)$$

この勾配を用いてパラメータを式 (2.58) に従い更新する。ここで、 t は学習の進捗度であり、 $\epsilon(t) \Delta t$ は学習係数であり、0.001 程度の値が用いられる [13]。

$$u(t + \Delta t) = u(t) + \epsilon(t) \Delta t \left\{ - \frac{\partial l}{\partial u} \right\} \quad (2.58)$$

勾配降下法による学習方法の妥当性については依然として検証が十分ではないが、多くの文献で実用的な学習が行えるものとして用いられている [13]。

パラメータが多い場合に過学習 [13] の発生を抑制するために、式 (2.58) の学習則に罰則項を与えた式 (2.59) が広く持ちいられている。 α_{dec} は減衰係数であり、0.001 程度の値がよく用いられる [60].

$$u(t + \Delta t) = u(t) + \epsilon(t) \Delta t \left\{ -\frac{\partial l}{\partial u} - \alpha_{\text{dec}} u(t) \right\} \quad (2.59)$$

また、パラメータ空間における探索効率をよくすることを目的に慣性項を付加した式 (2.60) の学習則もよく用いられる。 α_{mom} は慣性項係数であり、0.9 程度の値とすることが多い [13].

$$u(t + \Delta t) = u(t) + \epsilon(t) \Delta t \left\{ -\frac{\partial l}{\partial u} - \alpha_{\text{dec}} u(t) \right\} + \alpha_{\text{mom}} \{u(t) - u(t - \Delta t)\} \quad (2.60)$$

2.4 評価実験で用いる計算環境

擬似乱数の生成には Mersenne Twister [54] を用いている。Mersenne Twister の初期化には python version 3.5.2 (2018 年現在)*¹ から利用できるハードウェア乱数を用い、Recall[27] 等は Leave One Out 法 [18] による平均値を表示している。計算速度の実行時間 [s] による評価は、プログラムを C++ 14*² により書き、Ubuntu 16.04.5*³ 上において、GNU Compiler tool set, g++ version 5.4.0*⁴ により最適化オプション -O3 を付加してコンパイルし、同 OS 上で実行した結果に基づく。個々の実験はすべて単一スレッドにより実行されている。CPU には Ryzen Thread Ripper 1920X*⁵ を用いている。

計算の精度に関わる事柄について、3 章の実験には単精度浮動小数点数、4 章の実験には倍精度浮動小数点数を用いている [3].

*¹ <https://www.python.org/>

*² <https://isocpp.org/>

*³ <https://www.ubuntu.com/>

*⁴ <https://www.gnu.org/software/gcc/>

*⁵ <https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-1920x>

表 3.1 3章で用いる記号の表

記号	定義	意味
U		ユーザの集合
P		アイテムの集合
r	$\in \mathbb{R}^{U \times P}$	評価値行列
\hat{r}	$\in \mathbb{R}^{U \times P}$	予測評価値
P_{sub}	$U \rightarrow 2^P$	ユーザが評価したアイテム集合
c_{sim}	$U^2 \rightarrow \mathbb{R}$	ユーザ間の相関係数
D_{latent}	$\in \mathbb{N}$	潜在因子の次元数
N_{base}	$\in \mathbb{N}$	基底ベクトルの数, 式 (3.18)
$\Omega_M^{(\text{data})}$	$\in 2^{\Omega'_M}$	訓練データ集合
$\Omega_M^{(t)}$	$t \in \mathbb{R} \mapsto x \subset \Omega'_M$	時刻 t における Batch 集合
α_{spa}	$\in [0, \infty)$	スパース正則化係数
K	$\in \mathbb{N}$	BM の層数
T_k	$\in (0, \infty)$	BM の第 $k (\in \{1, 2, \dots, K\})$ 層の温度
\mathcal{V}_k	$\subset A$	BM の第 $k (\in \{1, 2, \dots, K\})$ 層の変数に対する添字集合
ϕ_k	$\Omega'_A \rightarrow \mathbb{R}$	BM の第 $k (\in \{1, 2, \dots, K\})$ 層のエネルギー関数
$\phi_{i,k}$	$\Omega'_A \rightarrow \mathbb{R}$	BM の第 $k (\in \{1, 2, \dots, K\})$ 層のユニットから添字 $i (\in A)$ で識別されるユニットへの情報の流れ

第 3 章

Boltzmann Machine に対する焼きなまし法の適用

3.1 Boltzmann Machine を用いた潜在因子推定

3.1.1 はじめに

本節では Restricted Boltzmann Machine (RBM) [60] を用いて評価値行列から抽出された潜在因子を調査した結果に基づき、事後分布推定における Boltzmann Machine (BM) の特性について考察する。ユーザ (利用者: user) を行に、アイテム (商品: item) を列とし、各要素がユーザのアイテムに対する評価値を表しているものを評価値行列と呼ぶ。この評価値行列にはユーザが評価していないアイテムについて未知の要素があり、すでに得られている評価値行列中の値を用いて未知の要素の値を推測する技術は総称して協調フィルタリング (CF: Collaborative Filtering) と呼ばれる。CF はアイテムの内容に関する情報を必要とせずとも比較的高い評価値の推定精度を持つことから、多くの推薦システムで用いられている。協調フィルタリングを実現する手法のうち、もっとも古くに提案され、現在でもベースラインや各種拡張手法のベースとして用いられている手法に k-最近傍法がある [58]。しかしながら、この手法では近傍ユーザとの相関係数を用いた予測評価値の計算量が多いことが問題点として挙げられている。近年、この評価値行列を低次元の行列の積に分解する Matrix Factorization (行列分解法) [59] [65] が注目を集めている。行列分解法では分解することに時間はかかるが、ユーザ毎に推薦を行う段階では低次元のベクトル同士の内積を計算すれば良いので、実際の運用に向いている。また行列分解法は高い推定精度を持ち、内容ベース (content-base) 推薦との相性も良いため、多くの派生手法が研究されている [61]。

行列分解法に属する手法には Singular Value Decomposition (SVD), Probabilistic Matrix Factorization (PMF) [62] 等の他、RBM を用いた手法も提案されている [57]。RBM はグラフィカルモデル (Graphical Model) [21] としては SVD, PMF と同様潜在因子と観測因子からなる 2 階層の構造を持つが、SVD, PMF が潜在因子から観測因子への有効グラフ (Graph) であるのに対し、RBM は無向グラフである。文献 [57] では RBM の可視層を Soft-max 関数とし、この関数が出力するクラス (Class) が評価値に対応するとしている。この RBM をユーザの数だけ用意し、アイテムの潜在因子は RBM の隠れ層と可視層をつなぐ辺の重みであるとし、それらをすべてのユーザ間

で共有することで協調フィルタリングの効果を期待している．評価実験においてこの手法は SVD を用いた推薦手法を多少上回る結果を示したが，その差は小さい．また，その結果に対する数学的な議論は与えられていない．

本調査ではまず，RBM を用いた行列分解法の別の定式化を提案する．提案する定式化ではシステム全体に一つの RBM を割り当てる．また，先行研究 [57] ではアイテムの潜在因子が重みに割り当てられ，他の Neural Network から得られる特徴量は通常重みではなくニューロン（グラフにおける節点）の状態として出力されるため，内容ベース推薦手法との互換性が悪いと考える．そこで，本節で提案する定式化では，アイテムの潜在因子を RBM の隠れ層に割り当て，ユーザの潜在因子は辺の重みに割り当てる．さらに，可視層に Bernoulli Unit を用い，評価値はその状態の期待値として表現されるとモデル化する．本定式化において抽出されたアイテムの潜在因子は，人工データを作成した潜在因子をある条件において完全に復元することを評価実験により示す．また，出力される期待値の確率的挙動についても議論し，事後分布推定における BM の特性について議論する．

本節で用いる主要な記号を表 3.1 に記載する．

3.1.2 協調フィルタリング

本節において， U をユーザの集合， P をアイテムの集合， $r := (r_{up} \in \mathbb{R} | u \in U, p \in P)$ を評価値行列とする．また，ユーザ u ($u \in U$) が評価したアイテム集合を $P_{\text{sub}}(u)$ ($\subset P$) と表す．

前述の通り，協調フィルタリングの目的は，評価値行列から未評価項目の評価値を推測することである．ユーザ u のアイテム p に対する評価値を推定するとき， k -最近傍法ではまず， p に対する評価値を持ち， u と嗜好の近い k 人のユーザ t を相関係数 $c_{\text{sim}}(u, t)$ の高い順から選ぶ．相関係数として，ここではよく用いられる式 (3.1) の Pearson 相関係数により $c_{\text{sim}}(u, t)$ を与える例を示す．

$$c_{\text{sim}}(u, t) = \frac{\sum_{l \in P_{\text{sub}}(u) \cap P_{\text{sub}}(t)} (r_{ul} - \bar{r}'_u) (r_{tl} - \bar{r}'_t)}{\epsilon + \sqrt{\sum_{l \in P_{\text{sub}}(u) \cap P_{\text{sub}}(t)} (r_{ul} - \bar{r}'_u)^2} \sqrt{\sum_{l \in P_{\text{sub}}(u) \cap P_{\text{sub}}(t)} (r_{tl} - \bar{r}'_t)^2}} \quad (3.1)$$

$$\bar{r}'_u = \frac{1}{|P_{\text{sub}}(u) \cap P_{\text{sub}}(t)|} \sum_{l \in P_{\text{sub}}(u) \cap P_{\text{sub}}(t)} r_{ul} \quad (3.2)$$

$$\bar{r}'_t = \frac{1}{|P_{\text{sub}}(u) \cap P_{\text{sub}}(t)|} \sum_{l \in P_{\text{sub}}(u) \cap P_{\text{sub}}(t)} r_{tl} \quad (3.3)$$

ϵ はゼロ除算を防ぐための微小な定数である．

ここで，ユーザ u に対して，相関係数 c_{sim} の値が降順となるよう並べられたユーザの列 t_u を式 (3.4) で定義する．

$$t_u := (t_{ui} \in U | i = 1, 2, \dots, |U|, \forall j = \{i + 1, i + 2, \dots, |U|\}, c_{\text{sim}}(u, t_{ui}) \geq c_{\text{sim}}(u, t_{uj})) \quad (3.4)$$

このとき，予測評価値 \hat{r}_{up} は式 (3.5) により算出される．

$$\hat{r}_{up} = \bar{r}_u + \frac{\sum_{j=1}^k c_{\text{sim}}(u, t_{uj}) (r_{t_{uj}p} - \bar{r}_{t_{uj}})}{\sum_{j=1}^k c_{\text{sim}}(u, t_{uj})} \quad (3.5)$$

$$\bar{r}_x = \frac{1}{|P_{\text{sub}}(x)|} \sum_{l \in P_{\text{sub}}(x)} r_{xl}, x \in U \quad (3.6)$$

同様の問題に対し, Singular Value Decomposition (SVD) は式 (3.7) で与えられる目的関数 f を最小化する \hat{r}_{up} を予測評価値として用いる. $h := (h_{pk} \in \mathbb{R} | p \in P, k = 1, 2, \dots, D_{\text{latent}})$ はアイテムの潜在因子, $w := (w_{uk} \in \mathbb{R} | u \in U, k = 1, 2, \dots, D_{\text{latent}})$ はユーザの潜在因子である.

$$f = \frac{1}{2} \sum_{u \in U} \sum_{p \in P} \left\{ \delta(p \in P_{\text{sub}}(u)) (\hat{r}_{up} - r_{up})^2 + \alpha_{\text{dec}} \sum_{k=1}^{D_{\text{latent}}} (h_{pk}^2 + w_{uk}^2) \right\} \quad (3.7)$$

ここで, δ は一般化 Kronecker Delta であり, \hat{r}_{up} は式 (3.8) により定義される.

$$\hat{r}_{up} = \sum_{k=1}^{D_{\text{latent}}} h_{pk} w_{uk} \quad (3.8)$$

Probabilistic Matrix Factorization (PMF) は最大事後確率推定 (Maximizing Posterior Probability Estimation) [18] により潜在因子を求める. 最大事後確率を求めるために, 文献 [62] では事後確率の対数を目的関数としたものを最大化する手法を用いている. $\mathbf{r} := (\mathbf{r}_{up} | u \in U, p \in P)$, $\mathbf{h} := (\mathbf{h}_{pk} | p \in P, k = 1, 2, \dots, D_{\text{latent}})$, $\mathbf{w} := (\mathbf{w}_{uk} | u \in U, k = 1, 2, \dots, D_{\text{latent}})$ をそれぞれ評価値行列, アイテムの隠れ因子, ユーザの隠れ因子の確率変数, $\hat{\mathbf{r}}_{up}$ を \hat{r}_{up} の確率変数とする. \mathbf{r}_{up} , \mathbf{h}_{pk} , \mathbf{w}_{uk} がそれぞれ平均 $\hat{\mathbf{r}}_{up}$, 分散 σ^2 , 平均 0, 分散 σ_H , 平均 0, 分散 σ_W の正規分布に従うとし, 文献 [62] では目的関数を式 (3.9) により与えている.

$$\ln p(\mathbf{h}, \mathbf{w} | \mathbf{r}) = -\frac{1}{2} \sum_{u \in U} \sum_{p \in P} \left\{ \delta(p \in P_{\text{sub}}(u)) (\hat{\mathbf{r}}_{up} - \mathbf{r}_{up})^2 + \sum_{k=1}^{D_{\text{latent}}} \left(\frac{\sigma^2}{\sigma_H^2} \mathbf{h}_{pk}^2 + \frac{\sigma^2}{\sigma_W^2} \mathbf{w}_{uk}^2 \right) \right\} \quad (3.9)$$

評価値行列の事後分布は式 (3.10) により与えられる.

$$p(\mathbf{r}_{up} | \mathbf{h}, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{r}_{up} - \hat{\mathbf{r}}_{up})^2}{2\sigma^2}\right) \quad (3.10)$$

PMF を拡張し, MCMC 法を用いて最大事後確率推定を行うことで, より高い精度の推薦を可能とする手法も提案されている [63].

文献 [57] ではユーザごとに RBM を割り当て, RBM 間で重みを共有することにより協調フィルタリングを実現する手法が提案されている. 評価値は RBM の可視層に割り当てられた Soft max ユニットによる離散値として表現され, 5 段階の評価値にそれぞれ対応した 5 つのユニットを持つ可視層が用いられている. この定式化では隠れ層の状態はユーザの潜在因子を表し, 辺の重みがアイテムの潜在因子を表現している. 図 3.1(1) に文献 [57] による評価値行列に対する RBM の適用法を示す. この図において評価値行列の欠損値 (ユーザ (行) が評価していないアイテム (列) となる項) を灰色で表現してある.

3.1.3 本調査における定式化

本調査ではアイテムの潜在因子に隠れユニットを割り当てる. 図 3.1(2) に提案する RBM の適用法を示す. 多数の RBM を用意する代わりに, 本調査では評価値を与えるアイテムを識別する,

$p_1, p_2, \dots, p_{|P|}$ と記された識別子ユニットを隠れ層に接続している．各アイテムに対応したあるアイテムについての評価値を考えると、そのアイテムに対応する識別子ユニットのみが 1 の状態を取り、それ以外はすべて 0 の状態を取る．この定式化では、識別子ユニットに繋がれた重みを通じて、隠れ層に現れるアイテムの潜在因子の状態が定められる．文献 [57] の定式化では、辺の重みに対してアイテムの潜在因子が割り当てられるため、Neural Networks 等他のグラフィカルモデルとの複合モデルを構築する際に、辺に対して外部モデルのユニットからの特徴量入力が行われるという一貫性のない定式化となる．これに対し、本定式化は他のグラフィカルモデルと組み合わせる定式化を、グラフの構造を単純に拡張することで行えるという利点を持つ．図 3.2 に複合モデルの構成例を示す．

文献 [57] では評価値の表現を分類問題として扱っていたが、そのような定式化では数値間にある相関を捉えることができないため、本調査では評価値を Bernoulli ユニット (状態は $\{0, 1\}$) の取る期待値として表現する．これにより、数値間に相関を持つ連続値を再現するように定式化される SVD と妥当な比較が可能となる．

RBM の状態を示す確率変数を添字集合 Λ を用いて表す． $\Lambda = M \cup \{1, 2, \dots, D_{\text{latent}}\}$ 、可視ユニット (ユーザの評価値を表現) の添字集合を M ($|M| = |U|$)、隠れユニットの添字集合を $\Lambda \setminus M$ とし、アイテム p ($\in P$) に対する経験分布 $q(\mathbf{x}_M)$ を式 (3.11) で与える．ただし、 $r_{up} \in (0, 1)$ であり、 $m : M \rightarrow U$ は添字集合 M からユーザ集合 U への全単射である．

$$q(\mathbf{x}_M) = \prod_{i \in M} r_{m(i)p}^{x_i} (1 - r_{m(i)p})^{1-x_i} \quad (3.11)$$

RBM の学習には、2.3.4 節で説明した勾配降下法に基づき、パラメータ $w_{c(i,k)}$, $i \in M, k \in \Lambda \setminus M$ の更新式を、式 (2.58) を拡張した式 (3.12) により与える． $l : \theta \rightarrow \mathbb{R}$ は尤度関数であり、 θ はパラメータの集合である．

$$w_{c(i,k)}(t + \Delta t) = w_{c(i,k)}(t) + \Delta t \epsilon(t) \left(\frac{\partial l(\theta)}{\partial w_{c(i,k)}} + f_s - \alpha_{\text{dec}} w_{c(i,k)}(t) \right) + \alpha_{\text{mom}} (w_{c(i,k)}(t) - w_{c(i,k)}(t - \Delta t)) \quad (3.12)$$

ここで、 α_{mom} は慣性項係数である．また、 f_s はスパース正則化項であり [60]、式 (3.13) により定義される．

$$f_s = \alpha_{\text{spa}} \{ \rho (1 - \mathbb{E}[\mathbf{x}_k]) - (1 - \rho) \mathbb{E}[\mathbf{x}_k] \} \mathbb{E}[\mathbf{x}_u] \quad (3.13)$$

スパース正則化は、目標活性化率 ρ と k ($\in \Lambda \setminus M$) 番目隠れユニットの平均活性化率 $\mathbb{E}[\mathbf{x}_k]$ の間の Kullback-Leibler 情報量を対数尤度関数に罰則として加算し、活性化率 (期待値) を ρ に近づけるものである．この正則化により、隠れ層の取る状態を制御することができる． α_{spa} はスパース正則化係数であり、この係数に小さな値を設定することで、隠れ層が疎な状態を取るように学習を行うことができる．

$\mathbb{E}[\mathbf{x}_k]$ は、Pulse Density Modulation (PDM) [72] により近似することができる [13]．学習時刻 t における期待値を $\mathbb{E}[\mathbf{x}_k](t)$ とし、全時刻における期待値 $\mathbb{E}[\mathbf{x}_k] = \int_0^1 \mathbb{E}[\mathbf{x}_k](t) dt$ は式 (3.14) により与えられる $I_k(t)$ を用いて近似する． τ は時定数であり、 $\mathbb{E}[\mathbf{x}_k]$ の近似に主として影響を及ぼす

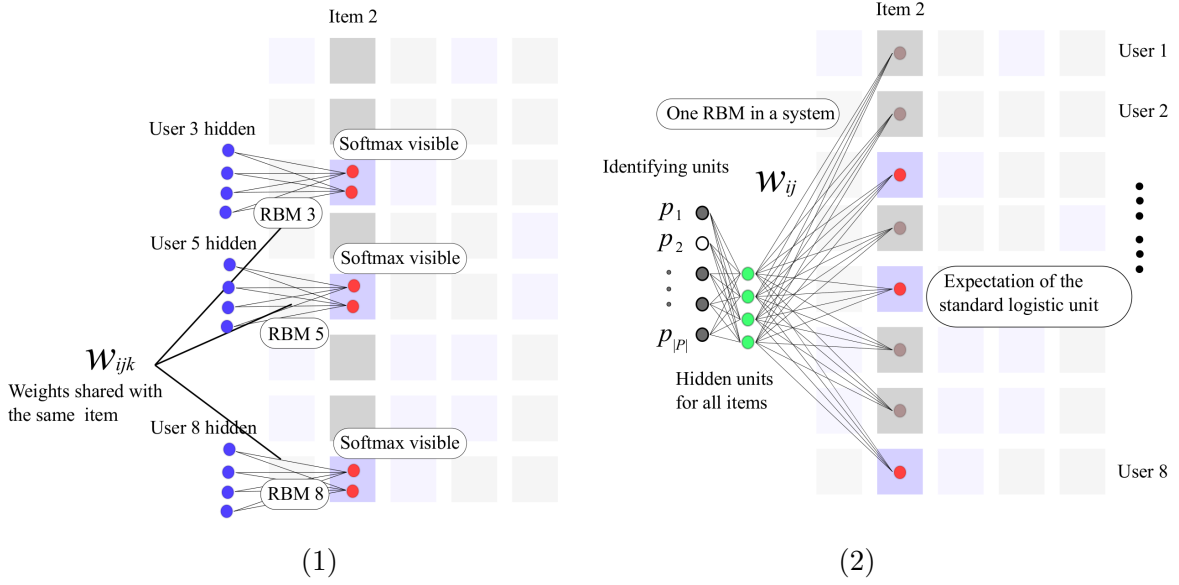


図 3.1 評価値行列に対する RBM の適用法. (1):先行研究 [57], (2):本調査における定式化

$\mathbb{E}[\mathbf{x}_k](t)$ が含まれる t の区間は、この値に比例した長さとなる.

$$\tau \frac{dI_k(t)}{dt} = -I_k(t) + \mathbb{E}[\mathbf{x}_k](t) \quad (3.14)$$

式 (3.14) を Euler 陽解法による式 (3.15) により離散的に解く.

$$\tau \frac{I_k(t + \Delta t) - I_k(t)}{\Delta t} = -I_k(t) + \mathbb{E}[\mathbf{x}_k](t) \quad (3.15)$$

図 3.1 では、各列がひとつの訓練データに対応していることを濃い色で表現している。欠損している項によるノイズが学習に影響を与えることを防ぐため、未観測データに対応するユニットの状態は 0 と設定し、学習を行う。定式化は異なるが、このように未評価値の影響を取り除く仕組みは、文献 [57] による手法でも用いられている。

本定式化において、識別子ユニットと隠れ層の結合は双方向ではなく、隠れ層の状態が識別子ユニットの状態に影響を与えることはない。同一の列、つまり同一のアイテムについてのデータを学習している間は、識別子ユニットの状態が変わることはない。

3.1.4 評価実験

評価実験において、比較手法として SVD を用いる。また、本調査で用いる RBM の出力値の範囲は $[0, 1]$ であるため、SVD により計算される予測評価値の範囲もこれと合わせる必要がある。このため、元の定式化で用いられている評価値に対し Sigmoid 関数を適用し、式 (3.16) により値域を $(0, 1)$ とした SVD を比較手法として用いる。なお、RBM のユニットの期待値がとる値域は $[0, 1]$ であるが、この違いは Root Mean Squared Error (RMSE) により性能を評価する本実験において問題とならない。

$$\hat{r}_{up} = f_{\sigma} \left(\sum_{k=1}^{D_{\text{latent}}} w_{uk} h_{pk} \right), u \in U, p \in P \quad (3.16)$$

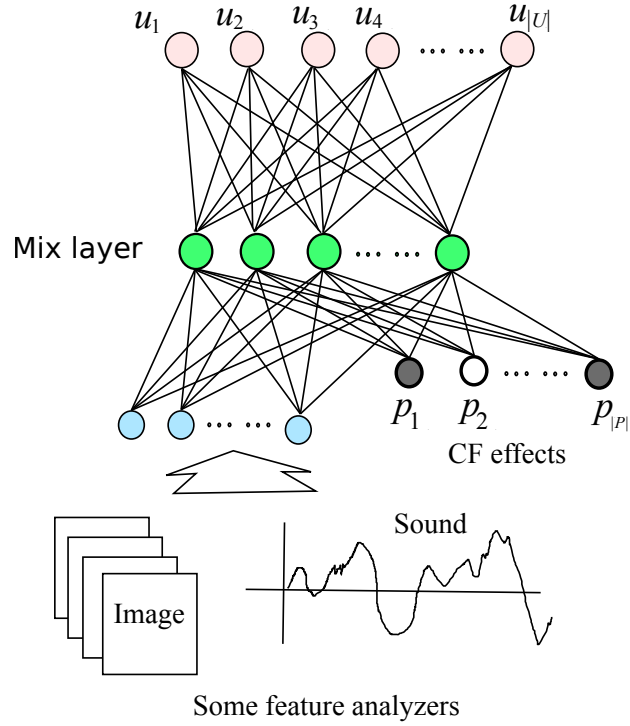


図 3.2 本調査における定式化と他の Neural Networks による複合モデル. $p_1, p_2, \dots, p_{|P|}$ と記されたユニットはアイテムに対応する識別子ユニットを表す.

評価実験では, 人工的に設定した潜在因子から生成した評価値行列を用いる. この行列の要素のうちいくつかを欠損値として設定し, その値を復元する能力を前述の通り RMSE により評価する. 評価実験において, アイテムの潜在因子に対する設定方法が異なる 2 つのデータセットを用いる. どちらの場合においてもユーザの潜在因子は同様に $[0, 1]$ の範囲の一樣乱数で与える. 得られたユーザ, アイテムの潜在因子から, 式 (3.17) を用いて評価値行列 $r \in (0, 1)^{U \times P}$ を生成する.

$$r_{up} = f_{\sigma} \left(\sum_{k=1}^{D_{\text{latent}}} w_{uk} h_{pk} \right), u \in U, p \in P \quad (3.17)$$

3.1.4. A Basic Vectors Case の設定

このデータセットでは, アイテムの潜在因子 h_{pk} を式 (3.18) により与える.

$$h_{pk} = \frac{1}{N_{\text{base}}} \sum_{l=1}^{N_{\text{base}}} \delta(l = \mathbf{x}_{pm}) \quad (3.18)$$

$N_{\text{base}} (\in \mathbb{N})$ は基底ベクトルの数で, \mathbf{x}_{pm} は式 (3.19) で与えられる確率分布に従う確率変数である.

$$p(\mathbf{x}_{pm}) = \sum_{k=1}^{N_{\text{base}}} \frac{1}{D_{\text{latent}}} \delta(k = \mathbf{x}_{pm}) \quad (3.19)$$

3.1.4. B Random Vectors Case の設定

このデータセットは, h_{pk} を区間 $[0, 1]$ における一様乱数により決定する. この条件は, Basic vector case において $N_{\text{base}} \rightarrow \infty$ とした場合に対応する.

3.1.4. C 評価実験における共通の設定

すべての条件において, $\epsilon(t), t \in \mathbb{R}$ は式 (3.20) により決定する. この式により決定される $\epsilon(t)$ は, $\forall t \in \mathbb{R}, \epsilon(t+1) = \frac{1}{A}\epsilon(t)$ となる性質を持つ.

$$\frac{d\epsilon(t)}{dt} = -\epsilon(t) \ln(A) \quad (3.20)$$

評価実験において, $A = 10$, $\Delta t \epsilon(0) = 0.01$, $\Delta t = 2 \times 10^{-8}$ と設定した. スパース正則化は隠れ層にだけ適用し, $\tau = 0.1$, $\alpha_{\text{spa}} = 0.01$, $\rho = 0.2$ の設定値を用いた. また慣性項の係数には $\alpha_{\text{mom}} = 0.9$ を用いた. 期待値を近似するための Persistent CD 法における標本抽出回数は, 文献 [57] が示した実験結果でパラメータ更新ごとに 1 回でも十分であるとされているため, 本調査でも更新毎に 1 回とした. ユーザ数 $|U| = 100$, アイテム数 $|P| = 100$, データセットの生成に用いる潜在因子の次元数は $D_{\text{latent}} = 5$ とした. 同じ列を連続で 30 回適用し, 次の列に移ることを繰り返し学習を行った.

3.1.4. D Basic Vectors Case の実験結果

本調査で用いる RBM の定式化と, SVD による, Basic Vectors Case における実験結果を示す.

本調査では $N_{\text{base}} = 1$ の場合のみ扱う. 隠れ層の次元は $D_{\text{latent}} = 10$ とした. 図 3.3 に示すのは, 様々な行列密度のデータセットに対する予測評価値の RMSE の比較である. 行列の密度が 20%, 90% の場合において SVD が RBM よりも低い RMSE を達成しており, 10% と 30% の場合においては RBM のほうが低い RMSE を達成していることがわかる.

図 3.4 に示すのは, 行列密度 10% のデータセットから推定されたアイテムの潜在因子の可視化結果である. 図の左に, 元となるデータセット, 右に潜在因子の可視化を, 評価値の値が高いほど白くなるよう濃淡をつけて示した. 右に示される可視化結果はそれぞれ, データセット作成に用いた潜在因子 (original), RBM により復元されたもの, SVD により復元されたものに対応する. SVD による復元結果は, 元の潜在因子とは大きく異なる一方で, RBM のものはほとんど元の潜在因子に等しいことがわかる.

図 3.5 は行列密度 90% の場合における結果であり, 図の見方については図 3.4 と同様である. SVD により得られた結果に Gram-Schmidt の直交化法を用いることで元の基底ベクトルの復元を試みた結果を SVD + O と示す. この場合において, RBM は元の潜在因子を完全に復元している. SVD による結果は, SVD + O の場合であっても元の状態を復元することは困難であることがわかる.

3.1.4. E Random Vectors Case の実験結果

Random Vectors Case における結果を示す. 本節の実験において隠れ層の次元は $D_{\text{latent}} = 20$ とした. 図 3.6 は図 3.3 に対応する, 行列密度が 10%, 20%, 90% の場合における予測評価値の

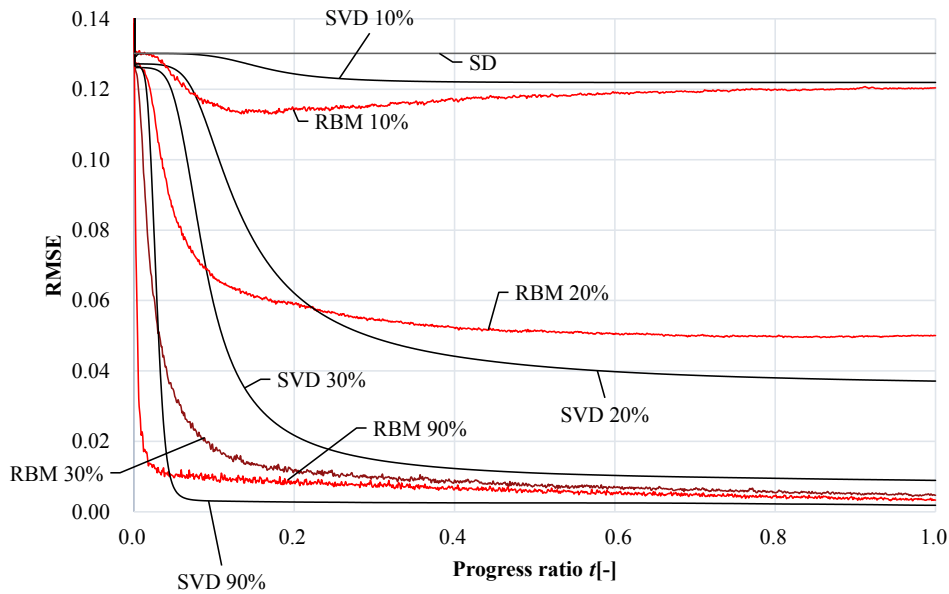


図 3.3 $N_{\text{base}} = 1$ とした Basic Vectors Case から生成されたデータセットを用いた実験における, SVD と RBM の RMSE の比較. ラベル中の各数値は評価値行列の密度を表している. 横軸は $t/\Delta t$ を示す. SD (Standard Deviation) はユーザごとに評価値の平均値を取り, そのユーザの予測評価値とした結果.

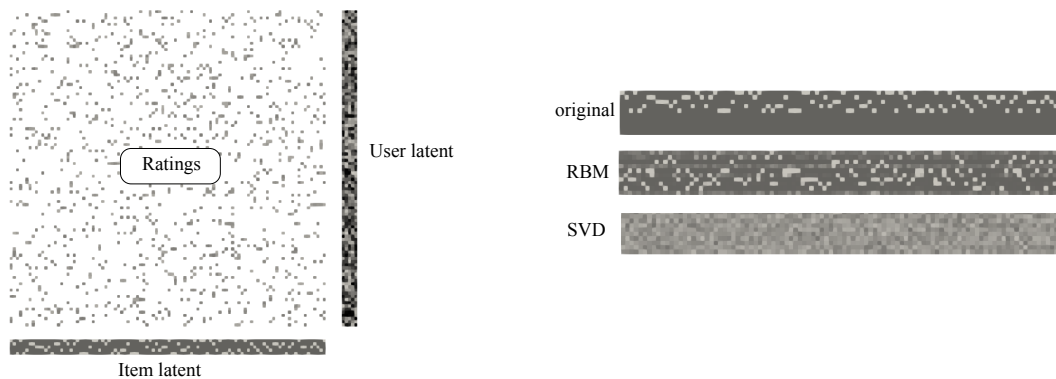


図 3.4 Basic Vectors Case を用いた, 行列密度 10% の場合における結果の可視化. 左: データセットを構築する評価値行列と潜在因子. 右: それぞれの手法, RBM, SVD により復元されたアイテムの潜在因子. original は左側に表示されているアイテムの潜在因子と同じもので, これを拡大したものである.

RMSE の比較である. SVD が RBM に対しいずれの場合も優れていることがわかる.

潜在因子の状態の可視化を図 3.7 に示す. RBM は 図 3.4 の場合と比べより多数の隠れユニットを潜在因子の表現に用いていることがわかる. Random Vectors Case は $N_{\text{base}} \rightarrow \infty$ とした Basic Vectors Case のデータセットに等しいので, これは妥当な結果である. Random vectors case のような連続値からなる潜在因子に対応する場合は, バイアスのみが異なるユニットを無限に複製することで連続値の表現を可能とした Rectified Linear Units [67] 等の利用が考えられる.

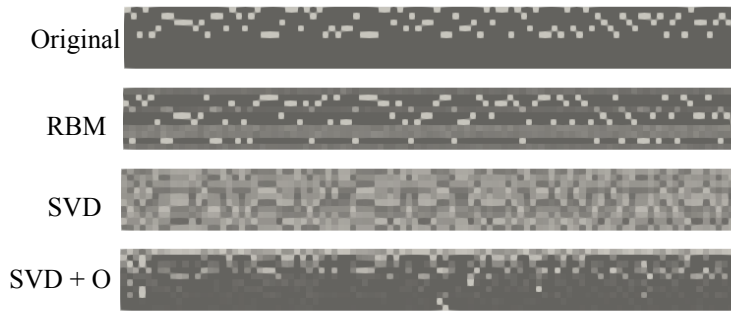


図 3.5 Basic Vectors Case を用いた，行列密度 90% の場合における潜在因子の可視化．SVD + O は SVD の結果に Gram-Schmidt の直交化法を適用したもの

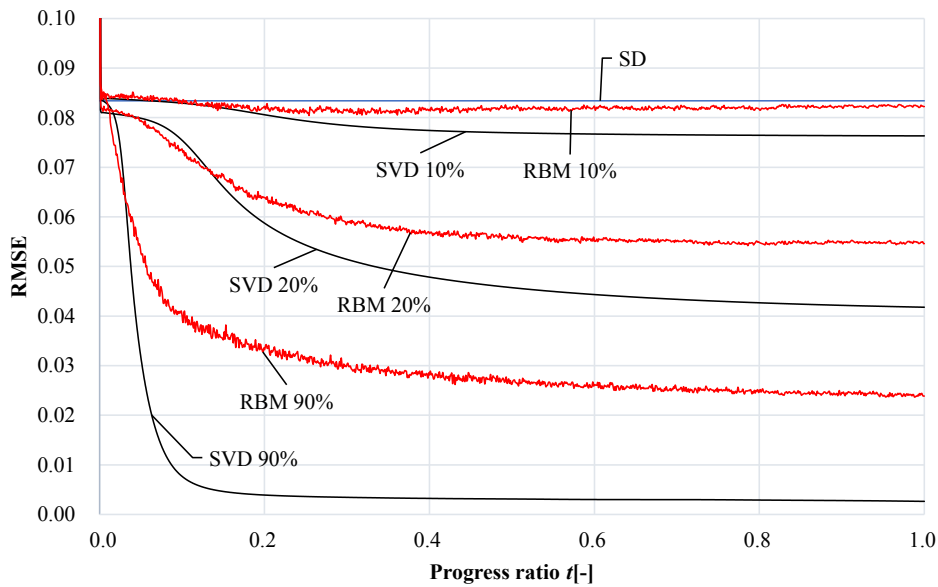


図 3.6 Random Vectors Case から生成されたデータセットを用いた実験における，SVD と RBM の RMSE の比較．それぞれの数値は評価値行列の密度を表している．横軸は $t/\Delta t$ を示す．SD (Standard Deviation) はユーザごとに評価値の平均値を取り，そのユーザの予測評価値とした結果．

3.1.5 まとめ

本節では Restricted Boltzmann Machine を用いて評価値行列から抽出される潜在因子の特性を調査した．また，先行研究 [57] よりも拡張性の高い評価値行列に対する RBM 適用法を提案した．Basic Vectors Case におけるデータセットを用いた評価実験から，RBM は評価値行列が少数の基底ベクトルを持つ潜在因子から構成されている場合に，SVD を上回る精度を示す可能性があることを示した．実世界のデータセットは Basic Vectors Case のような潜在因子が存在すると考えられるため，RBM が有効に機能する場面があることが期待できる．Random Vectors Case の結果も合わせた多くの場合で，RBM は SVD よりも高い RMSE となったが，得られた潜在因子を可視化した

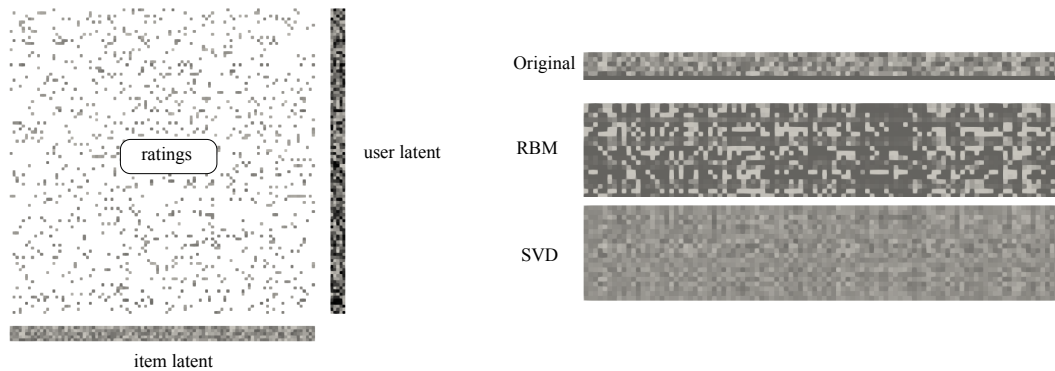


図 3.7 Random Vectors Case を用いた，行列密度 90% の場合における結果の可視化．左：データセットを構築する評価値行列とアイテムの潜在因子．右：それぞれの手法，RBM，SVD により復元されたアイテムの潜在因子．original は左側に表示されている潜在因子と同じもので，これを拡大したものである．

結果，RBM は Basic Vectors Case において，データセットの生成に用いたアイテムの潜在因子を完全に復元した．このような性質を発見できたことは本調査の最も大きな貢献であると言える．また，調査を通して，学習曲線等のグラフにも見られる通り，RBM により生成される予測評価値は確率的にばらつきを持つことが判明した．これがノイズとなり，RMSE における指標では性能が悪く見積もられている可能性があると考ええる．特に，0 か 1 の値しか取らない変数を用いて中間の値を期待値で近似する場合，分布の分散は大きくなることから，連続値を表現可能な分布を用いる必要性があり，その場合においても分布の期待値を高精度に見積もる手法が必要であると言える．精度向上のために複雑なモデルを用いる場合には，学習，推論の手法として簡素で汎用性の高いものが望まれることから，焼きなまし法を適用可能な方法を研究する価値は高いと考える．

3.2 焼きなまし法の結晶化効果

3.2.1 はじめに

本節では温度分布を用いた焼きなまし法により BM の事後分布を高精度に推定する手法を提案する。

近年、今までは難しいと考えられていた多くの問題において、Neural Networks を利用した手法が成功を取めている。たとえば、Convolutional Neural Networks (CNN) [47][48][49] は画像認識において高い分類精度、特徴抽出能力を持ち、Recurrent Neural Networks [50] は音声データや自然言語等の系列データの解析に広く用いられている。

BM [37][38] もまた Neural Networks の一つとみなされているが、他の Neural Networks と違い、本モデルの状態変数同士は双方向性の結合を持つ。この特徴により BM はマルチモーダルデータに対するモデルとして適していると期待されている [52]。また BM は高い潜在的な特徴抽出能力を持つことから、教師あり学習 [39][40]、教師なし学習 [55][68] どちらの手法としても広く用いられている。Boltzmann 分布と確率論により、他の Neural Networks と比較して緻密に定義されており、数学的解析が行い易いことから BM をベースとした拡張手法の定式化もしやすいという特徴がある。

BM の持つ欠点の一つとして、分布を推定するのに計算負荷の高い標本抽出法 (Sampling Method) を必要とする点が挙げられる。学習時には十分な計算時間を用意することができるため、BM は事前学習手法として広く用いられるが、推論時には高精度で高速な手法が必要となり、確率的ばらつきによる不安定性や、標本抽出法の計算量の多さが問題となるため、推論モデルとしては通常用いられない [38]。このことは、3.1 節の調査においても、BM による推論の結果得られた精度が確率的ばらつきを持つことを示した通りである。BM のような確率的生成モデルを推論モデルとして用いる場合、入力を与えた状態における事後分布を求める問題に帰着される。事後分布の近似推定法として変分推論 (Variational Inference) [38][23] があるが、この手法もまた近似された状態を得るために反復法を必要とし、さらに、Deep Boltzmann Machine (DBM) のような階層モデルに適用するためには、反復回数を削減するために、各層ごとに分けて入力側から順番に初期化する方法が用いられる [38]。反復法の利用を避けるために、通常 BM は推論時において順伝搬型 Neural Networks へ変換され用いられる。しかしながら、このとき BM の持つ双方向性結合の性質は失われてしまい、さらに、学習時と推論時において別の手法を必要とすることは定式化だけでなく実装も複雑にする原因となるため、Field-Programmable Gate Array (FPGA) や Application Specific Integrated Circuit (ASIC) 等 [76] による実装を困難とする。

焼きなまし法は Boltzmann 分布を用いて定式化されていることから、BM に対しても適用可能であるが、著者の調査した限りこの課題に対して十分な研究は行われていない。本節の評価実験においても示すが、焼きなまし法を一般的な方法により BM に適用しただけでは推論の精度を改善できないばかりか、従来から用いられている標本抽出手法である Gibbs sampling [44] に劣るため、これまで利用されてこなかった可能性がある。

BM への焼きなまし法の適用に関するこれらの問題を解決するために、本節では BM 本来の形式

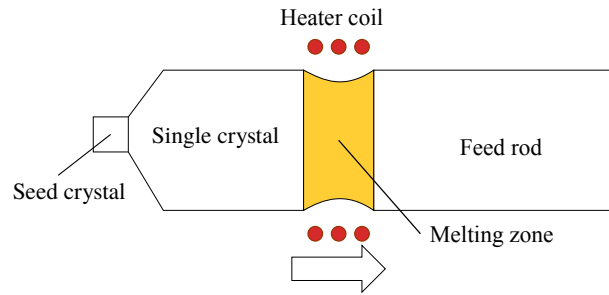


図 3.8 FZ 法の概念図. 融解領域を種結晶 (Seed crystal) 側から原料 (Feed rod) 側へ移動させていくことで単結晶を得ることができる.

を変えないことなく、事後分布を高精度、安定的に推定する手法を提案する。提案手法の特徴は、状態変数間で一様でない温度分布を用いて焼きなまし法を適用する点にある。従来から BM の分布推定に使われている Gibbs Sampling 法と、状態変数間で一定の温度を用いる従来の焼きなまし法との比較実験を人工データセットと MNIST [47] を用いて行った結果に基づき、提案手法の有効性を示す。また、従来の焼きなまし法が推論精度の向上に対し効果がなく、一様でない温度分布を用いることの有効性も様々な温度分布を用いた実験により示す。さらに、数学的解析に基づき、非一様な温度分布を BM に与えた場合における、状態変数間の情報の流れについても議論する。

3.2.2 焼きなまし法と Floating Zone 法

Floating Zone 法 [46][71] は材料科学における、シリコンの単結晶を作成する手法の一つである。図 3.7 に FZ 法の概念図を示す。焼きなまし材料の不均一性や欠陥、歪を取り除く事を主な目的としてその性質を変化させるため、系全体で一様な温度を用いるのに対し [70], FZ 法では単結晶を得るために、融解した高温部を種結晶側から原料側へ移動させる。温度が一様な分布を保ち十分な時間をかけて低下した場合には、物質を構成する分子及び原子は近くの結晶構造を延長させるように並ぶが、近くに結晶化したものがない、あるいは温度低下が十分に緩やかでない場合には、独立した結晶が成長を始める。焼きなましでは内部で常に独立した結晶が成長しやすい状況にあるため、単結晶を得るには非現実的に長い時間を要する。FZ 法では連続的に種結晶側から温度を低下させていくことで、常に延長すべき結晶構造が近くにある状態を保つため単結晶を得ることができる。図 3.9 に FZ 法が適用されているシリコン内の粒子の挙動の概念図を示す。図中において、左側の単結晶側から温度を低下させることで、種結晶の結晶構造を延長するように粒子が並ぶ様子が示されている。また、図 3.10 には矩形容器を仮定した場合の焼きなましと FZ 法における結晶粒成長の様子の概略を示す。焼きなましでは、急速に冷却した場合に比べれば大きな結晶を持つ材料を生成することはできるが、それら結晶粒が障害となり、種結晶の結晶構造が反対側に反映されない。

3.2.3 提案手法

BM は学習後、入力のクラスに対応したいくつかの平衡分布を持つ。それらはたとえば数字の“1”や“2”，犬や猫に対応していると推測される。本節ではそれらクラスはそれぞれの分布から定まる結晶構造を持つと考え、FZ 法の効果を狙った焼きなまし法を BM へ適用することを提案する。

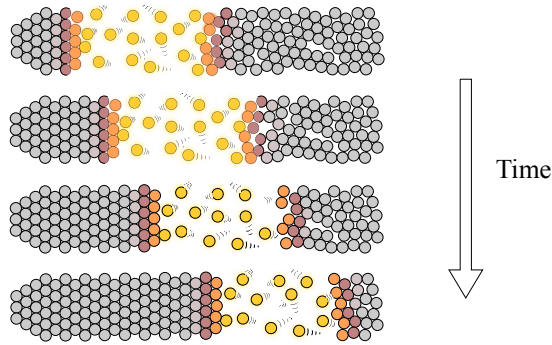


図 3.9 FZ 法における粒子の運動

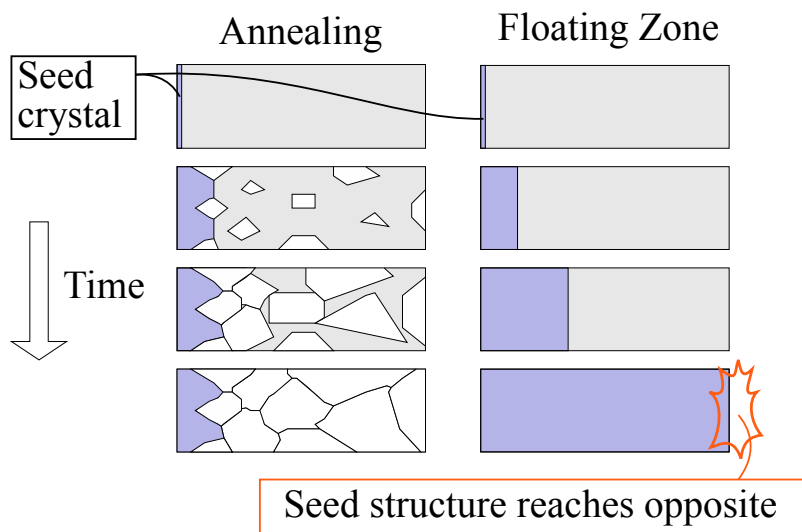


図 3.10 焼きなまし (Annealing) と FZ 法における結晶粒成長の様子

結晶構造は入力の状態 (種結晶) から決定されると考え、入力側から温度を低下させていくことで入力の分布に対応した単結晶の構造を持つ分布が得られることを期待する。3.2.2 節で述べたとおり、温度を材料全体で一様に低下させればたくさんの異なる結晶が内部で成長し、全体として単結晶を得られにくくなるため、組み合わせ爆発の問題から焼きなましを用いて単結晶を得ることは現実的に困難である。また、ある場所の結晶構造は遠くの結晶構造へ影響を及ぼしにくいため、系の規模が大きくなるにつれて最適化の困難性は増す。DBM 等の、階層構造を持ち、変数間の結合の距離に差があるモデルへ焼きなまし法を適用した場合にも同様の議論を当てはめることができ、中間層において入力の分布とは独立な分布 (結晶) が成長し、出力側に入力の情報が伝わらなくなることが想定される。

一様温度分布を用いる従来の焼きなまし法の代わりに、提案手法では式 (3.21) により BM の分布を定義することで、一様でない温度分布を用いた焼きなまし法を適用可能とする。

$$p(\mathbf{x}_\Lambda) = \frac{1}{Z_p} \exp \left(- \sum_{k=1}^K \frac{1}{T_k} \phi_k(\mathbf{x}_\Lambda) \right) \quad (3.21)$$

K は層数であり, $\phi_k(\mathbf{x}_A)$ は第 k 層のエネルギー関数であり, 式 (3.22) により定義する.

$$\phi_k(\mathbf{x}_A) = -\frac{1}{2} \sum_{i \in \mathcal{V}_k} \sum_{j \in \mathcal{V}_{k-1} \cup \mathcal{V}_{k+1}} \mathbf{x}_i w_{c(i,j)} \mathbf{x}_j - \sum_{i \in \mathcal{V}_k} \mathbf{x}_i b_{d(i)} \quad (3.22)$$

ここで, \mathcal{V}_k は全体の添字集合 A に対し, $A = \cup_{k=1}^K \mathcal{V}_k \wedge \mathcal{V}_0 = \emptyset \wedge \mathcal{V}_K = \emptyset$ を満たすものとする. なお, 本節において一様でない温度分布は推論時にのみ用いるとし, 学習時にはすべての温度は一様に 1 とする.

本定式化における \mathbf{x}_i の条件付き分布は式 (3.23) で与えられる. ただし, $i \in \mathcal{V}_k$ であるとする.

$$p(\mathbf{x}_i = 1 | \mathbf{x}_{A \setminus \{i\}}) = f_\sigma \left(\frac{1}{T_k} b_{d(i)} + \frac{1}{T_{k-\frac{1}{2}}} \phi_{i,k-1}(\mathbf{x}_A) + \frac{1}{T_{k+\frac{1}{2}}} \phi_{i,k+1}(\mathbf{x}_A) \right) \quad (3.23)$$

$f_\sigma(x) = 1/(1 + \exp(-x))$ は Sigmoid 関数である. $i \in \mathcal{V}_k$ であるとき, $\phi_{i,k-1}(\mathbf{x}_A)$ は第 $k-1$ 層, $\phi_{i,k+1}(\mathbf{x}_A)$ は第 $k+1$ 層から第 k 層に位置する i で識別されるユニットへの情報の流れをそれぞれ表し, 式 (3.24) により定義される.

$$\phi_{i,k}(\mathbf{x}_A) = \sum_{j \in \mathcal{V}_k} \mathbf{x}_j w_{c(i,j)} \quad (3.24)$$

$T_{k-\frac{1}{2}}$ は式 (3.25) により定義される. ただし, $T_0 = 1, T_{K+1} = 1$ であるとする.

$$T_{k-\frac{1}{2}} = \frac{2T_k T_{k-1}}{T_k + T_{k-1}} \quad (3.25)$$

式 (3.23) において, T_{k-1} が T_k, T_{k+1} にくらべ十分に小さい時, $1/T_{k-\frac{1}{2}}$ の値がほかと比べて大きくなるため, $\phi_{i,k-1}(\mathbf{x}_A)$ が支配的になる. このことは, 情報の流れが温度の低い側から高い側へ制限されることを意味する. 結果として, 情報の流れを温度分布により操作することが可能になることがわかる.

推論時, 入力を与えられた場合の出力の事後分布を求めなければならないが, 階層型モデルにおいて出力層と入力層の間には中間層があるため, 本節冒頭で述べたとおり, 中間層に独立な分布 (結晶) が生成されやすいと考える. その結果, 出力と入力の情報伝搬が妨げられることが, 事後分布を不安定にする原因の一つと考える. FZ 法はこのような現象を防ぐために用いることができ, 式 (3.23) に対する議論から, FZ 法は確率論の視点からも情報の流れを制御する効果をもつため, Boltzmann 分布を用いたどのクラスの分布に対しても適用可能と考える. もし温度を入力側から下げていけば, 情報の流れは入力から出力側へ方向づけられることとなり, この考えを再帰的に用いることで, 出力の分布を入力の状態から決定可能と考える. 結果として出力層の分布の不安定な挙動を取り除くことができ, 高精度な推論が可能と考える.

入力側から低下していく温度分布として, 本節では, 簡易的に式 (3.26) を用いて実験を行う.

$$T(t, l) = \min \left\{ T^* \left(t, \frac{l-1}{K-1}, \frac{r-1}{K-1} \right), T^* \left(t, \frac{K-l}{K-1}, \frac{K-r}{K-1} \right) \right\} \quad (3.26)$$

$T^*(t, x, y)$ は以下で定義される.

$$T^*(t, x, y) = (T_H - T_L) f_\sigma \left\{ -c_1 \left[\frac{x + \beta_1}{y + \beta_1} - t(1 - c_2) - c_2 \right] - c_3 \right\} + T_L \quad (3.27)$$

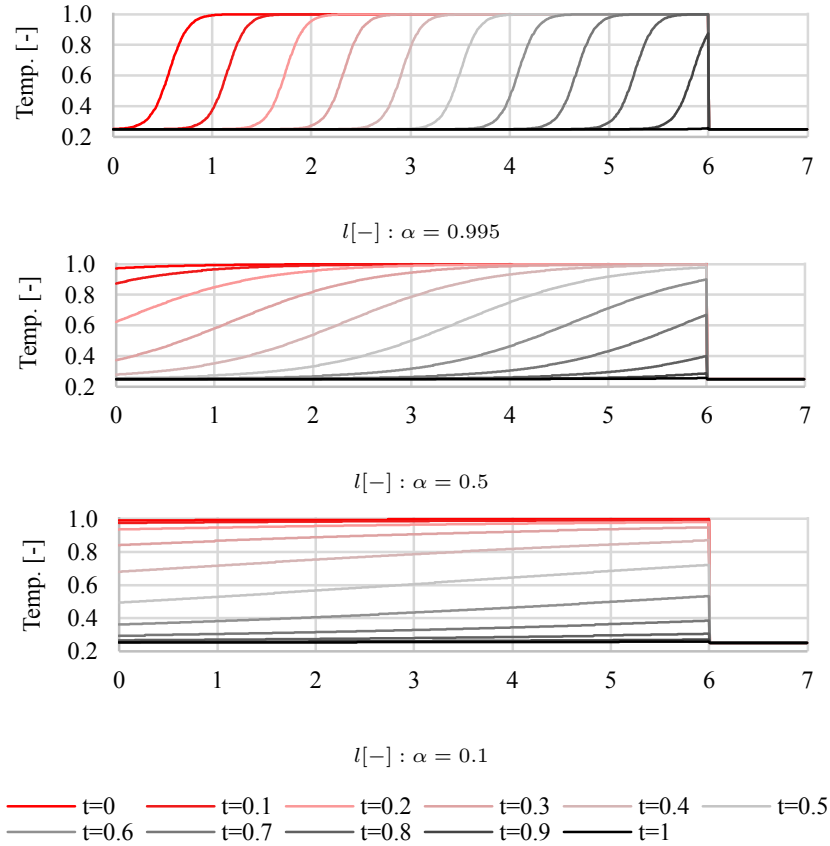


図 3.11 式 (3.26) により定義される温度分布. 様々な α による比較,

ここで, $f_\sigma(x)$ は sigmoid 関数である. $c_1 = 2(K - 1) \ln(1 - \alpha), c_2 = 2c_3/c_1, c_3 = \ln((1 - \beta_2)/\beta_2)$ である. $\alpha (\in (0, 1))$ は温度降下速度を決定するパラメータであり, β_2 は f_σ を $(x, t) = (1, 1)$ において β_2 , $(x, t) = (0, 0)$ において $1 - \beta_2$ とするパラメータである. β_1 は $y = 0$ におけるゼロ除算を防ぐための定数で, 同項中の分子に存在するのは $T^*(t, y, y) = T^*(t, 1 - y, 1 - y), \forall y, t \in [0, 1]$ を満たすことを保証するためである. β_1, β_2 には微小な値を設定している限り, 関数の形状に主たる影響はないが, 数値計算において丸め誤差の影響を受けない程度の小さな値とする必要がある. 倍精度浮動小数点数 [3] の有効数字を考慮し, 余裕をもたせて本節では $\beta_1 = 10^{-8}, \beta_2 = 0.01$ を常に用いる. β_2 は対数関数に含まれており, 丸め誤差の影響を受けやすいため β_1 に比べ大きめの値としている. 図 3.11, 図 3.12 は共に $K = 6, T_H = 1, T_L = 0.25$ における式 (3.26) を描いたものであり, 共通でない設定は次のようになっている.

図 3.11 $(r - 1)/(K - 1) = 1, \alpha \in \{0.995, 0.5, 0.1\}$

図 3.12 $(r - 1)/(K - 1) \in \{0.75, 0.25, 0\}, \alpha = 0.995$

また, 図 3.13 には図 3.11, 図 3.12 と同様の設定で, $(r - 1)/(K - 1) = 1, \alpha \in \{0.995\}$ とした場合の各層における温度の時間変化を示す.

3.2.4 節の評価実験では, 式 (3.28) で定義された, 入力と出力間で線形な勾配を持ち, 時間変化の

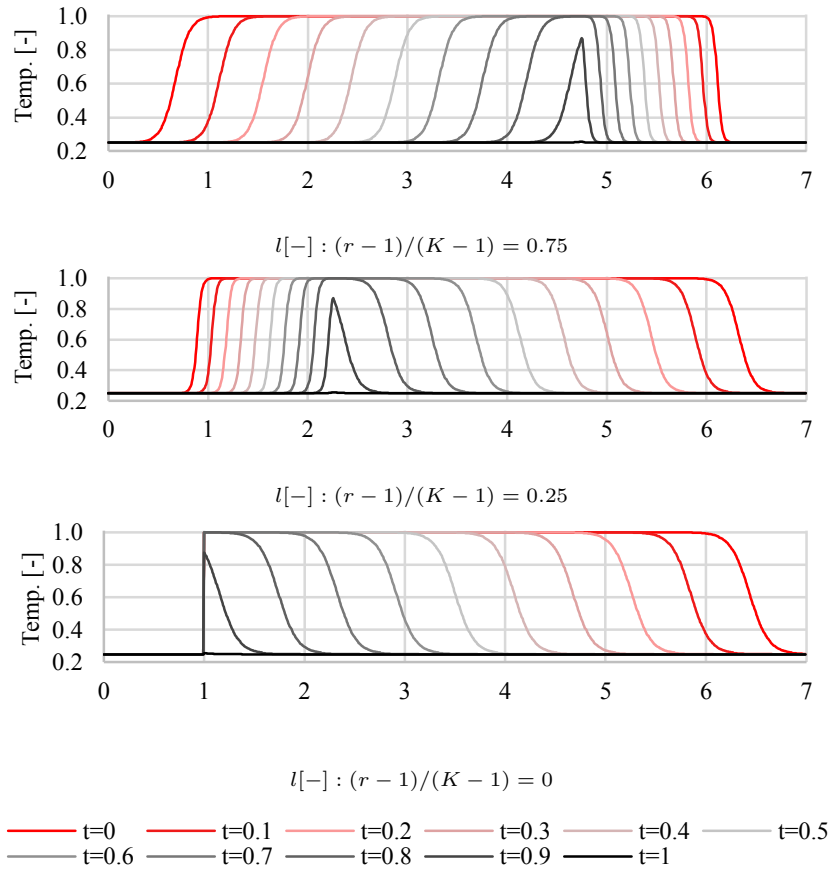


図 3.12 式 (3.26) により定義される温度分布. 様々な $(r - 1)/(K - 1)$ の値による比較.

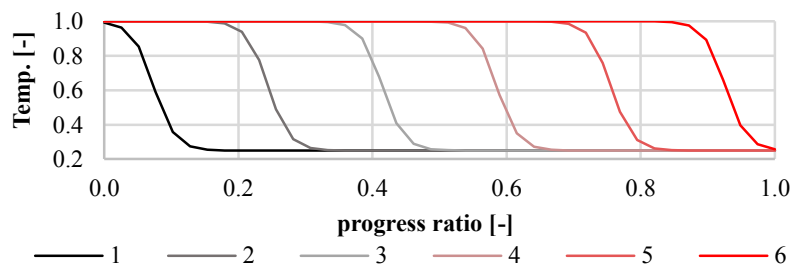


図 3.13 式 (3.26) により定義される温度分布の, 各層における時間変化. 凡例の各数字は層の番号に対応する.

ない温度分布も用いる. これは, 温度を減少させることなく, 温度勾配による情報の流れの制御のみで事後分布の推定精度を改善できるかどうか試験するために用いる.

$$T^*(l) = \frac{T_H - T_L}{K - 1} (l - 1) + T_L \tag{3.28}$$

提案手法のことを本節では Crystalizing Simulated Annealing (CSA) と呼ぶ.

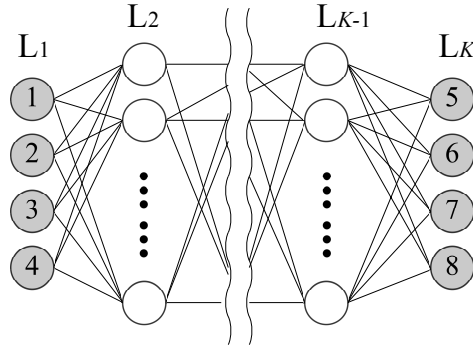


図 3.14 人工データを用いた実験で用いる Boltzmann Machine

3.2.4 評価実験

評価実験においては，人工データセットと手書き文字データセットである MNIST [47] を用い，分類の推論精度を調査する．まず，これらのデータセットについて簡単に説明する．

3.2.4. A 人工データセット

本節で用いる入出力層のユニットの添字集合 Γ と，人工データセットの状態集合は式 (3.29) で定義される．

$$\begin{aligned} \Gamma &= \{1, 2, 3, 4, 5, 6, 7, 8\} \\ \Omega_M^{(\text{data})} &= \{(\delta_{1j}, \delta_{2j}, \delta_{3j}, \delta_{4j}, \delta_{1j}, \delta_{2j}, \delta_{3j}, \delta_{4j}) \mid j \in \{1, 2, 3, 4\}\} \end{aligned} \quad (3.29)$$

δ_{ij} は Kronecker Delta であり，2.1.2 節で説明される通り， $\delta_{ij} = \delta(i = j)$ を満たす．図 3.14 はこのデータセットにおける実験で用いられる BM の構造を示したものである． M は Γ に対応するデータセットの分布のための集合である．また，この実験において各隠れ層のユニット数はすべて 16 とする．以上をまとめると，本実験で用いる BM の添字集合は， $\mathcal{V}_0 = \emptyset, \mathcal{V}_1 = \{1, 2, 3, 4\}, |\mathcal{V}_2| = 16, \dots, |\mathcal{V}_{K-1}| = 16, \mathcal{V}_K = \{5, 6, 7, 8\}, \mathcal{V}_{K+1} = \emptyset, |A| = 8 + 16(K - 2)$ となる．

このデータセットは出力と入力が入力と出力が 1 of K vector [19] となり，またそれぞれの状態が互いに同期するものであり，学習後のモデルに入力の状態を与えた状態で出力が入力と等しくなることを検証する．なお，このデータセットを用いた実験は推論精度および，入力情報が出力まで伝達されることを調査するものである．訓練集合に用いたものと同様のデータセットを試験に用いるため，モデルの汎化性能を評価するものではないことに注意されたい．

3.2.4. B MNIST

MNIST [47] は手書き数字の画像を集めたデータセットであり， 28×28 [pixel] の訓練用画像 60,000，試験用画像 10,000 から構成される．それぞれの画像には対応する正解データとして 0-9 の値が割り当てられている．このデータセットを学習するための BM の構成は， $\mathcal{V}_0 = \emptyset, \mathcal{V}_1 = \{1, 2, \dots, 28^2\}, \mathcal{V}_K = \{|\mathcal{V}_1| + 1, |\mathcal{V}_1| + 2, \dots, |\mathcal{V}_1| + 10\}, \mathcal{V}_{K+1} = \emptyset$ を満たすものとなる．

3.2.4. C 共通の設定

式 (2.60) を学習則として用い, Persistent CD [40] により勾配の値を近似する. この学習において必要なパラメータは以下のように設定する.

- 重み減衰係数: $\alpha_{\text{dec}} = 0.001$
- 慣性項係数: $\alpha_{\text{mom}} = 0.9$
- Batch 集合の大きさ: $\forall t, |\Omega_M^{(t)}| = 8$
- Persistent CD における標本抽出回数: 10 回
- Batch 集合に変更があった場合の Persistent CD における事前標本抽出回数: 20 回

特に断りがない場合, 温度分布の設定値には以下のものを用いる.

- 温度の最低値: $T_L = 0.25$
- 温度の最高値: $T_H = 1$
- 温度の降下速度を決める係数: $\alpha = 0.995$
- 温度の最高点: $(r - 1)/(K - 1) = 1$

推論時において, 焼きなまし法の進行度を式 (3.30) の $\tau \in [0, 1]$ により計算する. 温度分布として式 (3.26) を用いる場合, 第一引数にこの値を用いる.

$$\tau = \frac{c_{\text{samp}} - 1}{N_{\text{pre}}^{(\text{infer})} - 1} \quad (3.30)$$

c_{samp} は標本抽出回数, $N_{\text{pre}}^{(\text{infer})}$ は事前標本抽出回数である. 焼きなまし法は事前標本抽出の間だけ適用し, その結果得られる事後分布の平衡分布から抽出された出力の標本により期待値を推定する. 各パラメータの設定には以下のものを用いる.

- 事前標本抽出回数: $N_{\text{pre}}^{(\text{infer})} = 40$
- 期待値推定のための標本抽出回数: 5

最も高い期待値を持つユニットが入力のクラスに対応する.

学習の終了判定方法は自明でなく, 推論の精度を評価する上で学習の状態による影響が無視できないため, 評価実験では学習曲線により結果を示す. 学習曲線の横軸には式 (3.31) で与えられる学習率の総和を用いる. パラメータの更新回数よりもこちらのほうが学習の進捗をよく表すと考えたためである.

$$S_{\text{eps}}(t) = \int_0^t \epsilon(t) dt \quad (3.31)$$

$\epsilon(t)$ は式 (3.32) により決定する.

$$\epsilon(t) = \epsilon(0) \exp(-\alpha_{\epsilon} t) \quad (3.32)$$

式 (3.32) の曲線を決定するための設定値には以下のものを用いる.

- 学習係数の初期値: $\epsilon(0) = 10$

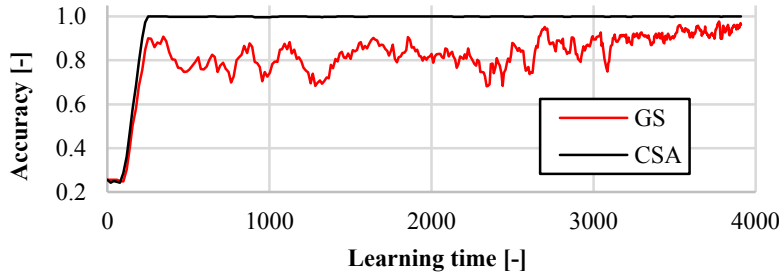


図 3.15 $K = 4$ における，通常の Gibbs Sampling (GS) と提案手法 (CSA) の推論精度の比較.

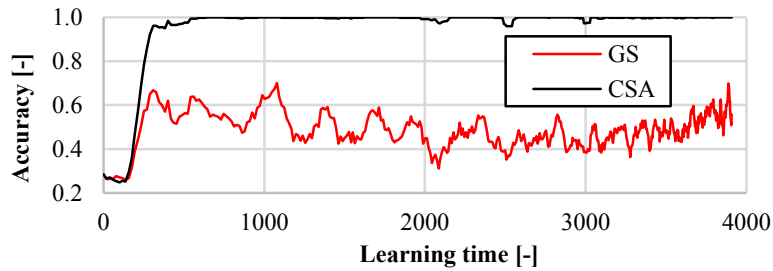


図 3.16 $K = 5$ における，通常の Gibbs Sampling (GS) と提案手法 (CSA) の推論精度の比較.

- 学習係数の減衰係数: $\alpha_\epsilon = \ln 10$

学習曲線の縦軸には式 (3.33) で与えられる \hat{a} を用いる．これは 8 回の試行における移動平均である．

$$\hat{a}(t) = \frac{1}{8} \sum_{j=0}^7 a(t - j\Delta t) \quad (3.33)$$

$a(t)$ は時刻 t における推論精度であり，それぞれの時刻において，テスト集合から無作為に抜き出した 500 回のデータに対する推論の精度を平均したものである．学習曲線の各点は異なるテストデータ集合から算出されており，この曲線に対する各パラメータの影響を考察することは，Cross Validation を行った学習終了時の収束値のみによる評価と同等に妥当なものである．

$\Delta t = 10^{-3}$ とし，各パラメータの更新は t が Δt 進むごとに行う．

学習パラメータの設定値は人工データセット，MNIST 双方において比較的良い結果を示したものである．なお，本実験では，温度分布による推論および事後分布推定精度の違いを調査するものであり，学習の性能を測ることを重視したものではないことに注意されたい．

3.2.4. D 実験結果

層数の違いの影響を調べた結果を図 3.15, 図 3.16, 図 3.17 に示す．GS は従来から用いられている Gibbs Sampling によるものである．いずれの場合も CSA の方が高い推論精度を達成している．また，層数が増加するごとにその差は大きくなっている．さらに，図 3.17 には一様な温度分布を用いる通常の焼きなまし法の結果を SA として示している．3.2.3 節の始めで述べた想定どおり，SA は CSA に比べ推論精度が劣り，GS よりも悪いことがわかる．

図 3.18 には様々な温度設定による推論精度を示す．L は式 (3.28) の温度分布を用いた場合，II

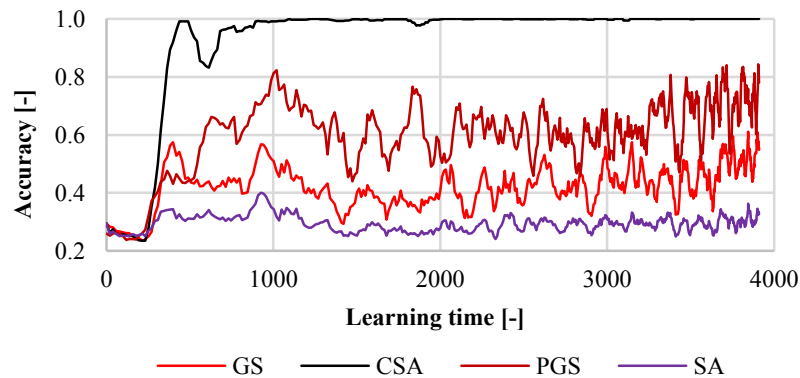


図 3.17 $K = 6$ における，通常の Gibbs Sampling (GS) と提案手法 (CSA)，焼きなまし法 (SA) の推論精度の結果の比較. PGS は予備，実標本抽出回数を他に比べて 4 倍とした Gibbs Sampling の結果

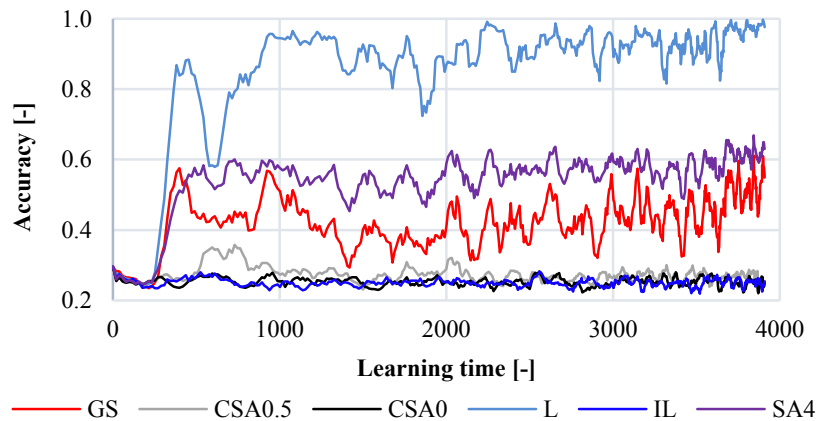


図 3.18 各種温度設定を用いた推論精度の比較. SA4 は $T_H = 4$ を用いた一様温度分布を用いた焼きなまし法，L は入力から出力へ線形に上昇する温度分布，IL は線形に減少する温度分布を用いたもの.

は式 (3.28) の引数に $K + 1 - l$ を代入して得られる分布を用いた場合，SA4 は $T_H = 4$ を用いた場合の結果である．また，CSA0, CSA0.5 はそれぞれ $(r - 1)/(K - 1) = 0, 0.5$ とした結果であり，この設定は IL と同様に出力層において温度勾配が入力側に向かって正となるものである．CSA0, CSA0.5 に用いた温度分布はそれぞれ図 3.12 の 3 段目，2 段目に示す温度分布に近い形状となる．この結果から，出力側付近で温度勾配が入力側に向かって正となる分布では低い精度となることがわかる．また，入力側から出力側へ温度勾配をもたせるだけでも推論精度が改善しており，温度分布が情報の流れを制御するという，3.2.3 節で示した議論が妥当であるとわかる．一方，SA4 においては，開始温度を高くすることで多少，推論精度を改善することはできたが，GS の推論精度との間に大きな差はなく，効果は殆ど無いと言える．

図 3.19 は α の違いによる影響を示したものである．温度分布の α による形状の違いは図 3.11 に示している． α が小さな値における CSA の結果は SA のものに近づくことがわかる． α が小さすぎれば式 (3.26) を用いた温度分布の変化は SA によるものに近づくため，この結果は妥当である．

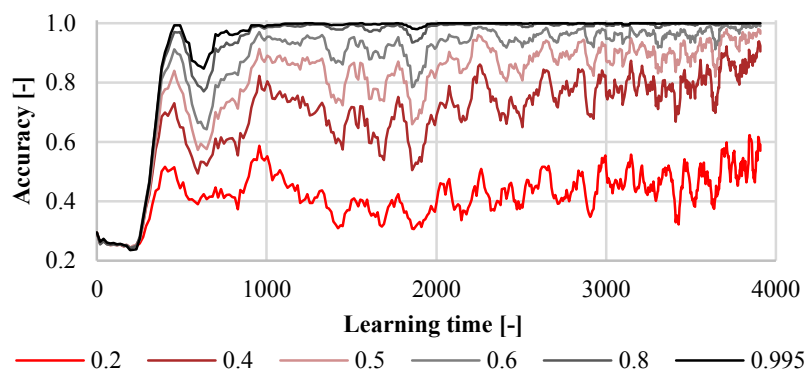


図 3.19 各種温度設定を用いた推論精度に対する、様々な α ($\in \{0.2, 0.4, 0.5, 0.6, 0.8, 0.995\}$) を用いた CSA の結果の比較. 温度分布の形状は図 3.11 で説明したものに対応.

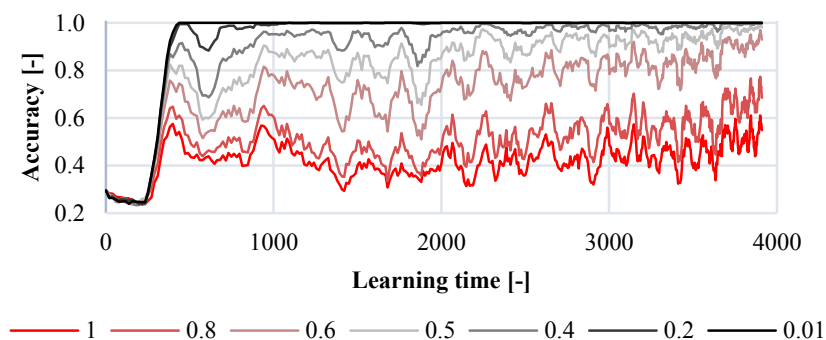


図 3.20 各種温度設定を用いた推論精度に対する、様々な T_L ($\in \{1, 0.8, 0.6, 0.5, 0.4, 0.2, 0.01\}$) を用いた CSA の結果の比較.

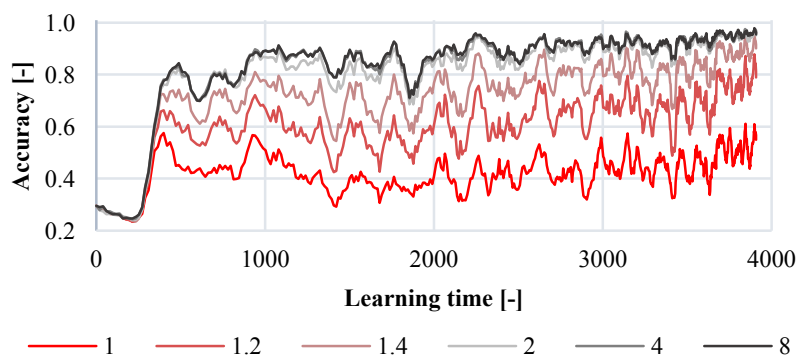


図 3.21 各種温度設定を用いた推論精度に対する、様々な T_H ($\in \{1, 1.2, 1.4, 2, 4, 8\}$) を用いた CSA の結果の比較.

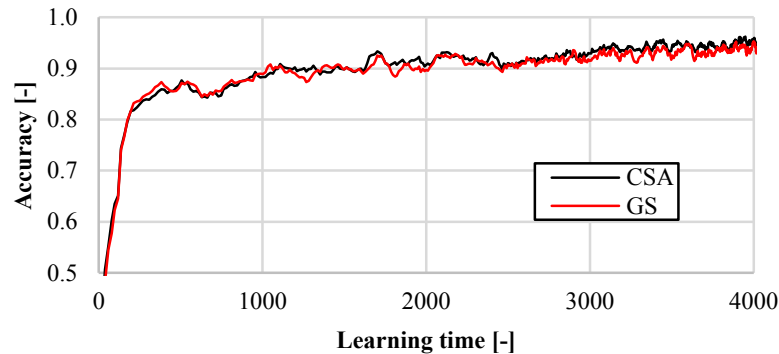


図 3.22 MNIST を用いた実験結果. CSA は提案手法, GS は Gibbs Sampling による結果.
 $|\mathcal{V}_2| = 400, K = 3$

図 3.20 は T_L の違いによる結果を示したものである. T_L が低い設定のものほど高い精度を達成している. この結果と同様の傾向が他の問題に対しても一般に見られるものであるかどうか定かではないが, SA を適用する際, 系の状態を極小点で安定的にするために温度をある程度下げる必要がある, この結果は CSA が内包する SA の性質によるもので妥当なものと考えられる.

図 3.21 は T_H の違いによる結果を示したものである. T_H が高い設定とすることも高い精度を達成可能なことがわかる. また, 精度向上には飽和点があり, T_H を高くするだけでは T_L を低くする程の精度向上は望めないことがわかる. しかしながら, この結果は $T_L = 1$ とした場合でも事後分布推定の精度を改善できることを示しており, 本設定を温度分布を用いない BM の学習法において, 期待値推定のための事前標本抽出に用いることができると考えられる. DBM の学習において, BM の分布の初期化に多くの反復を必要とすることが報告されており, その計算量が問題とされている [38]. これに対し, 提案手法を用いることで学習時における期待値推定の精度も改善できる可能性がある.

図 3.22, 図 3.23, 図 3.24 は提案手法を MNIST の分類問題に適用した結果である. 人工データにおける結果と傾向は同じで, 層数が増加するごとに提案手法と従来手法の差が開いていくことがわかる.

MNIST の結果において, 層が増えるごとに, 推論精度はどちらの手法においても低下している. モデルが複雑化することによる過学習の影響であると考えられることもできるが, 人工データセットを用いた予備実験においても, 標本抽出回数を少なくした時に層が増えるにつれ, 提案手法は高い推論精度を達成することができなかつたため, この精度低下の原因は過学習によるものではなく, 期待値の推定精度の低さに起因している可能性があると考えられる.

3.2.5 まとめ

本節では一様でない温度分布を表現可能な BM の定式化と, FZ 法における温度変化を模した焼きなまし法を用いることで, BM における事後分布の推定精度を向上させる手法を提案した. 人工データセットと手書き文字データセット (MNIST) を用いた評価実験では, 提案手法は Gibbs Sampling に比べ高い推論精度を持つことを示した. この結果は, 学習, 推論双方において同じ手法

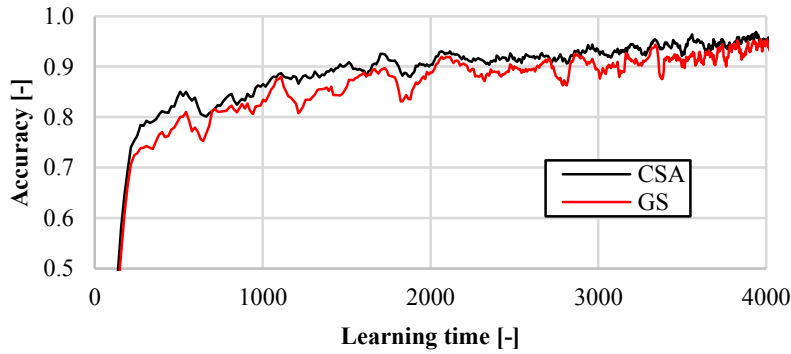


図 3.23 MNIST を用いた実験結果. CSA は提案手法, GS は Gibbs Sampling による結果.
 $|\mathcal{V}_2| = 400, |\mathcal{V}_3| = 100, K = 4$

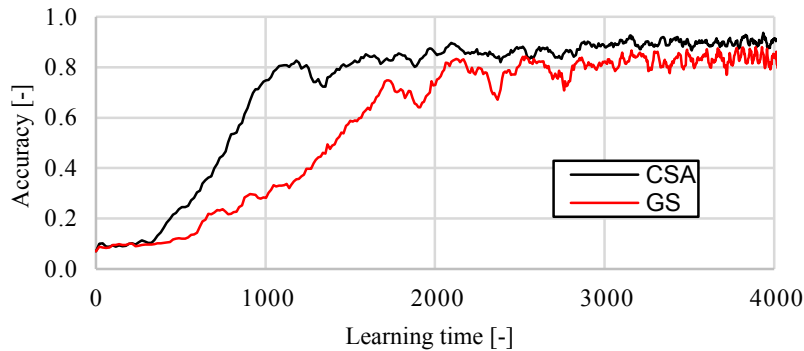


図 3.24 MNIST を用いた実験結果. CSA は提案手法, GS は Gibbs Sampling による結果.
 $|\mathcal{V}_2| = 400, |\mathcal{V}_3| = 100, |\mathcal{V}_4| = 64, K = 5$

を用いることのできる可能性を意味しており, FPGA や ASIC 等の特定用途向けプロセッサを用いた実装等に貢献することが期待できる.

また数学的な解析により, 情報の流れが温度勾配に従うことを示した. この性質により提案手法は高い推論精度を達成したと考える. さまざまな温度分布を用いた実験により, 入力側が低い温度となるものを用いた場合のみ, 高い推論精度を達成できていることから, 得られた理論は妥当であると考えられる.

提案手法を用いた結果において, 焼きなまし法の開始温度を高い値とし, 終了温度として通常の BM の定式化に相当する値を一様に与えた場合にも事後分布推定の精度が改善している. これより, 温度分布による違いが吸収されてしまう BM の学習時においても, 対数尤度関数の勾配を推定するために必要な期待値推定の事前標本抽出に提案手法を用いることで, 学習の精度および速度を改善できる可能性があり, MNIST に対する実験結果において示した, 層が増加するごとに学習自体が困難となる問題に対して, 学習精度を改善できる可能性があると考えられる.

グラフの次元が低く, 高度に構造化された Markov Random Field [53] においては境界条件を与えるユニットから遠くなるユニットが多い. このとき, 3.2.3 節における考察から, DBM よりも独立な分布が内部に生成しやすいと考えられるため, 提案手法の適用を試みる価値は大きい.

第 4 章

観光経路推薦問題に対する焼きなまし法の適用

4.1 はじめに

観光旅行をする際、ガイドブックや旅行会社が提供する観光ツアーを利用するのではなく、SNS などの Web から入手した情報を用いて旅行者自身で観光計画を立案することも増えてきている。観光情報サイトなどでは、エリアなどの指定条件を満たす観光スポットをランキング形式で提示するサービスを提供しているが、知名度の高いスポットや宣伝に力を入れているスポットが上位に表示される傾向がある。ユーザの検索行動として、ランキング上位の数件のみしか確認しないことが

表 4.1 4 章で用いる記号の表

記号	定義	意味
f_c	$\mathcal{E}_p \rightarrow [0, \infty)$	負荷関数
F_c	$\Omega'_A \rightarrow [0, \infty)$	推薦経路の負荷の総和 (経路の経路長, 所要時間など)
f_d	$\mathcal{E}_p \rightarrow [0, \infty)$	価値関数
f_p	$\mathcal{E}_p \rightarrow [0, \infty)$	罰則関数
f_s	$\mathcal{E}_p \rightarrow [0, \infty)$	最短経路負荷関数
C_{pa}	$\in (1, \infty)$	罰則係数
C_{pb}	$\in [0, \infty)$	罰則係数
S	$\mathcal{E}_p \rightarrow 2^{\mathcal{E}}$	最短経路を構成する辺の集合
C_c	$\in \mathbb{R}$	時間制約, 目標経路長
σ^2	$\in (0, \infty)$	違反許容量
U		ユーザの集合
t_{dur}	$\mathcal{N} \rightarrow [0, \infty)$	スポット滞在時間 [27]
t_{queue}	$\mathcal{N} \times \mathbb{R} \rightarrow [0, \infty)$	スポットにおける待ち時間 [27]
f_{pop}	$\mathcal{N} \rightarrow [0, \infty)$	スポットの人気度 [27]
f_{int}	$U \times \mathcal{N} \rightarrow [0, \infty)$	個人のスポットに対する興味 [27]

一般的であるため、結果として個人の趣味嗜好に合った観光地を訪れる機会の損失を招き、一部の観光地へ旅行客が集中するなど、旅行客、観光地側双方に不利益を生じる。こういった問題を防ぐためにも、ユーザの好みに応じたスポットを選び、限られた時間で効率よく巡回する経路を提案する観光経路推薦手法は重要であり、研究が進められている [10][29][27]。

与えられた節点すべてを 1 度ずつ巡回する最短経路を求める問題は、Traveling Salesman Problem (TSP) [9] として古くから研究されている。NP-hard に属する TSP を解くためには近似解法が用いられ、代表的なものには 2-Optimization (2-opt) 法 [25], SA [41] がある。さらに、スポットに価値を付加し、与えられた総負荷制限の中で価値の総和が最大となる経路を選ぶ問題として、Selective Traveling Salesman Problem (STSP) [26] が定式化されている。また、訪れた時間に依存して価値が変化する問題を文献 [10] では最適観光経路問題として定式化しており、文献 [27], [29] ではこれと同様の定式化において、スポットの価値を Web から取得したデータを用いて推定する方法を提案している。

これらの既存研究では、スポットを節点として扱い、任意の節点間における負荷はそれらを結ぶ最短経路に基づき決定している。このことは特定の交通機関や道路しか用いられないことに等しく、混雑の発生につながると考える。また、効率よく移動して多数のスポットを回りたい人もいれば、景観を楽しみながらのんびり移動したい人もいると考えられる。したがって、スポット間の経路選択も、観光経路推薦において考慮すべきと考える。しかし、経路に関する情報は辺への重みとして表現できるが、節点に対する重みを用いて定式化している既存手法に導入した場合、問題の定式化が複雑となり、最適経路の発見を行う手法の構築が容易でなくなる。

本章では、スポットのみでなく、途中の経路における街並みや景観等もユーザにとって重要な因子であると考え、これらを考慮可能な観光経路推薦手法を提案する。提案手法では、既存研究とは異なり、スポットおよび経路に関する価値や負荷すべてを辺に対する重みとして定式化する。これにより、スポットと経路の情報を統一的に扱えるため、探索の過程において両者を同時に調整しながら最適経路の発見が可能となる。最適化方法として、確率場により経路生成をモデル化し、近似的な棄却サンプリングと SA を用いた解法を提案する。人工的な問題と既存研究 [27] で評価に用いられている、テーマパークにおける観光客の行動履歴から作成されたデータセットを用いた評価実験により、提案手法が有効に機能することを示す。

本章で用いる主要な記号を表 4.1 に記載する。

4.2 辺ベクトルによる観光経路推薦問題の定式化

4.2.1 関連研究

TSP には全ての節点を一度ずつ訪れるという制約があるため、スポットに対して取捨選択性のある観光経路推薦問題にそのまま適用することはできない。既存研究では、訪れる節点に関する制約の代わりに、通過する経路の負荷の合計に対応する総所要時間に対して制約を設け、スポットに設定された価値の総和を最大化する問題として定式化するのが一般的であり [10][27][28][29]、代表的なものに STSP [26] がある。

文献 [27] では、待ち時間が頻繁に発生するテーマパークにおける経路推薦手法を提案してい

る。アトラクションを節点として、巡回する節点の総数 D を用い推薦する経路を節点の列で $n = (n_i \in \mathcal{N} | i = 1, \dots, D)$, 各節点を訪れた時刻を $t = (t_i \in \mathbb{R} | i = 1, \dots, D)$ と表現している。 n_1, n_D はそれぞれ固定的な始点、終点であり、最適化の過程で変化することはないが、TSP とは異なり、巡回する節点の総数 D は最適化の過程で変化する。 U をユーザの集合とし、各アトラクションにはユーザごとに異なる価値 (好み) $f_{\text{int}} : U \times \mathcal{N} \rightarrow [0, \infty)$, 全ユーザで共通の人気度 $f_{\text{pop}} : \mathcal{N} \rightarrow [0, \infty)$, ある時刻における待ち時間 $t_{\text{queue}} : \mathcal{N} \times \mathbb{R} \rightarrow [0, \infty)$, アトラクションの利用時間 $t_{\text{dur}} : \mathcal{N} \rightarrow [0, \infty)$ が定義される。また、2つのアトラクション間について、移動時間 $f_c : \mathcal{N}^2 \rightarrow [0, \infty)$ が定義されている。以上の設定において、ユーザ u ($\in U$) に対してテーマパーク内のアトラクションを巡る最適な経路を推薦する問題を、式 (4.2) で与えられる所要時間 C_c ($\in [0, \infty)$) の制約のもと、式 (4.1) に定義される目的関数 F を最大化する n を求める問題と定式化している。式 (4.1) と式 (4.2) による問題の定義は文献 [27] に書かれているものと多少異なるが、説明を簡潔にするため最低限必要な条件のみを満たすように記述している。

$$F = \sum_{i=1}^D \left(\frac{f_{\text{int}}(u, n_i) + f_{\text{pop}}(n_i)}{t_{\text{queue}}(n_i, t_i)} \right) \quad (4.1)$$

$$\sum_{i=1}^D \{f_c(n_i, n_{i+1}) + t_{\text{dur}}(n_i) + t_{\text{queue}}(n_i, t_i)\} < C_c \quad (4.2)$$

解法としては Monte Carlo Tree Search[35] を拡張した PersQ と呼ばれる手法が提案されている。STSP では式 (4.2) の制約の他に、各スポットを訪れる回数が一度以下である制約、始点と終点を必ず訪れる制約、経路が繋がっている制約を満たす必要があるが、PersQ はこれらを満たす解のみを探索するように実装されている。

文献 [10] も同様のアプローチを採用しているが、スポットにおける価値の時間的変化も考慮するように拡張している。CTPlanner [29] は、ユーザと対話的に経路を決定していく支援システムである。節点には対応するスポットに対するユーザの好み価値として与えられる。辺には Google Map^{*1} 等の経路案内サービスを用いてあらかじめ算出した値を移動負荷として設定している。解法には遺伝的アルゴリズム [66] を用いている。

4.2.2 辺ベクトルと Boltzmann 分布による定式化

観光経路推薦問題では、経路に含まれる観光スポットの数は事前に指定されない場合が一般的である。そのため、要素の数が固定である TSP と比べ定式化は複雑となる。提案手法では、自己回帰辺を許容することで、訪問スポット数によらず固定長のベクトルによる表現が可能な定式化を行う。また、道路や交通機関に対する価値や負荷と、スポットに対する価値や負荷を、辺に対する重みとして統一的に扱う方法を提案する。

既存研究では、節点間の経路を事前に最短経路として計算し、任意の節点間に経路が存在する完全グラフを扱っていた。これに対し提案手法では実際の道路レベルで経路を求める。そのため、主要観光スポットだけでなく、地図の交差点や曲がり角なども節点として扱う。

*1 <https://en.wikipedia.org/wiki/Google> or <https://www.google.com>

一般の道路ネットワークに対する適用法については 4.2.4 節で説明し、本節ではまず、自己回帰辺を用いた辺ベクトルによる経路の表現方法について説明する。

辺の集合 \mathcal{E}_p に対し、 D 個の辺からなる経路を含む辺ベクトル $\mathbf{e} := (e_i | i = 1, 2, \dots, D)$ の集合を \mathcal{E}_p^D と定義する。すべての節点を最大で 1 度ずつ訪れる経路までを対象とする場合、 $D = |\mathcal{N}|$ とする。複数回訪れる経路を対象とし、 $D > |\mathcal{N}|$ とすることもできるが、本章ではこの場合を扱わない。 $n_{\text{st}}(e_1)$ は経路の始点、 $n_{\text{ed}}(e_D)$ は経路の終点であり、最適化中変化しないものとする。辺 $e \in \mathcal{E}_p$ のうち、 $n_{\text{st}}(e) = n_{\text{ed}}(e)$ であるものを自己回帰辺と呼ぶ。辺ベクトルは自己回帰辺を含むことで、要素数を固定としながら様々な経路を表現することができる。図 4.1 は節点 1, 2, 7, 8 をたどる経路の例であり、この時の辺ベクトルは $((n_{\text{st}}(e_i), n_{\text{ed}}(e_i)) | i = 1, \dots, D) = ((1, 2), (2, 7), (3, 3), (4, 4), (5, 5), (6, 6), (7, 8))$ となる。辺ベクトルは加法とスカラー倍、距離等が定義される狭義の意味でのベクトルの性質を満たさないが、提案手法では完全グラフではなく実在の道路ネットワークを対象とすることから、 \mathbf{e} は物理空間上に展開されるものであり、また今後の展望として複数経路の結合等の手法が考えられるためこの命名を行った。

提案手法では、最適化手法として SA を用いることを前提とし、辺ベクトルは確率分布から生成されるとモデル化する。SA の実装法として広く用いられている局所最適化法と Metropolis 基準を組み合わせる手法であれば、確率的生成モデルとして明示的に定義する必要はない。しかしながら、提案手法では提案分布の設計も考慮するため、確率的生成モデルとして定式化する必要がある。

\mathbf{e} の確率変数を \mathbf{e}_A 、その分布 $p(\mathbf{e}_A)$ を式 (4.3) により定める。

$$p(\mathbf{e}_A) = \frac{1}{Z_p} \exp\left(-\frac{1}{kT} \phi(\mathbf{e}_A)\right) \quad (4.3)$$

A は辺ベクトルの確率変数に対する添字集合であり、 \mathbf{e}_A の状態空間 Ω'_A は $\Omega'_A \subset \mathcal{E}_p^A$ を満たし、かつ経路としての制約を満たすものであるとする。制約については 4.2.3 節、4.2.4 節で説明する。 $\phi: \Omega'_A \rightarrow \mathbb{R}$ は経路に対する目的関数である。この関数の値が最小となる経路が最適解であり、それら、あるいはそれらに近いものを推薦することを目的とする。

SA を用いて式 (4.4) の分布を近似することを考える。ここでは提案分布を条件付き確率により構成する Gibbs Sampling 法を用いる。後に、4.2.6 節において、より高効率な探索を可能とする提案分布の与え方を提案する。経路変更を行うための最小要素は二つの辺である。変更対象の 2 つの変数の添字を $i, j \in A$ とし、これら以外の変数 $\mathbf{e}_{A \setminus \{i, j\}}$ により条件付けられた分布は式 (4.4) により与えられる。

$$p(\mathbf{e}_{\{i, j\}} | \mathbf{e}_{A \setminus \{i, j\}}) = \frac{\exp\left(-\frac{1}{kT} \phi(\mathbf{e}_A)\right)}{\sum_{\mathbf{y}_{\{i, j\}}} \exp\left(-\frac{1}{kT} \phi(\mathbf{e}_{A \setminus \{i, j\}}, \mathbf{y}_{\{i, j\}})\right)} \quad (4.4)$$

ここで、 $\mathbf{y}_{\{i, j\}}$ の取りうる状態 $\Omega'_{\{i, j\}}$ は辺ベクトルに含まれる経路が、経路としての制約を満たすようなものでなければならない。次節でこの状態変化の与え方について説明する。

4.2.3 TSP への適用と 2-opt 法

TSP における目的関数は、経路上に割り当てられた負荷 (コスト) の総和で与えられる。負荷の例としては辺の両端の節点間を移動するための時間、エネルギー消費量等が想定される。TSP はこのようなコストの総和を最小化する経路を、すべての節点を訪れるという制約のもと求める問題と

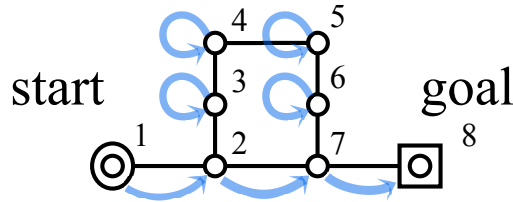


図 4.1 辺ベクトルによる経路表現の説明に用いる, $\mathcal{N} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ であるグラフ. 青は経路であり, 同一辺を直接結ぶものは自己回帰辺である.

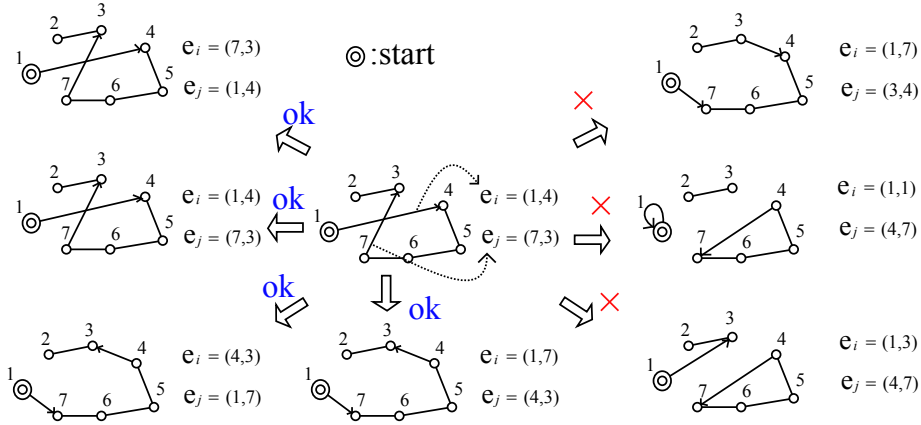


図 4.2 経路に含まれる節点に変更がない状態変化

して定式化される. 式 (4.5) に提案する定式化における TSP の目的関数を示す.

$$\phi(\mathbf{e}_A) = \sum_{i \in A} f_c(\mathbf{e}_i) \tag{4.5}$$

$f_c : \mathcal{E}_p \rightarrow [0, \infty)$ は辺に対する負荷を与える関数であり, 問題に応じて値が定められる. また, 本論文において f_c は辺の向きによらないものとする.

k 個の辺を同時に考慮し, TSP の制約を保つ状態変化を与える手法として, k -opt 法 [25] が提案されている. 提案手法では $k = 2$ の 2-opt 法を用い, 与えられる状態変化を, 条件付き分布の状態集合 $\Omega'_{\{i,j\}}$ として与える. 図 4.2 に, 訪れる節点集合に変化を起こさず, かつ経路としての制約を保つような状態変化を示す. “ok” と示される変化が制約を満たす全ての变化例であり, これらを用いたうえで, 初期状態ですべての節点を通る経路を辺ベクトルに与えた場合, 全ての節点を通る経路のみが解として許されるという TSP の制約を帰納的に満たすことができる. “×” と書かれたものは満たさない例であり, 経路が分離されてしまうもの, 辺の向きが始点から終点へ向かう方向に対して逆行するようなもの等である.

図 4.2 の例を一般化する. Markov 連鎖の $n - 1$ 番目の試行において, 操作対象の辺の添字を $\{i, j\}$ とする. $\mathbf{e}_{A(n-1)} = \mathbf{y}_A$ であるとき, $\mathbf{e}_{A(n)} = \mathbf{e}_A$ の部分空間 $\mathbf{e}_{\{i,j\}}$ の取りうる状態の集合 $\Omega'_{\{i,j\}}$ は式 (4.6) で与えられる.

$$\Omega'_{\{i,j\}} = \{(\mathbf{y}_i, \mathbf{y}_j), (\mathbf{z}_i, \mathbf{z}_j)\}, \text{ where} \\ n_{\text{st}}(\mathbf{z}_i) = n_{\text{st}}(\mathbf{y}_i) \wedge n_{\text{ed}}(\mathbf{z}_i) = n_{\text{st}}(\mathbf{y}_j) \wedge n_{\text{st}}(\mathbf{z}_j) = n_{\text{ed}}(\mathbf{y}_i) \wedge n_{\text{ed}}(\mathbf{z}_j) = n_{\text{ed}}(\mathbf{y}_j) \tag{4.6}$$

ただし、 $\{i, j\}$ は、式 (4.7) の $Q(\mathbf{y}_{\{i, j\}})$ が真 (T) となるものが選ばれるとする。これは、経路上で隣接する 2 辺、あるいは自己回帰辺に対して、この変化を適用しないことを意味する。

$$Q(\mathbf{y}_{\{i, j\}}) = \{n_{\text{st}}(\mathbf{y}_i) \neq n_{\text{ed}}(\mathbf{y}_j) \wedge n_{\text{ed}}(\mathbf{y}_i) \neq n_{\text{st}}(\mathbf{y}_j) \wedge n_{\text{st}}(\mathbf{y}_i) \neq n_{\text{ed}}(\mathbf{y}_i) \wedge n_{\text{ed}}(\mathbf{y}_j) \neq n_{\text{st}}(\mathbf{y}_j)\} \quad (4.7)$$

$Q(\mathbf{y}_{\{i, j\}})$ を真とする $\{i, j\}$ のみが選ばれる Gibbs Sampling の提案分布 (遷移確率分布) は式 (4.8) で与えることができる。 Z_p は式 (4.8) が確率の定義を満たすようにするための正規化定数であるが、実際に計算する必要はない。

$$p(\mathbf{e}_{\Lambda(n)} = \mathbf{e}_{\Lambda} | \mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) = \frac{1}{Z_p} \sum_{i, j \in \Lambda} \delta(Q(\mathbf{y}_{\{i, j\}})) \delta(\mathbf{e}_{\Lambda \setminus \{i, j\}} = \mathbf{y}_{\Lambda \setminus \{i, j\}}) p(\mathbf{e}_{\{i, j\}} | \mathbf{e}_{\Lambda \setminus \{i, j\}}) \quad (4.8)$$

ここで、 $\mathbf{e}_{\{i, j\}} \in \Omega'_{\{i, j\}}$ が成り立っており、また、このとき式 (4.6) と式 (4.7) の関係から $Q(\mathbf{y}_{\{i, j\}}) \Leftrightarrow Q(\mathbf{e}_{\{i, j\}})$ であるので、式 (4.8) は以下の変形から、詳細釣り合い条件を満たすことがわかる。

$$\begin{aligned} & p(\mathbf{e}_{\Lambda(n)} = \mathbf{e}_{\Lambda} | \mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) p(\mathbf{y}_{\Lambda}) \\ &= \frac{1}{Z_p} \sum_{i, j \in \Lambda} p(\mathbf{y}_{\{i, j\}} | \mathbf{y}_{\Lambda \setminus \{i, j\}}) p(\mathbf{y}_{\Lambda \setminus \{i, j\}}) \delta(Q(\mathbf{y}_{\{i, j\}})) \delta(\mathbf{e}_{\Lambda \setminus \{i, j\}} = \mathbf{y}_{\Lambda \setminus \{i, j\}}) p(\mathbf{e}_{\{i, j\}} | \mathbf{e}_{\Lambda \setminus \{i, j\}}) \\ &= \frac{1}{Z_p} \sum_{i, j \in \Lambda} p(\mathbf{y}_{\{i, j\}} | \mathbf{y}_{\Lambda \setminus \{i, j\}}) p(\mathbf{e}_{\Lambda \setminus \{i, j\}}) \delta(Q(\mathbf{e}_{\{i, j\}})) \delta(\mathbf{e}_{\Lambda \setminus \{i, j\}} = \mathbf{y}_{\Lambda \setminus \{i, j\}}) p(\mathbf{e}_{\{i, j\}} | \mathbf{e}_{\Lambda \setminus \{i, j\}}) \\ &= p(\mathbf{e}_{\Lambda(n)} = \mathbf{y}_{\Lambda} | \mathbf{e}_{\Lambda(n-1)} = \mathbf{e}_{\Lambda}) p(\mathbf{e}_{\Lambda}) \end{aligned} \quad (4.9)$$

よって、この操作により構築された MCMC 法により、Boltzmann 分布は正しく計算されることから、SA により最適化を行うことができる。 $\mathbf{e}_{\{i, j\}}$ に対して変更があった場合、経路上の他の辺に対しては巡回方向が正しくなるように修正を加える。この操作は、式 (4.6) に影響を与えないため、詳細釣り合い条件にも影響はない。実際に状態遷移を行うときは式 (4.8) をそのまま用いることはなく、式 (4.7) の右辺が T となる $\{i, j\}$ を一様に選び、式 (4.6) の状態集合と式 (4.4) の条件付き分布により計算される確率に従い、状態を変更するかどうかを決定する。

4.2.4 非存在辺の導入

本節において、問題 (道路ネットワーク等) に含まれる辺を実在辺と呼び、その集合を \mathcal{E} ($\subset \mathcal{E}_p$) とする。また、 $e \notin \mathcal{E}$ である e を非存在辺と呼ぶ。

本研究で対象とするグラフ構造は観光スポットのみを節点とする既存研究よりも複雑、大規模であり、完全グラフに対する局所最適化法をそのまま適用することはできない。そこで、経路を変更していく途中、対応する道が存在しない非存在辺の経路への追加を許可し、一時的に問題のグラフを \mathcal{E}_p で与えられるものに等しいものとする。 \mathcal{E}_p は自己回帰辺も含むため、完全グラフとは多少異なるが、自己回帰辺も考慮した状態変化を与えることでこれに対応する。温度 $T \rightarrow 0$ の極限において、非存在辺がすべて実在辺に置き換わるように目的関数を定式化することで、実在の道路ネット

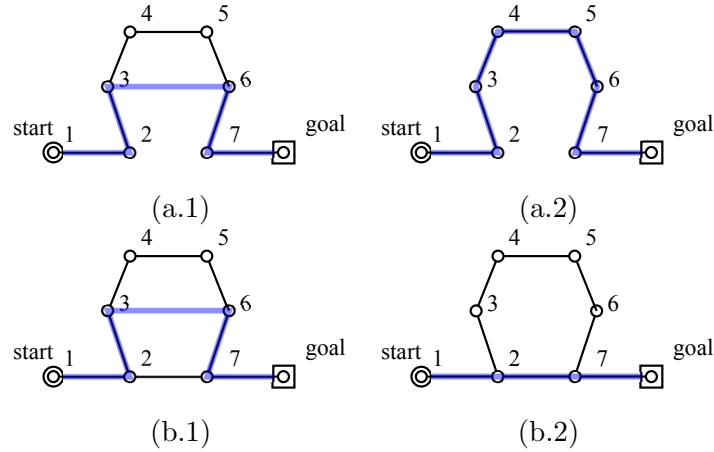


図 4.3 非存在辺と実在辺が入れ替わるにより経路の状態が変化する様子

ワークのようなグラフ構造を持つ問題に対応可能な手法とする．なお，自己回帰辺は実在辺に含まれるとする．

辺の向きによらない，非存在辺における罰則 $f_p : \mathcal{E}_p \rightarrow [0, \infty)$ を導入し，目的関数 ϕ を式 (4.10) に拡張する．ここで， $\forall e \notin \mathcal{E}, f_c(e) = 0$ とする．

$$\phi(\mathbf{e}_A) = \sum_{i \in A} f_c(\mathbf{e}_i) + \sum_{i \in A} f_p(\mathbf{e}_i) \quad (4.10)$$

式 (4.10) の目的関数を用いた分布に従う辺ベクトル \mathbf{e}_A が，温度が 0 になる極限で非存在辺が含まれなくなる条件， $\forall i \in A, \mathbf{e}_i \in \mathcal{E}, \text{at } T \rightarrow 0$ を満たすようにする罰則 f_p の与え方を，図 4.3 の例を用いながら説明する．

この説明に用いる図 4.3 の例において， $\forall e \in \mathcal{E}, f_c(e) = 1$ であるとする．ここで，非存在辺を含む経路 \mathbf{e}^c ，対応する実在辺のみからなる経路 \mathbf{e}^* があるとする．非存在辺が消滅するためには式 (4.11) の関係が満たされる必要がある．

$$\phi(\mathbf{e}^*) - \phi(\mathbf{e}^c) < 0 \quad (4.11)$$

(a.1) から (a.2) に遷移する例では，実在辺が増加することで非存在辺が消える．式 (4.11) に式 (4.10) を代入すると，式 (4.12) を得る．

$$\sum_{s \in S(\mathbf{e}^v)} f_c(s) - f_p(\mathbf{e}^v) < 0 \quad (4.12)$$

ここで， $\mathbf{e}^v (\notin \mathcal{E})$ は非存在辺， $S(\mathbf{e}^v) (\subset \mathcal{E})$ は \mathbf{e}^v の始点，終点を結ぶ実在する最短経路を構成する辺の集合である．

次に，(b.1) から (b.2) へ遷移する例を考える．このとき，節点 2,3,4,5,6,7 を通る経路に遷移した場合でも非存在辺は辺ベクトルからなくなるが，目的関数がより小さくなるのは (b.2) の場合である．このとき，式 (4.12) の総和項の値は，経路から除外される辺の負荷の分だけ減少する．従って，式 (4.12) が満たされるように f_p を設定すれば，この場合も非存在辺が最終的に実在辺に置き換わる．

式 (4.12) を満たす f_p として, 提案手法では式 (4.13) を用いる. $C_{pa} (> 1), C_{pb} (\geq 0)$ は罰則係数である.

$$f_p(e) = \begin{cases} 0 & , S(e) = \{e\} \\ C_{pb} + C_{pa} \sum_{s \in S} f_c(s) & , S(e) \neq \emptyset \wedge S(e) \neq \{e\} \\ \infty & , S(e) = \emptyset \end{cases} \quad (4.13)$$

f_p として別の関数を用いることもできるが, 罰則を過大に設定すると非存在辺への遷移確率を低下させ, 経路探索を行いつらくなる. すなわち, f_p が式 (4.12) の第一項に近いほど経路探索を行いやすくなる. また, f_p はすべての節点対に対して事前に計算しておく必要があり, Floyd-Warshall 法 [36] を用いた場合における, 一般的な道路ネットワークのような低次元なグラフに対して, 必要な記憶量, 計算量は $O(|\mathcal{N}|^2)$ となるが, 計算開始時に一度だけ計算しておけばよく, すべてのユーザで同じものを用いることができるためこの計算負荷は大きな問題とならない.

提案手法では自己回帰辺を自己回帰でない辺に変換し, 経路に追加することで経路を拡張し, また, 経路内の辺を自己回帰辺に変えることで経路を縮小する. 経路が途中で途切れない, 辺の向きが, 経路の始点から終点へ向かう方向と逆行しない制約を満たす中で, 2 辺の操作により実現されるこれらの変化を図 4.4 に示す. 図の見方は図 4.2 と同様に “ok” と書かれたものが制約を満たす変化, “x” と書かれたものが制約を満たさないものの一例である. また, (a) は自己回帰辺を変換し経路に追加する変化, (b) は経路中の辺を自己回帰辺に変換し経路から除外する変化を示す.

図 4.4 に示される変化を一般化する. Markov 連鎖の $n - 1$ 番目の試行において, 操作対象の辺の添字を $\{i, j\}$ とし, $\mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}$ であるとき, $\mathbf{e}_{\Lambda(n)} = \mathbf{e}_{\Lambda}$ の部分空間 $\mathbf{e}_{\{i,j\}}$ の取りうる状態の集合 $\Omega'_{\{i,j\}}$ は式 (4.14) で与えられる.

$$\Omega'_{\{i,j\}} = \{(\mathbf{y}_i, \mathbf{y}_j), (\mathbf{z}_i, \mathbf{z}_j)\}, \text{ where} \\ n_{st}(\mathbf{z}_i) = n_{st}(\mathbf{y}_i) \wedge n_{ed}(\mathbf{z}_i) = n_{ed}(\mathbf{y}_j) \wedge n_{st}(\mathbf{z}_j) = n_{st}(\mathbf{y}_j) \wedge n_{ed}(\mathbf{z}_j) = n_{ed}(\mathbf{y}_i) \quad (4.14)$$

この変化において, 式 (4.15) が満たされるものとする. これは, 経路上で隣接する 2 辺か, 少なくとも一方が自己回帰辺ではない 2 辺の対にのみ, この変化を適用することを意味する.

$$Q(\mathbf{y}_{\{i,j\}}) = \\ \{n_{st}(\mathbf{y}_i) = n_{ed}(\mathbf{y}_j) \vee n_{ed}(\mathbf{y}_i) = n_{st}(\mathbf{y}_j) \vee n_{st}(\mathbf{y}_i) \neq n_{ed}(\mathbf{y}_i) \vee n_{ed}(\mathbf{y}_j) \neq n_{st}(\mathbf{y}_j)\} \quad (4.15)$$

4.2.3 節における議論と同様に, $\mathbf{e}_{\{i,j\}} \in \Omega'_{\{i,j\}}$ が成り立っており, また, このとき式 (4.14) と式 (4.15) の関係から $Q(\mathbf{y}_{\{i,j\}}) \Leftrightarrow Q(\mathbf{e}_{\{i,j\}})$ であるので, 式 (4.15) を満たす $\{i, j\}$ に対する, 式 (4.14) の状態集合を用いた Gibbs Sampling を構成する遷移確率分布は詳細釣り合い条件を満たす.

4.2.5 焼きなまし法を適用可能な STSP の定式化

観光経路推薦では一般的に, 所要時間 (経路長) などの総負荷に関する制約がある中で, 訪問スポットが持つ価値の総和を最大化する手法がとられる. このような定式化は, ユーザが指定した制約を満たす解が確実に得られる利点がある反面, わずかな違反も許さないため, ユーザにとって妥協できる解を捨てている可能性がある. また, この制約は, 所要時間が制約値を超えたときに目的

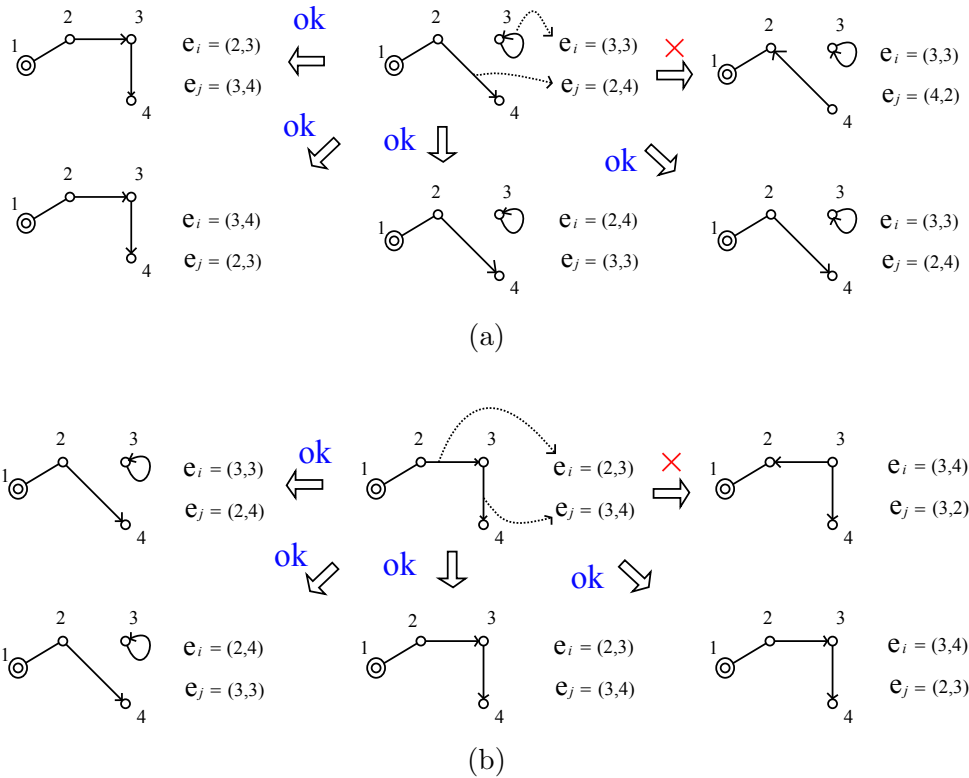


図 4.4 経路に対する辺の追加 (a) と除外 (b) を発生させる状態変化の説明

関数の値が無限大になると定式化することもでき、そのような目的関数に対して SA を適用しても、経路の所要時間が制約値となる付近において、MCMC 法における棄却率が高くなり、効率的な探索を行えないと考える。しかし、観光経路以外の経路推薦においては制約違反を許容することが一般的であり、観光経路推薦においても有効と考える。そこで、提案手法では制約違反に対する許容範囲と、制約違反により得られる価値とのバランスをモデル化することで、観光経路推薦においても多少の制約違反を許容する。著者の知る限り観光経路推薦においてこのような制約違反を許容した研究は存在せず、提案手法の新規性の一つである。

提案手法では、目的関数に対し、負荷についての極値を作ることで、所要時間についての制約を表現する。これにより、所要時間が制約値付近となる状態においても SA を問題なく動作させることができる。辺に対する価値を与える関数を $f_d : \mathcal{E} \rightarrow [0, \infty)$ としてこれを目的関数から減じる。スポットが持つ価値を辺の重みとして統一的に扱うため、スポット内部に 2 節点以上からなる経路を割り当て、それら内部経路における辺の重みとしてスポットの価値を設定する。 f_d の決定方法としては、スポットや道の属性を用いた推定方法 [27], Collaborative Filtering を組み合わせた手法 [31][61] 等の利用が考えられる。以上を考慮した目的関数は式 (4.16) で定義される。

$$\phi(\mathbf{e}_A) = \frac{\beta}{2\sigma^2} |F_c(\mathbf{e}_A) - C_c|^2 + (1 - \beta) \left| F_c(\mathbf{e}_A) - C_c - \frac{\sigma^2}{2} \right| - \sum_{i \in A} f_d(\mathbf{e}_i) + \sum_{i \in A} f_p(\mathbf{e}_i) \quad (4.16)$$

ここで、条件変数 $\beta \in \{0, 1\}$, 総負荷 $F_c : \Omega'_A \rightarrow \mathbb{R}$ はそれぞれ式 (4.17), 式 (4.18) で与えられる。

$$\beta = \delta(\sigma^2 > |F_c(\mathbf{e}_A)|) \quad (4.17)$$

$$F_c(\mathbf{e}_\Lambda) = \sum_{i \in \Lambda} f_c(\mathbf{e}_i) \quad (4.18)$$

f_d は $\forall e \in \mathcal{E}, 0 \leq f_d(e) < f_c(e)$ を満たすものとする．またその値は辺の向きによらない． f_d はこの制約があるため、 $|F_c(\mathbf{e}_\Lambda)| > \sigma^2$ となる範囲で f_d による利得が負荷の増加量を上回らず、負荷の総和に関する制約を無制限に超えることはない． f_d はユーザが対象スポットに対し、その所要時間に見合う価値がどれくらいあると判断しているか、と解釈することができる．負荷、価値とともに辺に対する重みとみなすことで、価値と時間のバランスを調整可能となる．なお、式 (4.16) は $\sigma^2 \rightarrow 0$ の極限において式 (4.17) の条件のもと式 (4.19) に収束する．

$$\phi(\mathbf{e}_\Lambda) = |F_c(\mathbf{e}_\Lambda) - C_c| - \sum_{i \in \Lambda} f_d(\mathbf{e}_i) + \sum_{i \in \Lambda} f_p(\mathbf{e}_i) \quad (4.19)$$

STSP では、経路長の最小化とスポットの探索を並行して行うために、4.2.3 節と 4.2.4 節で導入した 2 種類の状態変化を同時に用いる必要がある．これらの節ではどちらの操作も詳細釣り合い条件を満たし、正しく分布を近似することを示した．詳細釣り合い条件を満たす 2 つの遷移確率分布 p_1, p_2 に対して、式 (4.20) で与えられる p_3 もまた、詳細釣り合い条件を満たす [24]．よって、両節で導入した 2 つの状態変化を同時に用いて最適化を行うことが可能である．

$$p_3(\mathbf{e}_{\Lambda(n)} = \mathbf{e}_\Lambda | \mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_\Lambda) = \sum_{i=1,2} a_i p_i(\mathbf{e}_{\Lambda(n)} = \mathbf{e}_\Lambda | \mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_\Lambda) \quad (4.20)$$

ここで、 $a_1, a_2 (\in [0, 1])$ は結合係数で、 $a_1 + a_2 = 1$ を満たす．

4.2.6 Boltzmann 定数と提案分布

Boltzmann 定数の設定方法を述べる．温度 T は 1 から 0 へ変化させていくとし、Boltzmann 定数 k により温度の影響度を決定する．Markov 連鎖の $n-1$ 番目の状態が、 $\mathbf{e}_{\Lambda(n-1)} = \mathbf{y}_\Lambda$ であり、 n 番目の候補を $\mathbf{e}_{\Lambda(n)} = \mathbf{e}_\Lambda$ であるとし、式 (4.4) の条件付き分布の評価対象である $\{i, j\} \subset \Lambda$ の添字上に定義される $\mathbf{e}_{\{i,j\}}$ について考える． $\mathbf{y}_{\{i,j\}}$ から $\mathbf{e}_{\{i,j\}}$ への変化による目的関数の変化量 $\Delta\phi = \phi(\mathbf{e}_\Lambda) - \phi(\mathbf{y}_\Lambda)$ は最大で、 $f_c(\mathbf{e}_i), f_c(\mathbf{e}_j)$ と同じオーダーとなる．式 (4.4) は ϕ が減少するような状態の確率を高く与えるが、温度が高い時には、増加する方向に対しても遷移する確率が現実的な値であることが望ましい．式 (4.6)、式 (4.14) の状態集合はともに、 $\mathbf{y}_{\{i,j\}}$ から $\mathbf{e}_{\{i,j\}}$ に変化するかしないかの 2 通りだけとなっており、この場合変化しない確率のみを考え、変化する確率はその余事象とすることができるため、式 (4.4) は式 (4.21) と表せる．

$$p(\mathbf{e}_{\{i,j\}} = \mathbf{y}_{\{i,j\}} | \mathbf{e}_{\Lambda \setminus \{i,j\}}) = \frac{1}{1 + \exp\left(-\frac{1}{kT} \Delta\phi\right)} \quad (4.21)$$

このことから、 $k \approx \max(\Delta\phi) < \max_{x,y \in \mathcal{E}_p} (|f_c(x)| + f_c(y))$ であれば、温度 $T = 1$ において考えられる最大の $\Delta\phi$ を扱えることになる．実際には $\max_{x,y \in \mathcal{E}_p} (|f_c(x)| + f_c(y))$ を最大値とみなすことは過剰な設定であることが多いため、与えられた問題に対するある程度の予備実験を行い、全ての経路を探索可能である中で出来る限り小さい k を決定後、ユーザに対する経路生成においては同一の値を用いる．

MCMC 法として通常の Gibbs Sampling を用いた場合、あらゆる辺の組み合わせに関する変化を無作為に試行する提案分布となっている。しかしながら、互いに遠くにある 2 辺による経路の状態変化は起こりづらく、そのような状態変化を等しく試す提案分布は無駄が多い。ここでは、目的関数の増減量を近似的に見積もることで状態変化が起こる可能性が極端に小さくならない 2 辺を選ぶ提案分布の構築法を提案する。

無作為に選ばれた $\mathbf{y}_{\{i,j\}}$ はそのほとんどが式 (4.21) を 1 とする、経路の変更を起こさないものとなり無駄が多い。提案手法では、以下のアルゴリズムに従い $\{i,j\}$ を選択する。

1. $n_{\text{st}}(\mathbf{y}_i) \neq n_{\text{ed}}(\mathbf{y}_i)$ を満たすものの中から一様に $i \in (A)$ を選ぶ。
2. $v \in \{n_{\text{st}}(\mathbf{y}_i), n_{\text{ed}}(\mathbf{y}_i)\}$ を満たすものの中から一様に $v \in \mathcal{N}$ を選ぶ。
3. $x \in \mathcal{E}_p, n_{\text{st}}(x) = v, n_{\text{ed}}(x) = s$ を満たし、 $f_p(x)$ の値が小さい方から K 番目までとなるものの中から一様に $s \in \mathcal{N}$ を選ぶ。
4. $\mathbf{y}_j \in n_{\text{st}}^{-1}(s) \cup n_{\text{ed}}^{-1}(s)$ を満たすものの中から一様に $j \in (A)$ を選ぶ。

ステップ 3 における近傍探索数 K は温度 T に比例した値を選ぶべきと考えられるが、本論文では温度によらず一定の値としている。なお、 K は式 (2.36) の全探索性を満たすように設定する必要がある。具体的には、任意の 2 節点 $x, y \in \mathcal{N}$ について、 x から y へ至る経路を構成することを考え、このとき、途中の節点 n に接続する実在辺の内、負荷 f_c の低いものから K 番目以内の辺のみを選んだとしても、 x, y をつなぐ経路を構成できるようにする必要がある。このように設定することで、ある経路を示す状態から、辺の取捨選択を繰り返すことで別の経路を示す状態へ移れることになり、全探索性を満たすことが可能である。

本手法の条件を命題としてまとめると、式 (4.22) で定義される $Q: \mathcal{E}^2 \times \mathcal{N}^2 \rightarrow \{\text{T}, \text{F}\}$ となる。 $N: \mathcal{N} \rightarrow 2^{\mathcal{N}}$ はステップ 3 において、節点から s の候補となる節点の集合への写像である。

$$Q(\mathbf{e}_i, \mathbf{e}_j, v, s) = [n_{\text{st}}(\mathbf{e}_i) \neq n_{\text{ed}}(\mathbf{e}_i) \wedge v \in \{n_{\text{st}}(\mathbf{e}_i), n_{\text{ed}}(\mathbf{e}_i)\} \\ \wedge s \in N(v) \wedge \mathbf{e}_j \in n_{\text{st}}^{-1}(s) \cup n_{\text{ed}}^{-1}(s)] \quad (4.22)$$

式 (4.4) は式 (4.21) の線型結合と、式 (4.22) の条件を遷移確率分布として定義すれば式 (4.23) と書ける。 p_1 は 4.2.3 節で定義した遷移確率分布、 p_2 は 4.2.4 節で定義した遷移確率分布とする。

$$p(\mathbf{e}_{A(n)} = \mathbf{e}_A | \mathbf{e}_{A(n-1)} = \mathbf{y}_A) = \\ \frac{1}{Z_p} \sum_{i,j \in A, v, s \in \mathcal{N}} \sum_{k=1,2} \frac{1}{2} p_k(\mathbf{e}_{A(n)} = \mathbf{e}_{\{i,j\}} | \mathbf{e}_{A(n-1)} = \mathbf{e}_{A \setminus \{i,j\}}) \delta(Q(\mathbf{e}_i, \mathbf{e}_j, v, s)) \quad (4.23)$$

式 (4.23) の詳細釣り合い条件を導くことは容易でなく、別の定式化の考案を含めた解析が今後の課題として挙げられる。しかしながら、4.2.7 節の評価実験において、本節で提案するアルゴリズムは有効に機能することを示す。この実験結果から、本アルゴリズムを用いた $\{i,j\}$ の選択は、元の分布を近似可能な提案分布を構成していると推測される。

4.2.7 評価実験

提案手法の有効性を評価実験により検証する。4.2.7. A 節では人工データを用い、TSP に対し適用した結果、および非存在辺の消滅について検証した結果を示す。これらについては提案分布に、近

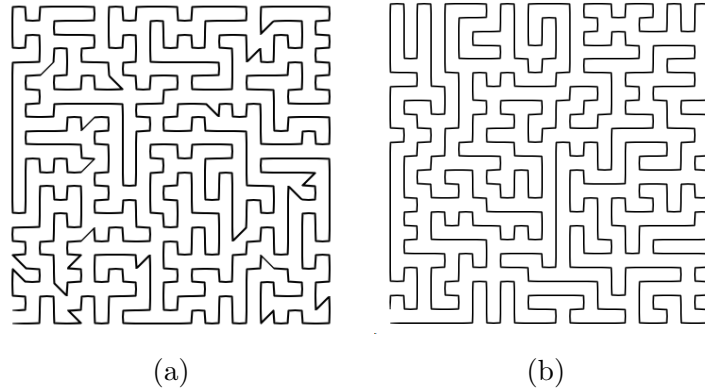


図 4.5 最適化性能の比較. (a): 無作為な探索を行う提案分布によるもの. (b): 近傍探索を行う提案分布によるもの

傍探索手法を用いた場合と、網羅的な選択を行った場合の比較も示す. 4.2.7. B 節で実問題に対する適用結果を示す. 本節では、計算の進度の割合を $t \in [0, 1]$ であらわし、 $t = 0$ が初期状態、 $t = 1$ が最終状態を意味する. 式 (4.13) の定数には $C_{pb} = 1.005$, $C_{pa} = 0.001$ をすべての実験で用いた.

4.2.7. A 人工データによる検証

経路に対する辺の追加、除外を行わず、入れ替えのみを考慮し、問題を完全グラフとした場合、対象問題は TSP となる. 幅 1 で等間隔に配置された 24 行 24 列の節点間に幾何学距離に等しい負荷を与えた、総節点数 576 の完全グラフに対し、提案手法を適用して求めた経路を図 4.5 に示す. 温度は $T = 1$ で一定、Boltzmann 定数には $k = 0.1$ を用いた. 近傍探索数は $K = 8$ とし、無作為に選んだ 2 辺の状態変化を繰り返して求めた結果を (a)、近傍探索の提案分布を用いて求めた結果を (b) にそれぞれ示す. また、図 4.6 は近傍探索の提案分布を用いた場合 (Neighbor)、用いないで総当たり試行を行った場合 (Normal) それぞれについて、目的関数の変化を示している. この問題における最適解の経路長 (目的関数の値) は 574 となる. これらの結果より、近傍探索の提案分布を用いた解法の方が、高速に目的関数を減少させ、また、図 4.5(b) の結果からは、最終的に最適解を得られていることがわかる.

次に、非存在辺が $T \rightarrow 0$ において消滅することを検証する. まず、 $K = 16$ の近傍探索の提案分布を用いて検証した結果を図 4.7 に示す. (a) は $t = 0$, (b) は $t = 0.46$, (c) は $t = 1$ の状態である. 実在辺を黒、辺ベクトルに含まれる経路を青でそれぞれ示している. 図 4.8 は近傍探索の提案分布を用いた場合 (Neighbor)、用いないで総当たり試行を行った場合 (Normal) それぞれについて、目的関数の変化を示している. 図 4.8 より、計算が進むにつ入れ、非存在辺が持つ僅かに大きな負荷が減ることで目的関数が減少しており、このことから、非存在辺が実在辺に置き換わっていることがわかる. また、近傍探索の提案分布を用いた結果の収束が速いことがわかる.

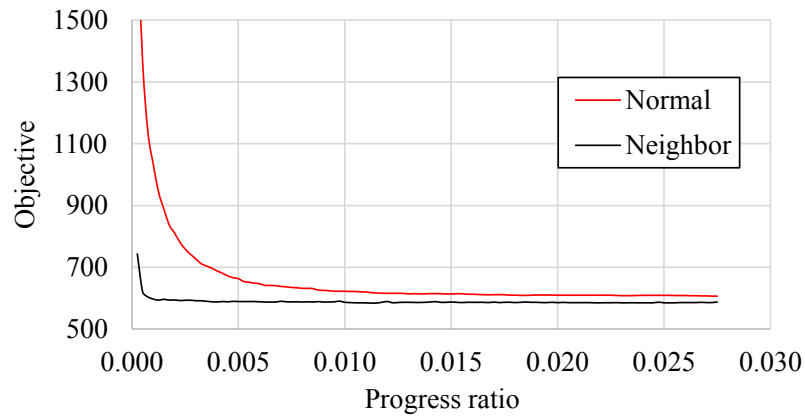


図 4.6 図 4.5 の実験における目的関数の変化. Normal: 無作為試行, Neighbor: 近傍探索

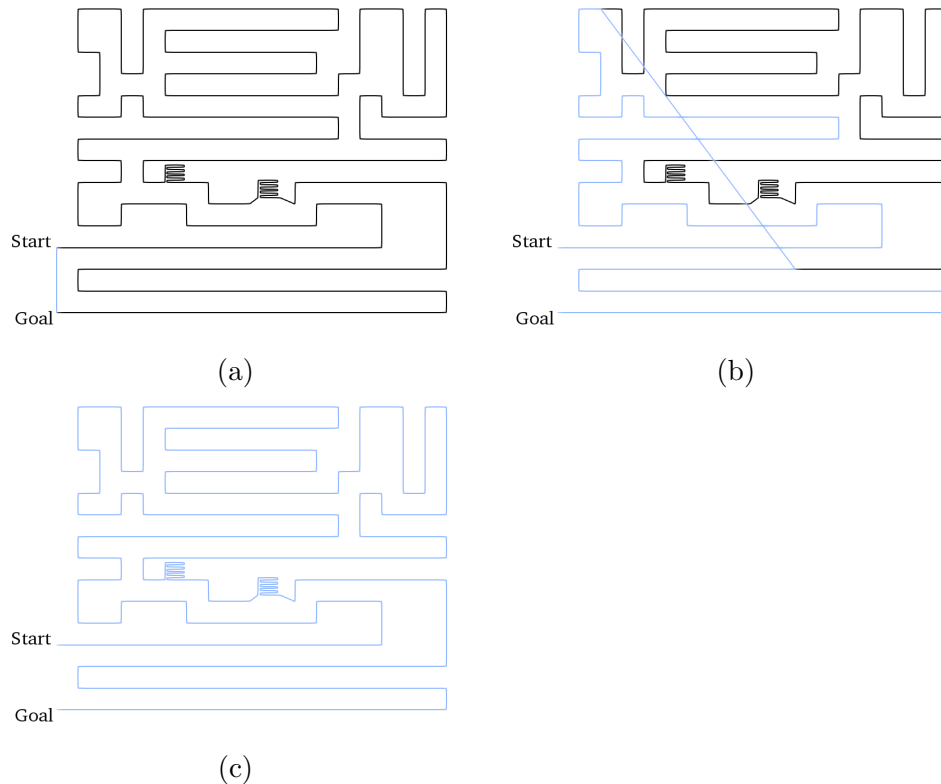


図 4.7 非存在辺が消滅することを確認する実験. (a): $t = 0$, (b): $t = 0.046$, (c): $t = 1$ における状態.

4.2.7. B 実データによる評価実験

文献 [27] が提供しているデータセットを用いて比較実験を行った. このデータセットはアメリカ合衆国の Walt Disney World^{*2}にある 5 つのテーマパーク, Epcot, Disneyland, California Adventure, Disney Hollywood, Magic Kingdom に関するものであり, 2007 年 8 月から 2016 年

^{*2} <https://disneyworld.disney.go.com>.

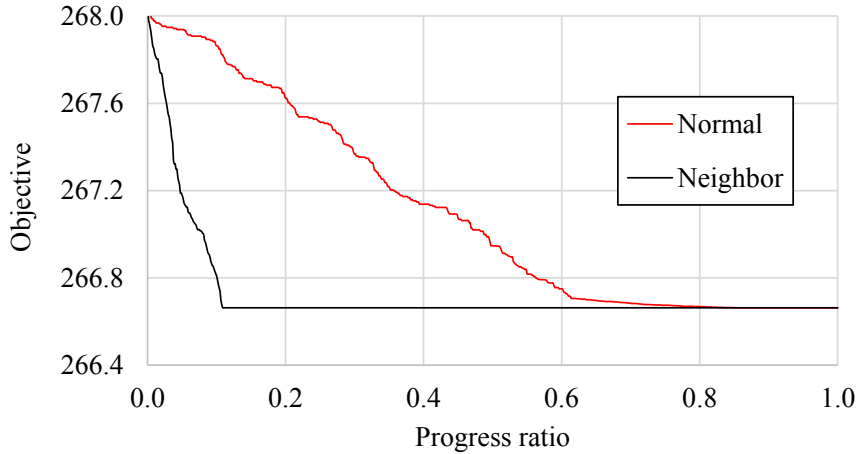


図 4.8 図 4.7 における目的関数の変化. Normal: 無作為試行, Neighbor: 近傍探索

8 月の間にユーザーによって撮影された座標, 時刻情報付きの写真をもとに作られたユーザーの行動履歴と, 各テーマパーク内アトラクションに関する情報, アトラクション間の移動時間, Wikipedia*³の情報をもとに作られた, アトラクションが持つ属性値から構成されている.

文献 [27] は待ち時間の削減を主な目的とした研究であり, 問題の定式化も異なるため, ここでは提案手法で利用可能な形式にデータを変更して用いた. そのため, 結果の厳密な比較はできないことに注意されたい. 基本的には [27] と同様の算出方法を用いるが, 本論文独自のものについてはその都度明記する.

提案手法においては価値が辺に対して与えられるため, 一つのスポットに対し 2 つの節点を割り当て, その間に辺を新たに構築した. 辺の負荷 f_c は平均待ち時間と利用時間を用いて式 (4.24) により求められる.

$$f_c(e) = t_{\text{dur}}(n(e)) + \frac{1}{|U|} \sum_{u \in U} t_{\text{queue}}^{(u)}(n(e)), e \in \mathcal{E} \quad (4.24)$$

$n(e)$ は辺 e の元となった節点を示す. $t_{\text{queue}}^{(u)}(n)$ はユーザー u の行動履歴から算出された, 節点 n に対応するスポットでの待ち時間である. 長いもので 30 分, 1 時間等のスポットも存在している. 一方, 0 [s] 以下の値に算出されるスポットもあったため, 待ち時間が 120 [s] 以下となるスポットの待ち時間は全て 120 [s] としてある. f_d に関しては提案手法独自の要素であるため, 文献 [27] におけるスポットの価値推定の考え方を参考に, 式 (4.25) を新たに定義して用いた.

$$f_d(e) = \max \left(1, \lambda_p^{(u)} q_0(n(e)) + \mathbf{q}_{\text{map}}(n(e))^T \mathbf{q}(u) \right) t_{\text{dur}}(n(e)), e \in \mathcal{E} \quad (4.25)$$

$$q_0(n(e)) = \frac{N_{\text{visit}}(n(e))}{\max_{x \in \mathcal{N}} N_{\text{visit}}(x)}, e \in \mathcal{E} \quad (4.26)$$

$N_{\text{visit}}(x)$ は節点 x の訪問者数であり, 同一ユーザーが複数回訪問した場合もそれぞれカウントしている. $\mathbf{q}(u) := (q_1(u), q_2(u), \dots, q_L(u))$ はユーザー u が持つ L 次元好みベクトルであり, 各要素はアトラクションが持つ属性 (L 種類) に対応している. 節点に対しても同様に L 次元属性ベクトル

*³ <https://www.wikipedia.org>.

\mathbf{q}_{map} があり, これを用いて, $\mathbf{q}(u)$ を式 (4.27) でモデル化する. $\mathbf{q}_{\text{map}}(x)$ の各要素は節点 x が対応する属性を持つときに 1, そうでないときに 0 となるものとする. $\mathcal{N}(u)$ はユーザ u が訪れた節点の集合である. なお, ひとつのスポットは複数の属性を持ちうる.

$$\mathbf{q}(u) = \frac{\hat{\mathbf{q}}(u)}{\max_{i \in \{1, 2, \dots, L\}} (\hat{\mathbf{q}}(u))_i} \quad (4.27)$$

$$\hat{\mathbf{q}}(u) = \sum_{x \in \mathcal{N}(u)} \mathbf{q}_{\text{map}}(x) \quad (4.28)$$

$\lambda_p^{(u)}$ はユーザ u の好みがどれだけ大衆の人気に左右されるかを定める値であり, 式 (4.29) により定義される. この値が大きいほど, 人気スポットを優先して経路を推薦する.

$$\lambda_p^{(u)} = 1 - \frac{\sum_{i=1}^L \max\{2, (\hat{\mathbf{q}}(u))_i\}}{2L} \quad (4.29)$$

$(\hat{\mathbf{q}}(u))_i$ は i 番目の属性を持つスポットをユーザ u が訪問した回数に等しい. ある属性を持つスポットを 2 回以上訪問したユーザはその属性に対して十分な知見を持っているとみなし, 人気よりも自身の興味を優先するようにモデル化している.

文献 [27] が提供するデータセット内で, 各節間の移動負荷 $t_{\text{dur}}(n)$ は明示的に与えられている. また, ユーザ u の行動履歴 $V(u) := (((n_{uki}, t_{uki}) | i = 1, 2, \dots, H(k, u)) | k = 1, 2, \dots, K(u))$ が与えられているので, ここから, $t_{\text{queue}}^{(u)}$ を求める [27]. $K(u)$ はユーザ u が持つ旅程の数であり, $H(k, u)$ は u が k 番目の旅程で訪れたアトラクションの数である. n_{uki} はユーザ u が k 番目の旅程において i 番目に訪れたスポット, t_{uki} は n_{uki} の訪問時刻を意味する. $t_{\text{queue}}^{(u)}$ を求める手順を次に示す.

1. $K(u) \geq 2$ であり, 所要時間が 2 時間を超えている旅程を少なくとも一つは持つユーザからなる集合 $\mathcal{U}_{\text{valid}}$ を求める.
2. ユーザ u ($\in \mathcal{U}_{\text{valid}}$) の $V(u)$ を構成する旅程のうち, 所要時間が 2 時間を超えているものから一つを無作為に選び評価用, 残りをモデル構築用とする. また, ユーザ u ($\notin \mathcal{U}_{\text{valid}}$) のデータはすべてモデル構築に用いる. u の経路 $k = 1, 2, \dots, K(u)$ における所要時間は, 最初と最後のアトラクションを訪れた時刻の差 $t_{ukj} - t_{uk1}$, $j = H(k, u)$ とする.

評価用の経路を無作為に選ぶことを繰り返し, Leave One Out Cross Validation[18][30] によりユーザごとの Recall (再現率) を求める. Recall はユーザの行動履歴に含まれるスポットのうち, 推薦経路に含まれるものの割合である. 文献 [27] では, Precision (適合率) も評価に用いているが, 与えられた時間的制約の下で, 実際よりも多数のスポットを巡る経路を推薦できた場合に適合率は低下してしまうため, 本論文では評価指標として採用しない.

Boltzmann 定数 $k = 1200$ [s] を用い, 温度履歴には $T(0) = 1, T(0.8) = 0.1, T(1) = 0$ で与える区分線形関数を用いた. $t = 0.8$ 以降冷却速度が小さくなるようにし, 細かい調整が行われるようにしている. k は 4.2.6 節で述べた通り, 辺に対するコストの最大値程度に設定することがすべての経路を検索するための十分条件であるが, あまりにも大きすぎると計算時間が増え問題となる. また, 実際にはこれより小さな値でも確率的に全体を探索できることが期待できる. データセットを調べた結果, 各アトラクションの滞在時間は 1 時間弱程度のものが多かったため 500 [s] から 3000 [s] の間で予備実験を行い, $k = 500$ [s] 程度でも十分に全体が探索されることを確認し, 余裕

表 4.2 Recall の比較: Edge Vector: 提案手法 PersQ: 既存研究 [27].

Method	Cali.Adv.	Hollywood	Epcot	Disneyland	Magic King.
PersQ	0.483	0.482	0.472	0.332	0.440
IHA	0.332	0.367	0.380	0.267	0.305
UBCF-I	0.258	0.297	0.291	0.235	0.265
PersTour	0.227	0.305	0.310	0.190	0.202
TourInt	0.228	0.302	0.317	0.188	0.200
TripBuild	0.225	0.309	0.300	0.176	0.193
Edge Vector	0.299	0.459	0.467	0.254	0.410

表 4.3 所要時間制約に対する Recall の比較.

C_c 's scaling					
factor	Cali.Adv.	Hollywood	Epcot	Disneyland	Magic King.
0.8	0.249	0.375	0.371	0.220	0.331
1.0	0.299	0.459	0.467	0.254	0.410
1.2	0.391	0.523	0.531	0.294	0.486

表 4.4 所要時間制約に対する推薦経路所要時間の RMSE [s] の変化.

C_c 's scaling					
factor	Cali.Adv.	Hollywood	Epcot	Disneyland	Magic King.
0.8	593	494	332	1186	964
1.0	892	2205	1003	2517	1965
1.2	1333	5010	3008	3724	4226

を持たせ $k = 1200$ [s] を設定値に用いた. また, 式 (4.16) の C_c には実験手順 2 で説明した所要時間を用い, $\sigma^2 = 0.01C_c$ とした. ユーザのテーマパークにおける所要時間の平均値と標準偏差は 17200 ± 9724 [s] であり, σ^2 はこれよりも小さい. 以上の設定による実験結果を示す.

表 4.2 には文献 [27] で報告されている既存研究の結果も含めた Recall 値を示している. この中で, PersQ が文献 [27] で提案されている手法, Edge Vector が提案手法である, また, 提案手法の特徴の一つである, 所要時間に関する制約 (所要時間制約) の変化に対する対応可能性を検証するため, C_c の値を変化させた場合における Recall および推薦経路の所要時間に対する RMSE (Root Mean Squared Error) の変化を表 4.3, 表 4.4 にそれぞれ示す. 0.8, 1.0, 1.2 の表記は, 所要時間に基づき設定した元の C_c の値に対する倍率を意味し, 1.0 の結果が表 4.2 のものと同一である.

データセットの利用方法で異なる部分があるため厳密な比較はできないが, 表 4.2 より Recall に関して同程度の結果が得られていることがわかる. また, 表 4.3 より, Recall が所要時間制約の増加に伴い上昇していることから, 与えられた所要時間制約に基づき, 高い価値が割り当てられたスポットを優先して巡る経路を推薦できていることもわかる.

表 4.4 に示した時間のずれ (RMSE) については, σ^2 の値により期待される量 (200 [s] 程度) より

大きいものとなっている。データセットを確認したところ、数時間に及ぶ所要時間のアトラクションが多数存在しており、所要時間の細かい調整が難しい問題であったことが原因として考えられる。例えば Disneyland では 12000 秒に達するものが存在した。また、テーマパーク内のすべてのアトラクションを回る経路を推薦したとしても、滞在時間 (C_c) が余るユーザが存在しており、このようなユーザへの推薦結果が RMSE に大きく影響していた。このようなユーザに対する誤差は、 C_c の倍率を上げるとさらに大きくなるため、倍率の増加に伴い RMSE が増加する傾向になったと考える。それでも、 C_c の倍率が 1.0 (実際の滞在時間に対応) の場合のずれは平均 30 分程度に収まっており、半日あるいは 1 日程度は滞在すると想定されるテーマパークにおける経路推薦としては問題ないと考えられるが、細かい調整が行える手法、例えば食事コーナーで時間をつぶす等の推薦を行える手法がこのような問題に有効だと考えられ、今後の課題として挙げられる。

計算時間に関しては、文献 [27] では PersQ の実行に 1 ユーザ当たり 3 分程度かかるように調整したとあるが、これに対し提案手法は表 4.2 の結果を出すために 1 人当たり 0.5 秒以下の時間を要するにとどまっており、高速に同等の性能が得られていることがわかる。

4.2.8 まとめ

本節では、経路推薦問題を、スポットや道の価値、移動に要する負荷などを全て辺に対する重みとして統一的に扱い、定式化する手法を提案した。提案する定式化により、違反許容量の範囲で生成される経路長が制約を違反することを許容することで、違反により得られる価値とのバランス調整が可能となるほか、辺ベクトルと自己回帰辺を導入することで状態変数の次元を固定とし、確率モデルとして定式化することを容易にした。また、非存在辺による経路の拡張、縮小を表現することで、実在の道路ネットワークに対応可能な定式化としたほか、STSP の制約を目的関数に極値を作ることで表現し、SA を適用可能とした。さらに、経路の変更に対する目的関数の増加量に基づき、棄却率の小さい提案分布を用い、SA を効率よく動作させる手法を提案した。評価実験において、通常的手法では最適解を得ることが困難な問題に対し、最適化が得られることを示した。また、実データを用いた評価実験においても、提案手法は先行研究と同程度の精度を高速に達成できることを示した。

ユーザの好みの推定方法や、スポットの人気度とユーザの個人的嗜好のバランス調整など、重みの適切な設定方法について検討することで、さらに有用な経路の発見が可能になると考える。また、提案分布に用いた探索数 K に関しては一定の値を評価実験に用いたが、温度に依存して変化するようにできれば、高温時には遠方の節点まで経路が拡張されやすいことを反映でき、また低温時には近い節点に対してのみ経路を最適化することで無駄が少なくなることが期待できる。温度履歴 $T(t)$ の設定方法について、今回扱った設定では区間線形関数を用いることで良好な結果を得られたが、最適な設定方法を検討することでさらなる性能向上が期待できる。Boltzmann 定数 k についても、負荷関数 f_c の分布に基づき決定することで性能向上が期待できると考える。

表 4.5 4.3 節で用いる記号の表

記号	定義	意味
F_{res}	$:\mathbb{R} \rightarrow \mathbb{R}$	対称化制約形状関数
F_s	$:\Omega'_A \rightarrow [0, \infty)$	罰則の総和
α_s	$:\in (-\infty, 1]$	対称化係数
α_g	$:\in (-\infty, 1]$	形状係数

4.3 非存在辺と実在辺の対称化による目的関数の改良

観光経路推薦の目的は、ユーザにより指定された滞在時間などの制約に基づき、訪問すべき観光スポットとそれらの間の経路を同時に決定し、ユーザに提案することである。訪問スポットの決定と、スポット間の経路の探索という、異なる 2 種類の問題を同時に扱う必要があるため、その定式化は複雑になる傾向がある。この問題に対し 4.2 節では、辺ベクトルを用いた観光経路推薦問題の定式化と SA を用いた解法を提案した。提案手法においてスポット・経路の情報は辺の重みとして統一的に扱われ、経路長に対して目的関数が極小値を持つように定式化することで TSP を拡張して観光経路問題を定式化した STSP を目的関数のみで表現可能としている。非存在辺の概念の導入により、任意の道路ネットワークにおける最適化に対し、SA を適用可能とすることで高速な解法を実現している。しかしながら、非存在辺と実在辺が目的関数において対等に扱われていないことにより、対称性が崩れる状態における解候補の近傍探索において SA の性能が低下するという問題点がある。この問題を解決するため、本節では経路問題としての制約を保持しつつ、非存在辺と実在辺に対して対称な目的関数の定式化を提案する。人工データセットと実データセットを用いた評価実験により、本定式化は有効に機能し、解の精度を向上させることを示す。

本節において、実在辺と非存在辺の対称性とは、辺ベクトルに含まれる非存在辺と対応する実在辺の交換における、目的関数の値の不変性を意味することとする。

4.2 節で定義される目的関数の値を最小化する経路は価値 f_d の総和が大きく、経路長が目標経路長 C_c に近い辺ベクトルである。非存在辺は最適化の過程で実在辺へ置換されるように目的関数が定式化されているが、この定式化では $F_c(\mathbf{e}) < \sigma^2$ となる $\mathbf{e} (\in \Omega'_A)$ の近傍探索において、SA の性能が低下すると考えられる。たとえば、辺ベクトル \mathbf{e} に、ある非存在辺 $e^v \notin \mathcal{E}$ を足した辺ベクトル \mathbf{e}^{+v} と対応する最短経路である $S(e^v) (\subset \mathcal{E})$ を足した経路 \mathbf{e}^{+s} について、 ϕ の増加量 $\Delta\phi^{+v} := \phi(\mathbf{e}^{+v}) - \phi(\mathbf{e})$ と $\Delta\phi^{+s} := \phi(\mathbf{e}^{+s}) - \phi(\mathbf{e})$ を考えたとき、 $C_{pa} \approx 1, C_{pb} \approx 0$ と設定する場合、 $F_c(\mathbf{e}) \geq \sigma^2$ において $\Delta\phi^{+v}$ と $\Delta\phi^{+s}$ はほぼ等しいが、 $F_c(\mathbf{e}) < \sigma^2$ においては $F_c(\mathbf{e})$ の値が小さくなるほど差異が大きくなる。このことは、非存在辺が実在辺へ置換されたとき、目的関数の値が大きく変わることを意味する。つまり、非存在辺を含む状態を経由するたびに探索の方向が適切なものからずれることになり、最適化の性能が低下すると推測される。

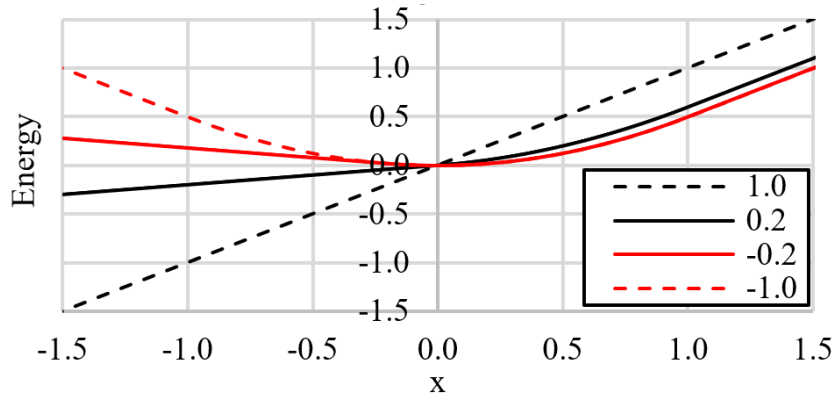


図 4.9 $\sigma^2 = 1, \alpha_g \in \{1, 0.2, -0.2, -1\}$ における $F_{\text{res}}(x)$ のグラフ.

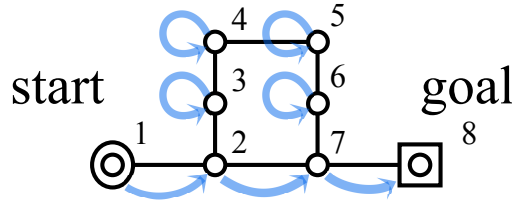


図 4.10 $\mathcal{N} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ とした道路ネットワークにおける辺ベクトルの例.

4.3.1 提案する定式化と対称性に関する考察

本節では、4.2 節で定式化した目的関数には対称性の点で問題があることを示す．またそれに対し、対称性の高い目的関数を提案し、高効率な SA の実行を可能とする方法を提案する．4.2 節の定式化では、経路長に対し STSP の制約を表現する関数が式 (4.16) 中の第 1, 2 項の複数項により表現されており、このままでは解析が難しいため、本節では制約を表現する関数を、式 (4.33) により定義する F_{res} のみとした式 (4.30) により目的関数 ϕ を定義する．

$$\phi(\mathbf{e}_\Lambda) = F_{\text{res}}(F_c(\mathbf{e}_\Lambda) + \alpha_s F_s(\mathbf{e}_\Lambda) - C_c) - \sum_{i \in \Lambda} f_d(\mathbf{e}_i) + (C_p - \alpha_s) F_s(\mathbf{e}_\Lambda) \quad (4.30)$$

$$F_s(\mathbf{e}_\Lambda) = \sum_{i \in \Lambda} f_s(\mathbf{e}_i) \quad (4.31)$$

$$f_s(e) = \begin{cases} f_c(e) - \sum_{s \in S(e)} f_c(s) & , S(e) \neq \emptyset \\ \infty & , S(e) = \emptyset \end{cases} \quad (4.32)$$

$$F_{\text{res}}(x) = \begin{cases} \alpha_g x + c_1 & , x < x_m \\ c_2 x^2 + c_3 x & , x_m \leq x \leq \sigma^2 \\ x + c_4 & , \sigma^2 < x \end{cases} \quad (4.33)$$

4.2 節で述べた手法では、非存在辺に罰則を与えるために f_p を用いていたが、提案する目的関数では式 (4.32) で定義するように、 f_c に対する追加の負荷としてこれを扱う．これにより、後述するように実在辺と非存在辺が探索において対称に扱われるようになる．しかし両者を完全に対称に扱った場合、非存在辺が最適化の過程で実在辺に置換されなくなり、実在辺上のスポットを探索

できなくなるため、僅かに対称性を崩す罰則項を $(C_p - \alpha_s) F_s(\mathbf{e}_A)$ として第 3 項目に定義した. C_p ($\in \mathbb{R}$) は罰則係数, α_s ($\in \mathbb{R}$) は対称化の度合いを決定する係数であり, 適切な設定方法については後述する.

本節最後の考察において, 4.2 節の定式化では対称性の崩れにより目標経路長よりも長い経路長を持つ経路が生成される傾向があると推測されることを示す. 4.3.2 節の評価実験において, この傾向は目的関数が単調増加でないことに起因するものでないことを検証するために, F_{res} は α_g ($\in \mathbb{R}$) の値により単調増加な関数から既存の定式化で用いた式 (4.16) のものまで表現できるように定義する. $x > 0$ において $F_{\text{res}}(x)$ を単調増加とするために, $F'_{\text{res}}(0) = \max(\alpha_g, 0)$ と定める. これにより経路長が目標経路長 C_c を超えたところで経路は拡張されにくくなる. また, $(\alpha_g, \alpha_s) = (-1, 0)$, $C_{\text{pb}} = 0, C_{\text{pa}} = C_p$ で式 (4.30) を式 (4.16) と一致させるために $c_1 = c_2 x_m^2 + c_3 x_m - \alpha_g x_m$, $c_2 = (1 - c_3)/(2\sigma^2)$, $c_3 = \max(\alpha_g, 0)$, $x_m = (\alpha_g - c_3)/(2c_2)$, $c_4 = c_2(\sigma^2)^2 + c_3\sigma^2 - \sigma^2$ と定める. 図 4.9 に様々な α_g の値における $F_{\text{res}}(x)$ のグラフを示す.

ある経路の状態に対応した辺ベクトルからの ϕ の変化量を $\Delta\phi$, F_s の変化量を ΔF_s , F_c の変化量を ΔF_c と表し, 経路の様々な遷移過程について, 非存在辺に対応する実在辺集合に置換したとき, およびその逆の置換を行ったときに ϕ が受ける影響を考察する. 式 (4.30) から, $\Delta\phi$ を式 (4.34) で近似する.

$$\Delta\phi = \xi\Delta F_c + \alpha_s\xi\Delta F_s + (C_p - \alpha_s)\Delta F_s \quad (4.34)$$

ここで, $\xi = F'_{\text{res}}(F_c(\mathbf{e}) + \alpha_s F_s(\mathbf{e}) - C_c)$, $\mathbf{e} \in \Omega'_A$ としている. また, 議論を簡単にするため, ここでは f_d について考慮しない. f_d は f_c よりも小さい値に設定するため影響は少ないと考えるが, 実際に f_d を含む問題であっても本議論により得られた提案手法が有効であることを 4.3.2 節の評価実験で示す. ΔF_s の変化量と ϕ の変化量の比 $\Delta\phi/\Delta F_s$ は式 (4.35) で与えられる. $\Delta F_s, \Delta F_c$ は $T \rightarrow 0$ において非存在辺が実在辺に置換される条件と, 式 (4.32) の関係式を満たす必要があり, 互いに独立でないため, 以降では $\Delta F_c/\Delta F_s$ としてまとめて扱う.

$$\frac{\Delta\phi}{\Delta F_s} = \frac{\Delta F_c}{\Delta F_s}\xi - \alpha_s(1 - \xi) + C_p \quad (4.35)$$

$\Delta F_c/\Delta F_s$ を見積もるために, ある非存在辺 e^v が対応する実在の最短経路 $S(e^v)$ に置換される場合を考える. このとき, ΔF_c と ΔF_s は式 (4.36), 式 (4.37) で表せる. 置換前の経路に含まれる実在辺と逆向きの辺が $S(e^v)$ に含まれる場合, 閉路が形成される. 4.2 節における状態変化の制約はこのような閉路を除外し, 式 (4.36) の第 2 項はこの処理に該当する. R は置換前の経路を構成する実在辺を, それぞれ向きを逆にした辺の集合である.

$$\Delta F_c = \sum_{s \in S(e^v)} f_c(s) - 2 \sum_{s \in S(e^v) \cap R} f_c(s) \quad (4.36)$$

$$\Delta F_s = - \sum_{s \in S(e^v)} f_c(s) \quad (4.37)$$

議論の簡単化のため, 以降閉路の除外, すなわち, 式 (4.36) の第 2 項を考慮しない. しかしながら, 4.3.2 節に示す評価実験において, 最適化中に閉路の除外が発生する場合でも本議論により導出される定式化は有効に機能することを示す. 第 2 項を無視した式 (4.36) と式 (4.37) から $\Delta F_c/\Delta F_s = -1$ となる.

非存在辺が実在辺に置換されるためには、 $\Delta F_s < 0 \rightarrow \Delta\phi < 0$ ，すなわち、 $\Delta\phi/\Delta F_s > 0$ を満たす必要があり、 $\Delta F_c/\Delta F_s = -1$ と式 (4.35) から式 (4.38) が導かれる、これは例えば図 4.10 において、(1, 2, 3, 5, 6, 7, 8) の経路が (1, 2, 3, 4, 5, 6, 7, 8) に移るときに対応する。

$$\frac{\Delta\phi}{\Delta F_s} = -\xi - \alpha_s(1 - \xi) + C_p > 0 \quad (4.38)$$

式 (4.33) から、 $\forall x \in \mathbb{R}, F'_{\text{res}}(x) = \xi \in [\alpha_g, 1]$ となり、この範囲における任意の ξ において式 (4.38) を満たす必要がある。まず、 $\xi = 1$ のとき式から α_s の項が消えるため、 $C_p > 1$ となる必要がある。式 (4.13) における C_{pa} と同様に、 C_p はできるだけ小さい値とする方がよいため、 $C_p \approx 1$ とすることを前提とする。このとき、 $\xi < 1$ の場合も式 (4.38) が成立するためには $1 \geq \xi + (1 - \xi)\alpha_s$ を満たす必要があり、 $1 \geq \alpha_s$ が導かれる。非存在辺が新たに生成される場合については、この逆の過程を考えれば $T \rightarrow 0$ において非存在辺が新たに生成されない条件として十分であり、 $\Delta F_c, \Delta F_s, \Delta\phi$ の符号がそれぞれ逆になるため同様の議論となる。

非存在辺に対応する実在辺で置換した場合に目的関数の値が変化しないとき、すなわち、式 (4.38) において $\Delta\phi/\Delta F_s = 0$ が成立するとき、両者の負荷に対し目的関数は対称であり、置換前後の状態を滑らかに移行できる。以下、本節で提案する定式化と 4.2 節の定式化における対称性について考察する。

本節で提案する定式化において、 $1 \geq \alpha_s$ を満たす最大値 $\alpha_s = 1$ で、 $\Delta\phi/\Delta F_s$ を最小にすることができる。このとき、 $\Delta\phi/\Delta F_s = -1 + C_p$ となり、 $C_p \approx 1$ と設定することを考えれば、 ξ の値によらず常に $\Delta\phi/\Delta F_s \approx 0$ となる。したがって、非存在辺と実在辺に対して目的関数は対称であり、滑らかな解の探索を期待できる。一方、4.2 節の定式化に対応する $\alpha_s = 0$ の場合は、 $\Delta\phi/\Delta F_s = C_p - \xi$ となり、 $\xi = 1$ 以外では、非存在辺と実在辺に対する目的関数の対称性が低く、探索は滑らかに行われない。また、Random Walk [24] と同様の性質により温度が高いときには経路長が長い状態を取り、そこから温度が下がるにつれ、非存在辺を含む状態と、対応する実在辺に置換された状態を交互に取りながら短い経路に遷移していくが、4.2 節の定式化では $|\Delta\phi/\Delta F_s|$ が大きいため低い温度において非存在辺を含む状態へ移行できず、経路は長い状態のまま硬直する可能性がある。その結果、最終的な経路長が長くなりやすいと考えられる。

対称性の議論からは外れるが、 α_g の設定次第で最適化性能の向上を望めるため、このパラメータの設定方法についても述べる。 α_g を大きくするほど、 $F_{\text{res}}(x)$ は単調な直線に近くなり目標経路長 C_c が制約として機能しづらくなるため、 α_g には小さな値を用いることが望ましい。また、 $\alpha_g \approx 0$ とすることで、 $\alpha_s = 1$ ならば $F_c(\mathbf{e}_A) + F_s(\mathbf{e}_A) < C_c$ である状態 \mathbf{e}_A について $\phi(\mathbf{e}_A)$ は $F_c(\mathbf{e}_A), F_s(\mathbf{e}_A)$ によらず一定となり、そのような経路間を滑らかに探索できるため、計算精度の向上が期待できる。また、 $\alpha_g < 0$ の設定では、経路長に対して目的関数は常に正の傾きを持つため、より短い経路長を持つ状態ほど目的関数の値を小さくすることができる。このことから、 α_g を 0 に近くかつ負の値に設定することで、無駄な経路を削減する計算精度の高い探索となることが期待できる。

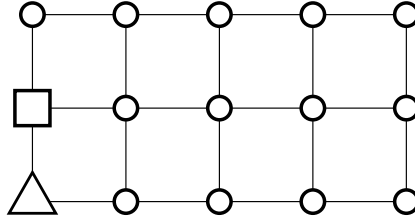


図 4.11 5×3 の節点からなる昇目構造グラフの人工データセット．四角は始点, 三角は終点をそれぞれ表す．

4.3.2 評価実験

4.3.1 節の考察で示したとおり, 既存手法では生成される辺ベクトルを \mathbf{e}_A^* としたとき, その経路の長さ $F_c(\mathbf{e}_A^*)$ が C_c よりも長くなると予想されることと, 提案する定式化を用いた場合にこれを解決できることを検証するため, 図 4.11 に示すような昇目構造を持つ人工データセットを用い実験を行った. 図において, 四角, 三角はそれぞれ始点, 終点を表す. 実験では 20×20 のサイズを用いたが, スペースの関係上図では 5×3 としている. また, $\forall e \in \mathcal{E}, f_c(e) = 1 \wedge f_d(e) = 0$ としている. $(\alpha_g, \alpha_s) \in \{(-1, 0), (-1, 1), (0.01, 0), (0.01, 1)\}$ における, $C_c = 100, \sigma^2 = 50$ とした場合の, 128 回の試行における $F_c(\mathbf{e}_A^*)$ の平均値と, 計算時間の平均値を表 4.6 に示す. $t \in [0, 1]$ を, $t = 0$ が計算開始, $t = 1$ が計算終了を意味する計算の進行度としたとき, 温度変化は $T(t) = 10^{-3t}$ により与え, t が 5×10^{-5} 進む毎に標本抽出を行い, $C_p = 1.005$, Boltzmann 定数には 4.2.6 節の指針に従い $k = 1.2$ を選び, 提案分布の探索範囲には $K = 16$ を用いた. 表 4.6 より, 4.2 節の定式化に対応する $(\alpha_g, \alpha_s) = (-1, 0)$ では 4.3.1 節での議論の通り, C_c の値より長い経路が生成される傾向にあることがわかる. $(\alpha_g, \alpha_s) = (0.01, 1)$ の場合でもこの傾向は変わらないことから, 目的関数を経路長に対して単調増加としただけではこの傾向は改善されないことがわかる. 一方, 4.3.1 節での議論に従い $\alpha_s = 1$ とした $(\alpha_g, \alpha_s) = (-1, 1)$ のときはほぼ期待される経路長が得られていることがわかる. また, $(\alpha_g, \alpha_s) = (0.01, 1)$ の実験結果では始点・終点を直接結ぶ長さ 1 の経路が生成される傾向にあることがわかる. 本実験では $\forall e \in \mathcal{E}, f_d(e) = 0$ としており, 経路長を増加させるメリットがないため妥当な結果といえる. これらの結果から, $\alpha_s = 1$ とする対称性の高い設定において, 目的関数の形状から期待される解を得られることがわかる. また, 生成される経路長が短いほど計算時間が減少しており, 無駄な経路を削減する α_g の設定と合わせれば, 提案する定式化は高速化にも貢献することがわかる.

4.2 節における評価実験と同様に, 文献 [27] で用いられているデータセットを用いて実験を行った. 評価指標には 4.2 節と同様, Recall を用いる. 既存手法および提案手法により生成された経路の Recall を表 4.7 に示す. PersQ は文献 [27] で提案された手法, $\text{EV}(\text{Edge Vector})(\alpha_g, \alpha_s)$ が提案する定式化である. $\text{EV}(-1, 0)$ が 4.2 節における定式化である式 (4.16) の目的関数に対応する. $C_p = 1.005$ を用い, 温度変化は $T(t) = 10^{-4t}$ に従うとし, t が 5×10^{-4} 進む毎に標本抽出を行った. その他の設定値については 4.2 節と同様である. このデータセットは f_d が 0 でない辺を含んでいるが, 表 4.7 から, 提案する定式化を用いることで性能が向上することを確認できる. 特にこのデータセットでは, α_g の調整が既存手法からの性能改善に貢献していることがわかる.

表 4.6 様々な (α_g, α_s) において提案する定式化を用い生成された辺ベクトル \mathbf{e}_A^* の経路の長さ $F_c(\mathbf{e}_A^*)$ と計算時間の平均値

(α_g, α_s)	平均経路長	平均計算時間 [s]
$(-1, 0)$	106.9	10.2
$(-1, 1)$	100.0	5.42
$(0.01, 0)$	106.7	9.96
$(0.01, 1)$	1.000	2.80

表 4.7 PersQ[27] と提案する定式化 ($EV(\alpha_g, \alpha_s)$) を用いた結果の Recall. $EV(-1, 0)$ が 4.2 節の定式化におけるものに対応する. C: Cali. Adv, H: Hollywood, D: Disneyland M: Magic King., E: Epcot

	C	H	D	M	E
PersQ	0.483	0.482	0.332	0.440	0.472
$EV(-1, 0)$	0.291	0.421	0.231	0.388	0.401
$EV(0.01, 0)$	0.406	0.461	0.405	0.450	0.530
$EV(0.01, 1)$	0.441	0.472	0.423	0.475	0.555

4.3.1 節の議論では f_d および閉路の除外操作を考慮の対象外としたが、人工データセット、文献 [27] のデータセットともに最適化中に閉路の除外が起こりうるものである。さらに後者のデータセットは f_d が 0 でない辺を含むが、どちらの場合でも提案する定式化が有効に機能していることがわかる。

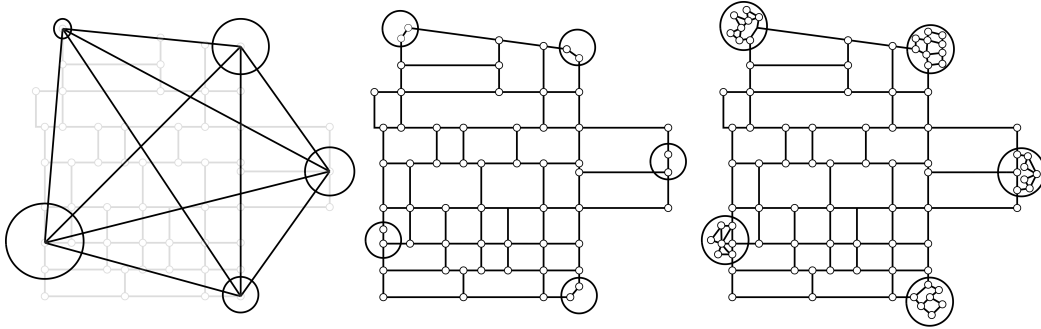


図 4.12 道路ネットワークに対するモデル化. (a): 従来手法 [9][26][27] で行われるモデル化. 節点の大きさの違いは, 割り当てられる滞在時間が異なる様子を表す. (b): 4.2 節におけるモデル化. 滞在時間はスポット内の辺に負荷として割り当てられる. (c): 本節で導入するモデル化.

4.4 冗長経路の導入によるスポット滞在時間の表現

4.4.1 はじめに

4.2 節では, 従来辺と節点に与えられていた重みを辺のみに統一した, 辺ベクトルによる観光経路推薦問題の定式化を行った. STSP の定式化を目的関数のみで表現することで SA を用いた最適化を適用可能とし, 提案分布を工夫することで高効率な探索を可能とした. 4.3 節では, 目的関数における実在辺と非存在辺の取り扱いを見直し, 対称性の概念に基づき目的関数の構築を行うことで, SA をより高効率に動作させることを可能とする定式化を提案した. 本節ではここまでの議論により得られた手法をさらに拡張し, スポットにおけるユーザの滞在時間を考慮した推薦手法を提案する. 前節までは, スポットに一つの辺を割り当て, その移動時間として滞在時間を表現していた. この方法ではスポットにおける滞在時間は固定的になり, 「興味のあるスポットは時間の許す限り長く滞在したい」などのような, 実際によく行われる柔軟な調整を考慮することができない. そこで本節で提案する手法ではスポット内に冗長な経路を含むグラフを配置し, 辺に与える価値に応じて生成される経路を決定する. これにより, 問題の定式化を変えることなく, グラフ構造の変更のみで, 可変な滞在時間を表現可能とする. 人工データセットを用いた評価実験により, 滞在時間の調整に関して提案手法が有効であることを示す.

4.4.2 提案手法

従来手法 [26][27] および 4.3 節までに本論文で対象としていた問題では, スポットにおける滞在時間はスポットに対応する辺あるいは節点を持つ固定された値として定式化される. 図 4.12 に, 道路ネットワークに対するこれらのモデル化の違いを示す.

図 4.12(a) は文献 [26][27] のモデル化に対応し, スポットに対応する節点への重みの一つとして滞在時間を表現する. 文献 [27] は節点を訪れる時刻に依存して滞在時間が変化する定式化を行っているが, 時刻に対して予め定義された滞在時間を用いることで時間帯による待ち時間の違いを考慮するものであり, 本節で対象とするような, 時間調整のための滞在時間変更には対応できない.

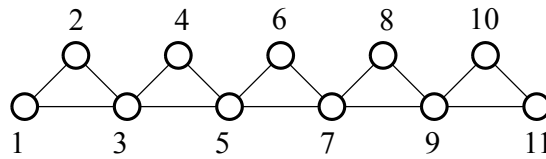


図 4.13 $\{1, 2, \dots, 11\} \subset \mathcal{N}$ であるグラフにおける, 可変な滞在時間を表現するためのスポット内冗長経路の例.

図 4.12(b) は 4.3 節までに提案した手法のものに対応する. スポットに対し辺を配置し, その負荷 (f_c) として滞在時間を表現している. 辺か節点かの違いはあるが, スポットに対する滞在時間が固定値となる点で同様である.

これらに対し, 経路全体を考慮して各スポットの滞在時間を柔軟に調節可能とするため, 本節では図 4.12(c) で示されるような, 冗長な経路を含むグラフをスポット内に配置するモデル化を提案する. スポットの滞在時間を, 「スポット内に存在する辺のうち, 経路として選択されたものが持つ負荷の総和」とみなすことで, 長い滞在時間はスポット内の冗長な経路をめぐる行動としてモデル化する. このとき, 人気のスポットや関心のあるスポット内にある辺の価値 (f_d) を高くすることで, スポット内の辺が選択されやすくなるのが可能である. これにより, 問題の定式化を変えることなく, グラフ構造の変更だけで可変な滞在時間が表現可能となる. 滞在時間を最適化中に決定することができれば, 行きたいスポットをすべて巡っても若干の余裕がある場合に, 好きなスポットの滞在時間を長くしたり, そのほかのスポットを短時間寄り道したり, あるいは土産物店やレストランで時間を潰すといったような柔軟な経路を生成可能となる. 本章で提案してきた解法は, 複雑なグラフ構造に対応し, 高速な解法となっていることから, 冗長な経路の導入により多少問題の規模が大きくなっても問題がないと考える.

スポット内のグラフに関して, スポット内にサブスポットがある場合はそれらをグラフとしてそのまま表現することができるが, 単に滞在時間を可変としたい場合は図 4.13 のようなグラフをスポット内に用意することができる. 図中の数字は節点番号を示す. 節点 2, 4, 6, 8, 10 は迂回経路を構成するものであり, こちらを通る場合の時間はスポットに滞在しているとはみなさない. このグラフにおいて, 例えば, $f_c((1, 3)) = 1$, $f_c((3, 5)) = 2$, $f_c((5, 7)) = 4$, $f_c((7, 9)) = 8$, $f_c((9, 11)) = 16$ とし, 迂回経路の辺の負荷をこれらに比べごく小さなものとすれば, 経路に応じて 0 から 31 の間の, 整数値の滞在時間を表現することが可能である. また, 最低所要時間が決まっている場合は, スポットを構成する辺に与える負荷の最低値を最低所要時間に設定すれば良い.

4.4.3 評価実験

提案手法の有効性を示すため, 2 種類の人工データセットを用いて評価実験を行う. 図 4.14 は 3 つのスポット, Spot 1, Spot 2, Spot 3 を含む人工データセット 1 であり, [] 内の数字は各辺の f_c の値を示す. 赤線, 黒破線, 赤破線の辺はそれぞれ Spot 1, Spot 2, Spot 3 内に存在するとし, それぞれの f_d は $e \in \mathcal{E}$ に対して以下のように設定する.

$$\text{Spot1} : f_d(e) = 0.8f_c(e)$$

$$\text{Spot2} : f_d(e) = 0.4f_c(e)$$

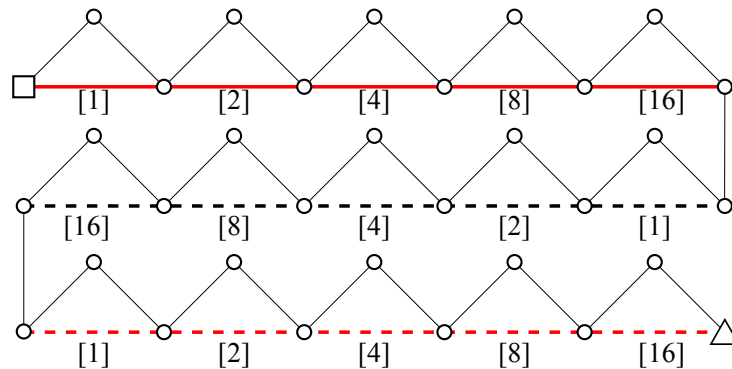


図 4.14 人工データセット 1. [] 内の数字は f_c の値を示す. 赤線, 黒破線, 赤破線の辺はそれぞれ Spot1, Spot 2, Spot 3 を構成する. 四角は始点, 三角は終点をそれぞれ表す.

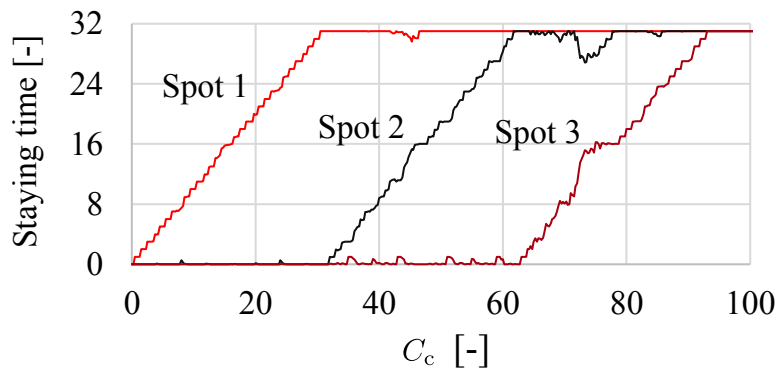


図 4.15 人工データセット 1 における, $C_c = \{0, 0.25, 0.5, \dots, 110\}$ において生成された経路における各スポットの滞在時間.

Spot3 : $f_d(e) = 0.2f_c(e)$

また, 黒実線で示されたそのほかの辺 e について $f_d(e) = 0, f_c(e) = 0.01$ としている. 四角は始点, 三角は終点をそれぞれ表す. この人工データセットに対して, 目標経路長 (旅行時間に対応) $C_c = 0.00, 0.25, 0.50, \dots, 110$ についてそれぞれ 32 回の試行を行い, 各スポットの滞在時間について平均値を求め比較する. パラメータの設定には $\alpha_s = 1, \alpha_g = 0.01, \sigma^2 = 0.1, k = 2, C_p = 1.005$ を用い, 温度変化は $T(t) = 10^{-3t}$ とした. t が 10^{-5} 進むごとにサンプリングを行い, 提案分布の探索範囲には $K = 16$ を用いた.

実験結果を図 4.15 に示す. 負荷に対する価値の割合が高いスポットから, 優先的に経路が割り当てられていることがわかる. すなわち, 価値の高いスポットの滞在時間を長くすることで, 目標旅行時間 (C_c) の余りを調節しつつ, さらに時間が余れば価値の低いスポットの滞在時間で調節していると解釈できる. この結果より, 提案手法は滞在時間を柔軟に調節可能であることがわかる. なお, C_c が 95 以上の場合では全スポットの滞在時間がほぼ等しくなっているのは, データセットの設定上, どのスポットにおける滞在時間もこれ以上増やすことができないためである.

図 4.16 に示す人工データ 2 を用いて行った評価結果を示す. このデータセットは図 4.14 の人工データセット 1 をスポット以外の経路も含むように拡張したものであり, スポット間の経路探索も

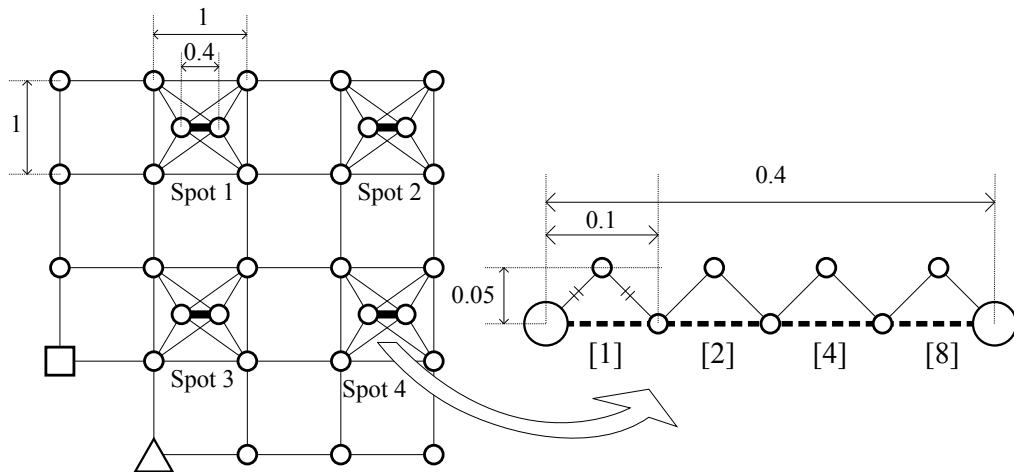


図 4.16 人工データセット 2. 4 つのスポットが配置され、太線で描かれた辺は右側グラフに対応する．黒破線下の [] 内の数字は各辺の f_c の値を示す．四角は始点、三角は終点をそれぞれ表す．

含む点で現実的な問題設定により近い．太線で描かれた辺は図の右側にあるスポットを構成するグラフに対応する．[] 内の数字はその辺の f_c の値である．このグラフには 4 つのスポット，Spot 1, Spot 2, Spot 3, Spot 4 があり，破線で描かれる辺 $e \in \mathcal{E}$ における f_d の値は以下のように設定する．

$$\text{Spot 1} : f_d(e) = 0.8f_c(e)$$

$$\text{Spot 2} : f_d(e) = 0.6f_c(e)$$

$$\text{Spot 3} : f_d(e) = 0.4f_c(e)$$

$$\text{Spot 4} : f_d(e) = 0.2f_c(e)$$

また、これら以外のすべての辺 e において、 $f_d(e) = 0$ とし、 $f_c(e)$ は幾何学距離に等しいものとする．幾何学的情報は図 4.16 中に記載してある．四角は始点、三角は終点をそれぞれ示す．パラメータの設定方法と実験方法については人工データセット 1 に対して用いたものと同様である．

実験結果を図 4.17 に示す．Move はスポット間の移動時間 [-] を示す．Spot 3 が終始点に近い場所にあるため、目標経路長 (旅行時間) C_c が小さい場合にこのスポットが優先されているが、長くなるにつれ、図 4.15 の結果と同様の傾向を示している．この結果から、冗長な経路による滞在時間表現は、スポット間の経路探索が必要な問題に対しても有効であることがわかる．また、スポット間の移動時間は $C_c = 10, 30$ 付近など、スポットが経路に追加されるときにのみ増加しており、このことから提案手法は不必要に経路を拡張しないこともわかる．

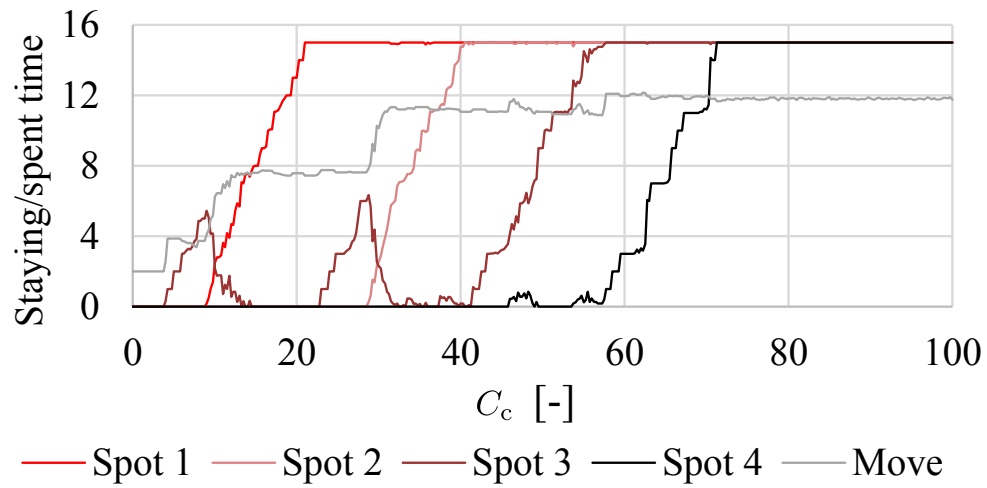


図 4.17 人工データセット 2 における, $C_c = \{0, 0.25, 0.5, \dots, 100\}$ において生成された経路に含まれる各スポットの滞在時間とスポット間の移動時間の総和.

第5章

終わりに

本論文では、エネルギー関数の形状に着目した焼きなまし法の適用法について提案し、Boltzmann Machine (BM)、観光経路推薦問題を対象としてその有効性を示した。

2章の関連研究においては、数学的解析を行うために新たな表記法を導入し、これにより確率分布の表記を従来用いられていた表記法に比べ厳密かつ拡張性の高いものとした。この表記法は3章以降の記述に用いている。

3章では、BMを用いた評価値行列から得られる潜在因子の特徴を調査した結果に基づき、事後分布推定におけるBMの特性について考察した。その後、Deep Boltzmann Machineの定式化を温度分布を設定可能なものに変更し、Floating Zone (FZ)法の効果を狙った焼きなまし法(SA)の適用を行い、高精度な事後分布の推定が可能であることを示した。また、数学的解析により、温度勾配に従い状態変数間を伝わる情報の流れが制限されることを示し、このことが、提案手法が高精度な事後分布の推定を可能とした理由であることを説明した。

4章においては、SAの適用が従来行われていなかった観光経路推薦問題に対して、全ての要素を辺のみを用い扱い、自己回帰辺を導入することで状態変数の次元を経路の拡張を伴う場合でも固定とする定式化を提案した。これにより確率分布より経路が生成されるとモデル化し、観光経路推薦問題における総移動時間の制約を連続な関数を用いて近似することでSAを適用可能とする定式化を提案した。提案手法はスポット間の経路も最適化の対象とするため、問題の規模が従来手法の対象問題よりも大きくなるが、状態の変化による目的関数の値の増減を近似的に見積もることで、棄却率の低い提案分布を構築し、SAを高い効率で動作させることを可能とした。人工データにおける実験により、通常は無作為な状態変化を起こす提案分布では最適解を得ることができなかった問題に対して、提案手法は最適解を得ることが可能なことを示した。また実データを用いた評価実験では、先行研究と同等の再現率(Recall)を持つ解を高速に生成可能なことを示した。

本論文で得られた成果は、SAを適用するにあたりエネルギー関数の形状を考慮することの重要性を示したものであり、対象としたBMや、観光経路推薦問題だけでなく、より多様な用途における焼きなまし法の適用可能性の拡大へ貢献するものである。今後、様々な問題に対する適切な適用法を調査していくことで、一般的な構築法に関する議論に貢献すると考える。

最後に、SAについての展望を、情報科学における究極の目標は人工知能を完成させることだという前提のもと述べさせていただきたい。人工知能を構築すべく、ヒトの知能を模擬する研究は広く行われ、中でも、ヒトの神経細胞のネットワークを模擬した手法は多くの課題へ適用され、注目を

集めている。しかしながら、ヒトの知能はネットワークの構造によってのみ成り立っているわけではなく、情報の伝達、細胞の活性度を制御する様々なホルモンとタンパク質の相互作用も重要な役割を果たしている [78]。ヒトが情報を扱っていることに疑いはなく、また、Shannon が示した確率分布による情報量の定式化 [8] から、ヒトの脳を確率モデルにより記述することは妥当だと考えられる。このとき、本論文で着目した SA における温度の役割は、ホルモンが神経細胞を制御する働きモデル化に貢献できる可能性があると考え。このような考察から、SA はただの最適化手法にとどまらず、情報の理論を知識表現の領域にまで拡張する可能性を秘めており、今後重要な役割を果たすと考える。

付録 A

Einstein の縮約記法

添字記法において、 D 次元ベクトル変数 \mathbf{x} ($\in \mathbb{R}^D$) と D 次元正方行列 A ($\in \mathbb{R}^{D \times D}$) の積 \mathbf{y} ($\in \mathbb{R}^D$) は式 (A.1) として表現される。

$$y_i = \sum_{j=1}^D A_{ij} x_j, i = 1, 2, \dots, D \quad (\text{A.1})$$

特に i の範囲について誤解が生まれない場合、 $i = 1, 2, \dots, D$ の表記は省略される。Einstein はこの記法において、同じ添字が2つ現れる積の項はそれらの総和を取る、と定めることで利便性を欠くことなく表記を簡潔にできる手法を用いた。これを縮約記法 (Einstein Summation Convention) と呼ぶ。添字記法と縮約記法におけるテンソルの扱いの例をいくつか示す。

以下はベクトルの内積である。

$$y_i y_i = \sum_{i=1}^D y_i y_i \quad (\text{A.2})$$

行列とベクトルの積は次のように表される。

$$y_i = \sum_{j=1}^D A_{ij} x_j = A_{ij} x_j \quad (\text{A.3})$$

式 (A.3) において、 i のような式がいくつも存在することを意味する添字を自由添字 (free index)、 j のように総和を意味するために用いられるものを偽添字 (dummy index) と呼ぶ。

転置行列は添字の順番を入れ替えることで表現される。

$$A^T \mathbf{x} = A_{ji} x_j \quad (\text{A.4})$$

逆行列と元の行列の積は次のように表現される。 δ_{ij} は Kronecker Delta であり、 $(i = j \Rightarrow \delta_{ij} = 1) \wedge (i \neq j \Rightarrow \delta_{ij} = 0)$ を満たす。

$$A^{-1} A = A_{ij}^{-1} A_{ij} = \delta_{ij} \quad (\text{A.5})$$

ここで、Kronecker Delta は式 (A.6) に示すように添字を入れ替える効果がある。

$$\delta_{ij} A_{jk} = A_{ik} \quad (\text{A.6})$$

この記法は統計学においても有用で、例えば共分散行列 A 、平均 $\boldsymbol{\mu}$ を持つ D 次元正規分布は式 (A.7) のように書ける。ここでは 2.1.6 節で提案する確率空間に対する添字記法は用いていない。

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |A|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x_i - \mu_i) A_{ij}^{-1} (x_j - \mu_j)\right) \quad (\text{A.7})$$

式 (A.8) 中の共分散行列 B_{ij} を縮約記法を用いて求める例を示す。添字が $\exp()$ 中で用いられる dummy index と重複するのでこちらには k, l を用いている。ここで、 $d\boldsymbol{x} = dx_1 dx_2 \cdots dx_D$ とする。

$$B_{ij} = \int_{\mathbb{R}^D} x_i x_j \frac{1}{(2\pi)^{\frac{D}{2}} |A|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x_l - \mu_l) A_{lk}^{-1} (x_k - \mu_k)\right) d\boldsymbol{x} \quad (\text{A.8})$$

A が共分散行列となると、それは正定値対称行列となるため、 A を式 (A.9) と表すことができる。

$$A_{ij} = P_{mi} \lambda_{mk} \lambda_{kl} P_{lj} \quad (\text{A.9})$$

λ_{ij} は A の固有値の平方根を対角成分にもつ対角行列、 P_{ij} は A の固有ベクトルからなる直交行列であり、 $P_{im} P_{jm} = \delta_{ij}$ を満たす。また、 $|P| = 1$ である。

これらを用いて、式 (A.10) の変換を用いれば、式 (A.8) を式 (A.11) と表すことができる。

$$y_i = \lambda_{ij}^{-1} P_{jk} (x_k - \mu_k) \quad (\text{A.10})$$

$$B_{ij} = \int_{\mathbb{R}^D} (x_i - \mu_i)(x_j - \mu_j) \frac{1}{(2\pi)^{\frac{D}{2}} \lambda_{11} \lambda_{22} \cdots \lambda_{DD}} \exp\left(-\frac{1}{2} y_k y_k\right) d\boldsymbol{x} \quad (\text{A.11})$$

縮約記法を用いれば以下のように式を変形し、 x_i を y_i で表すことができる。まず、式 (A.10) の両辺に λ_{li} を乗じる。この操作により、 l が free index、 i が dummy index となる。

$$\begin{aligned} \lambda_{li} y_i &= \lambda_{li} \lambda_{ij}^{-1} P_{jk} (x_k - \mu_k) \\ &= \delta_{lj} P_{jk} (x_k - \mu_k) \\ &= P_{lk} (x_k - \mu_k) \end{aligned}$$

同様に、両辺に P_{lm} を乗じれば式 (A.12) が得られる。

$$\begin{aligned} P_{lm} \lambda_{li} y_i &= P_{lm} P_{lk} (x_k - \mu_k) \\ &= \delta_{mk} (x_k - \mu_k) \end{aligned}$$

$$x_i - \mu_i = P_{li} \lambda_{lm} y_m \quad (\text{A.12})$$

式 (A.12) から微小要素は式 (A.13) の関係を持つ。

$$\delta_{im} dx_m = P_{li} \lambda_{lm} dy_m \quad (\text{A.13})$$

積分に現れる微小体積要素 (Jacobian) は基底ベクトルが描く微小体積に等しく、また、行列式は列ベクトルの描く体積に等しいため [33] 式 (A.14) と書ける。ただし、 $E_{ij}^x = \delta_{ij}$ 、 $E_{ij}^y = P_{li} \lambda_{lj}$ である。

$$|E^x| dx_1 dx_2 \cdots dx_D = |E^y| dy_1 dy_2 \cdots dy_D \quad (\text{A.14})$$

E_{ij}^y は式 (A.15) と求めることができる.

$$\begin{aligned} |E^y| &= |\lambda||P| \\ |E^y| &= \lambda_{11}\lambda_{22}\cdots\lambda_{DD}|P| \\ &= \lambda_{11}\lambda_{22}\cdots\lambda_{DD} \end{aligned} \quad (\text{A.15})$$

この結果を用いて, 式 (A.11) を式 (A.16) と変形できる. ただし, $d\mathbf{y} = dy_1 dy_2 \cdots dy_D$ としている.

$$B_{ij} = P_{ni}\lambda_{nh}P_{lj}\lambda_{lm} \int_{\mathbb{R}^D} y_h y_m \frac{1}{(2\pi)^{\frac{D}{2}}} \exp\left(-\frac{1}{2}y_k y_k\right) d\mathbf{y} \quad (\text{A.16})$$

添字表記では全てをスカラーとして扱い, 交換法則が成り立つことから式 (A.16) のように自由度の高い変形ができる. 一方で, 演算規則は添字の指定方法で定められているため, 厳密性は失われない.

ここで, Gauss 積分について式 (A.17) が成り立つ.

$$\int_{\mathbb{R}^D} y_h y_m \frac{1}{(2\pi)^{\frac{D}{2}}} \exp\left(-\frac{1}{2}y_k y_k\right) d\mathbf{y} = \delta_{hm} \quad (\text{A.17})$$

式 (A.17) を用いれば, 式 (A.16) を式 (A.18) と変形でき, これは式 (A.9) の定義と等しいため, 共分散行列 B は A に等しいことを確かめられる.

$$\begin{aligned} B_{ij} &= P_{ni}\lambda_{nh}P_{lj}\lambda_{lm}\delta_{hm} \\ &= P_{ni}\lambda_{nm}P_{lj}\lambda_{lm} \end{aligned}$$

$$B_{ij} = P_{ni}\lambda_{nm}\lambda_{ml}P_{lj} \quad (\text{A.18})$$

次元が固定的であるモデルに対しては縮約記法を用いることができ, より大規模なモデルになれば添字による表記を用いた解析は必須となる. また, 添字による表記法は線形代数で用いられる太字の表記と違い, そのままプログラムに実装することが可能であるため, 計算機科学, 情報学の分野において有用性が高いと考えられる.

付録 B

諸式の導出

B.1 Metropolis 基準

Metropolis 基準と対称な提案分布を用いた遷移確率分布 $p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda})$ と $p(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda})$ は式 (B.1), 式 (B.2) でそれぞれ表される.

$$p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) = \frac{1}{Z_p(\mathbf{y}_{\Lambda})} A_M(\mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda}) q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) \quad (\text{B.1})$$

$$p(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}) = \frac{1}{Z_p(\mathbf{x}_{\Lambda})} A_M(\mathbf{y}_{\Lambda}, \mathbf{x}_{\Lambda}) q(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}) \quad (\text{B.2})$$

上記式から, 式 (B.1), 式 (B.2) は詳細釣り合い条件を満たすことがわかる. また, 式 (B.3) の結果から周辺分布 $p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda})$ は元の分布 $p(\mathbf{x}_{\Lambda})$ に従うことがわかる.

$$\begin{aligned} p(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda}) &= \sum_{\mathbf{y}_{\Lambda}} \frac{1}{Z_p(\mathbf{y}_{\Lambda})} A_M(\mathbf{x}_{\Lambda}, \mathbf{y}_{\Lambda}) q(\mathbf{x}_{\Lambda(n)} = \mathbf{x}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{y}_{\Lambda}) p(\mathbf{y}_{\Lambda}) \\ &= p(\mathbf{x}_{\Lambda}) \sum_{\mathbf{y}_{\Lambda}} \frac{1}{Z_p(\mathbf{x}_{\Lambda})} A_M(\mathbf{y}_{\Lambda}, \mathbf{x}_{\Lambda}) q(\mathbf{x}_{\Lambda(n)} = \mathbf{y}_{\Lambda} | \mathbf{x}_{\Lambda(n-1)} = \mathbf{x}_{\Lambda}) \\ &= p(\mathbf{x}_{\Lambda}) \end{aligned} \quad (\text{B.3})$$

B.2 Boltzmann 分布と経験分布の相対情報量の勾配

相対情報量 $l(\theta)$ の $u (\in \theta)$ に関する勾配は次のように求まる. θ はパラメータの集合である.

$$\begin{aligned}
\frac{\partial l}{\partial u} &= - \sum_{\mathbf{x}_M} q(\mathbf{x}_M) \frac{\partial}{\partial u} \ln \left(\sum_{\mathbf{x}_{A \setminus M}} p(\mathbf{x}_A) \right) \\
&= - \sum_{\mathbf{x}_M} q(\mathbf{x}_M) \frac{1}{\sum_{\mathbf{x}_{A \setminus M}} p(\mathbf{x}_A)} \frac{\partial}{\partial u} \sum_{\mathbf{x}_{A \setminus M}} \frac{\exp(-\phi(\mathbf{x}_A))}{\sum_{\mathbf{y}_A} \exp(-\phi(\mathbf{y}_A))} \\
&= - \sum_{\mathbf{x}_M} q(\mathbf{x}_M) \frac{1}{p(\mathbf{x}_M)} \sum_{\mathbf{x}_{A \setminus M}} \left(- \frac{\partial \phi(\mathbf{x}_A)}{\partial u} p(\mathbf{x}_A) + p(\mathbf{x}_A) \sum_{\mathbf{y}_A} \frac{\partial \phi(\mathbf{y}_A)}{\partial u} p(\mathbf{y}_A) \right) \\
&= \sum_{\mathbf{x}_M} q(\mathbf{x}_M) \sum_{\mathbf{x}_{A \setminus M}} \frac{\partial \phi(\mathbf{x}_A)}{\partial u} p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) - \sum_{\mathbf{y}_A} \frac{\partial \phi(\mathbf{y}_A)}{\partial u} p(\mathbf{y}_A) \\
&= \sum_{\mathbf{x}_A} \left(\frac{\partial \phi(\mathbf{x}_A)}{\partial u} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) - \frac{\partial \phi(\mathbf{x}_A)}{\partial u} p(\mathbf{x}_A) \right) \tag{B.4}
\end{aligned}$$

式 (B.4) は式 (2.56) に等しい.

EM 法では, パラメータ更新を行うとき, 事後分布 $p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M)$ は定数であるとみなす. よって, 式 (2.55) を u で微分する手続きは以下ようになる.

$$\begin{aligned}
\frac{\partial l}{\partial u} &= - \sum_{\mathbf{x}_A} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) \frac{\partial}{\partial u} \ln p(\mathbf{x}_A) \\
&= - \sum_{\mathbf{x}_A} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) \frac{\partial}{\partial u} \left\{ -\phi(\mathbf{x}_A) - \ln \left[\sum_{\mathbf{y}_A} \exp(-\phi(\mathbf{y}_A)) \right] \right\} \\
&= - \sum_{\mathbf{x}_A} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) \left\{ - \frac{\partial \phi(\mathbf{x}_A)}{\partial u} + \sum_{\mathbf{y}_A} \frac{\partial \phi(\mathbf{y}_A)}{\partial u} p(\mathbf{y}_A) \right\} \\
&= \sum_{\mathbf{x}_A} \left(\frac{\partial \phi(\mathbf{x}_A)}{\partial u} q(\mathbf{x}_M) p(\mathbf{x}_{A \setminus M} | \mathbf{x}_M) - \frac{\partial \phi(\mathbf{x}_A)}{\partial u} p(\mathbf{x}_A) \right) \tag{B.5}
\end{aligned}$$

B.3 温度分布を持つ Boltzmann Machine の条件付き確率

式 (3.23) の導出は以下の手順に従う。

$\mathcal{V}_0 = \mathcal{V}_{K+1} = \emptyset, T_0 = 1, T_{K+1} = 1$ の仮定のもと、式 (3.21) における指数関数の引数となっている項に式 (3.22) を代入した結果は、式 (B.6) に等しい。

$$-\sum_{l=1}^K \frac{\phi_l(\mathbf{z}_\Lambda)}{T_l} = \sum_{l=1}^K \left\{ \frac{1}{T_l} \sum_{h \in \mathcal{V}_l} \mathbf{z}_h b_{d(h)} + \frac{1}{2T_l} \sum_{h \in \mathcal{V}_l} \sum_{j \in \mathcal{V}_{l-1}} \mathbf{z}_h w_{c(h,j)} \mathbf{z}_j + \frac{1}{2T_{l-1}} \sum_{h \in \mathcal{V}_{l-1}} \sum_{j \in \mathcal{V}_l} \mathbf{z}_h w_{c(h,j)} \mathbf{z}_j \right\}, \quad (\text{B.6})$$

2.3.4 節で説明する、 c に関する対称性の条件を用い、式 (B.6) を式 (B.7) と書くことができる。

$$-\sum_{l=1}^K \frac{\phi_l(\mathbf{z}_\Lambda)}{T_l} = \sum_{l=1}^K \left\{ \frac{1}{T_l} \sum_{h \in \mathcal{V}_l} \mathbf{z}_h b_{d(h)} + \frac{1}{T_{l-\frac{1}{2}}} \sum_{h \in \mathcal{V}_{l-1}} \sum_{j \in \mathcal{V}_l} \mathbf{z}_h w_{c(h,j)} \mathbf{z}_j \right\}. \quad (\text{B.7})$$

確率の乗法定理式 (2.8) より、条件付き確率 $p(\mathbf{x}_i | \mathbf{x}_{\Lambda \setminus \{i\}})$ は式 (B.8) と表せる。

$$p(\mathbf{x}_i | \mathbf{x}_{\Lambda \setminus \{i\}}) = \frac{p(\mathbf{x}_\Lambda)}{\sum_{\mathbf{y}_i} p(\mathbf{y}_i, \mathbf{x}_{\Lambda \setminus \{i\}})}. \quad (\text{B.8})$$

式 (B.7) と式 (3.21) を式 (B.8) に代入し、分母分子で共通の因数を約分することで以下の式を得ることができる。

$$p(\mathbf{x}_i | \mathbf{x}_{\Lambda \setminus \{i\}}) = \frac{\exp(A(\mathbf{x}_i, \mathbf{x}_{\Lambda \setminus \{i\}}))}{\sum_{\mathbf{y}_i} \exp(A(\mathbf{y}_i, \mathbf{x}_{\Lambda \setminus \{i\}}))}, \quad (\text{B.9})$$

A は式 (B.10) で定義される。ここで、3.2.3 節でも説明している通り、 $i \in \mathcal{V}_k$ であることに注意されたい。

$$A(\mathbf{z}_i, \mathbf{z}_{\Lambda \setminus \{i\}}) = \frac{\mathbf{z}_i b_{d(i)}}{T_k} + \sum_{j \in \mathcal{V}_{k-1}} \frac{\mathbf{z}_i w_{c(i,j)} \mathbf{z}_j}{T_{k-\frac{1}{2}}} + \sum_{j \in \mathcal{V}_{k+1}} \frac{\mathbf{z}_i w_{c(i,j)} \mathbf{z}_j}{T_{k+\frac{1}{2}}}. \quad (\text{B.10})$$

\mathbf{y}_i の属する状態空間 $\{0, 1\}$ と式 (B.9) において $A(0, \mathbf{x}_{\Lambda \setminus \{i\}}) = 0$ であることを考慮し、 $\mathbf{x}_i = 1$ となる確率は式 (B.11) により与えられ、これは式 (3.23) に等しい。また、 $\forall k \in \{1, \dots, K\}, T_k = 1$ とすれば、式 (2.31) に一致する。

$$p(\mathbf{x}_i = 1 | \mathbf{x}_{\Lambda \setminus \{i\}}) = f_\sigma(A(1, \mathbf{x}_{\Lambda \setminus \{i\}})). \quad (\text{B.11})$$

B.4 逆写像の共通部分

ある集合 A から B への写像 $f: A \rightarrow B$ があるとき、式 (B.12) が成り立つことを証明する。

$$\forall b, b' \in B, b \neq b' \Rightarrow f^{-1}(b) \cap f^{-1}(b') = \emptyset \quad (\text{B.12})$$

ここで、集合 U に対し、ある命題 P_1, P_2 により定義される集合 $X_1 := \{x \in U | P_1(x)\}$, $X_2 := \{x \in U | P_2(x)\}$ があるとき、 $\forall x \in U, x \in X_1 \cap X_2 \Leftrightarrow P_1(x) \wedge P_2(x)$ であり、また、 $X_1 = \emptyset \Leftrightarrow \neg(\exists x \in U, x \in X_1)$ である。これらから、 $X_1 \cap X_2 = \emptyset \Leftrightarrow \neg(\exists x \in U, P_1(x) \wedge P_2(x))$ である。この性質と、2.1.3 節で説明する逆写像の定義を用い、式 (B.12) を次のように変形できる。

$$\begin{aligned} \forall b, b' \in B, b \neq b' \Rightarrow f^{-1}(b) \cap f^{-1}(b') = \emptyset \\ \Leftrightarrow \forall b, b' \in B, b \neq b' \Rightarrow \neg(\exists a \in A, f(a) = b \wedge f(a) = b') \end{aligned} \quad (\text{B.13})$$

式 (B.13) が成り立つことは自明である。よって、式 (B.12) は成り立つ。

付録 C

Newton-Raphson 法と勾配降下法

Newton 法とは D 次元変数 $\mathbf{x} = (x_i \in \mathbb{R} | i = 1, 2, \dots, D)$ の上に定義される多次元目的関数 $\mathbf{f}(\mathbf{x}) = (f_i(\mathbf{x}) \in \mathbb{R} | i = 1, 2, \dots, D)$ が満たす, 式 (C.1) で与えられる連立方程式を数値的に解く手法である. また, 本手法の派生系として勾配降下法がある.

$$f_i = 0, i = 1, 2, \dots, D \quad (\text{C.1})$$

以下, 特に曖昧さがなければ $i = 1, 2, \dots, D$ の表記を省略する. また, Einstein の縮約記法を用いる (付録 A). 一次元変数 x とその上に定義される目的関数からなる $f(x)$ 方程式 $f(x) = 0$ について Newton 法を紹介する. 式 (C.2) は $\tau \rightarrow \infty$ の極限において 0 に収束する.

$$\frac{\partial f}{\partial \tau} = -f \quad (\text{C.2})$$

式 (C.2) を Euler 陽解法により解く方程式は式 (C.3) で与えられる.

$$\frac{f(\tau + \Delta\tau) - f(\tau)}{\Delta\tau} = -f(\tau) \quad (\text{C.3})$$

また, τ を $\Delta\tau$ で割った余りが 0 であるとき, 式 (C.4) と変形できることから, 式 (C.3) が $\tau \rightarrow \infty$ において収束するための必要条件は $0 < \Delta\tau < 2$ となる.

$$f(\tau) = f(0)(1 - \Delta\tau)^{\frac{\tau}{\Delta\tau}} \quad (\text{C.4})$$

ここで, 微分の連鎖規則を用いて式 (C.2) に変数 x を関連付ければ, 式 (C.5) で与えられる 1 次元変数に対する Newton-Raphson 法の更新式が得られる.

$$x(\tau + \Delta\tau) = x(\tau) - \Delta\tau \left(\frac{\partial f}{\partial x} \right)^{-1} f(\tau) \quad (\text{C.5})$$

D 次元連立方程式においても同様の考えで導出することができる. 式 (C.6) で与えられる連立微分方程式を考える.

$$\frac{df_i}{d\tau} = -f_i \quad (\text{C.6})$$

多次元変数についての連鎖規則は微小変化 df_i に対して式 (C.7) により全微分で与えられる.

$$df_i = \frac{\partial f_i}{\partial x_j} dx_j \quad (\text{C.7})$$

式 (C.7) を式 (C.6) へ代入し, Jacobi 行列 $\frac{\partial f_i}{\partial x_j}$ を J とおけば式 (C.8) が得られる.

$$\frac{dx_i}{d\tau} = - (J^{-1})_{ij} f_j \quad (\text{C.8})$$

式 (C.8) を Euler 陽解法で近似した式 (C.9) を用いて解を求める手法が D 次元連立方程式における Newton-Raphson 法である.

$$x_i(\tau + \Delta\tau) = x_i(\tau) - \Delta\tau (J^{-1})_{ij} f_j(\mathbf{x}(\tau)) \quad (\text{C.9})$$

また, J の代わりに, 対角成分以外を 0 と近似した式 (C.10) で与えられる R_{ij} を用いて更新式を式 (C.11) とした場合でも, J が対角優位行列であるならば解を得ることができる [20].

$$R_{ij} = \frac{\partial f_i}{\partial x_j} \delta(i = j) \quad (\text{C.10})$$

$$x_i(\tau + \Delta\tau) = x_i(\tau) - \Delta\tau (F)_{ij}^{-1} f_j(\mathbf{x}(\tau)) \quad (\text{C.11})$$

勾配降下法は D 次元変数の上に定義されたスカラーポテンシャル $E(\mathbf{x})$ を最小化する手法であり, \mathbf{x} が开区間に定義されているならば, $E(\mathbf{x})$ の最小値は極小値となる場所にある. Newton-Raphson 法の目的関数として式 (C.12) で与えられるものを用いる.

$$f_i = \frac{\partial E}{\partial x_i} \quad (\text{C.12})$$

このとき, $J_{ij} = \partial^2 E / (\partial x_j \partial x_i)$ となり, これは Hessian 行列とも呼ばれる. ここで, 勾配降下法では極値ではなく, より低いポテンシャルの値を求めることが目的であるため, \mathbf{x} は常に勾配の負の方向に修正されることが望ましい. そのような修正式は式 (C.11) を変形した式 (C.13) により与えられる.

$$x_i(\tau + \Delta\tau) = x_i(\tau) - \Delta\tau F_{ij}^{-1} \frac{\partial E}{\partial x_j} \quad (\text{C.13})$$

F_{ij} は式 (C.14) により定義される.

$$F_{ij} = \left| \frac{\partial^2 E}{\partial x_j \partial x_i} \right| \delta(i = j) \quad (\text{C.14})$$

機械学習の領域では, Hessian 行列と $\Delta\tau$ をまとめて学習率 ϵ とした式 (C.15) を通常勾配降下法と呼ぶ.

$$x_i(\tau + \Delta\tau) = x_i(\tau) - \epsilon \frac{\partial E}{\partial x_i} \quad (\text{C.15})$$

参考文献

- [1] The International Organization for Standardization, ISO 80000-2:2009(E) 2009, <https://people.engr.ncsu.edu/jwilson/files/mathsigns.pdf>.
- [2] 織田博行, 国際単位系 (SI) とその適用, 日本航海学会誌 NAVIGATION, Vol. 143, pp. 103-119, 2000, https://doi.org/10.18949/jinnavi.143.0_103.
- [3] IEEE Computer Society, “754-2008 - IEEE Standard for Floating-Point Arithmetic,” 2008, <https://doi.org/10.1109/IEEESTD.2008.4610935>.
- [4] C. Lin, Y. He, “Joint Sentiment/Topic Model for Sentiment Analysis,” Proceedings of 18th ACM Conference on Information and Knowledge Management, pp. 375-384, 2009, <https://doi.10.1145/1645953.1646003>
- [5] D. E. Smith, “History of Mathematics,” Dover Publications, Inc., Vol. 2, p. 692, 1958.
- [6] K. E. Iverson, “A Programming Language, p11,” New York: Wiley, 1962.
- [7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, “Equation of State Calculations by Fast Computing Machines,” The Journal of Chemical Physics, Vol. 21, No. 6, pp. 1087-1092, 1953, <https://doi.org/10.1063/1.1699114>.
- [8] C. E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, 1948, <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [9] G. Gutin, A. Punnen, “The Traveling Salesman Problem and Its Variations,” Springer Science & Business Media, 2006.
- [10] 松田善臣, 名嘉村盛和, 姜 東植, 宮城 隼夫, 最適観光経路問題とその解法, 電気学会論文誌 C, Vol. 124, No. 7, pp. 1507-1514, 2004, <https://doi.org/10.1541/ieejieiss.124.1507>.
- [11] D. Feillet, P. Dejax, M. Gendreau, “Traveling Salesman Problems with Profits: An Overview,” Journal of Transportation Science, Vol. 39, Issue 2, pp. 188-205, 2005, <https://doi.org/10.1287/trsc.1030.0079>.
- [12] 麻生英樹, 安田宗樹, 前田新一, 岡野原大輔, 岡谷貴之, 久保陽太郎, Danushka Bollegala, 神 尾敏弘, 松尾豊, 松原仁, 深層学習, 2 章:深層ボルツマンマシン, 近代科学社, 第 5 版, 2015.
- [13] 岡谷貴之, 深層学習, 機械学習 Professional Series, 講談社, 第 5 版, 2015.
- [14] D. Jannach, M. Zanker, A. Felfernig, A. Friedrich, (田中克己, 角谷和俊: 監訳, 奥健太, 北山大輔, 手塚太郎, 牛尼剛聡, 中島伸介, 土方嘉徳, 湯本高行, 河合由起子, 原田史子, 木村文則: 訳), 情報推薦システム入門: 理論と実践, 共立出版, 第 1 版, 2012.
- [15] 渡辺澄夫, 田村昇, 確立と統計: 情報学への架橋, コロナ社, 2016.

- [16] 松坂和夫, 集合・位相入門, 岩波書店, 第 59 版, 1968.
- [17] 加藤直樹, 数理計画法, コロナ社, 第 1 版, 2008.
- [18] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 1, Springer, 8th printing, 2006.
- [19] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 2, Springer, 8th printing, 2006.
- [20] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 5, Springer, 8th printing, 2006.
- [21] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 8, Springer, 8th printing, 2006.
- [22] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 9, Springer, 8th printing, 2006.
- [23] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 10, Springer, 8th printing, 2006.
- [24] C. M. Bishop, “Pattern Recognition and Machine Learning,” Chapter 11, Springer, 8th printing, 2006.
- [25] K. Helsgaun, “General k-opt Submoves for the LinKernighan TSP Heuristic,” Mathematical Programming Computation, Vol. 1, No. 2-3, pp. 119-163, 2009.
- [26] G. Laporte, S. Martello, “The Selective Travelling Salesman Problem,” Discrete applied mathematics, Vol. 26, No. 2-3, pp.193-207, 1990, [https://doi.org/10.1016/0166-218X\(90\)90100-Q](https://doi.org/10.1016/0166-218X(90)90100-Q).
- [27] K. H. Lim, J. Chan, S. Karunasekera, “Personalized Itinerary Recommendation with Queing Time Awareness,” Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325-334, 2017, <https://doi.org/10.1145/3077136.3080778>
- [28] I. Brillhante, F. M. Nardini, “Planning Touristic Tours with TripBuilder,” Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 757-762, 2013.
- [29] 倉田陽平, 有馬貴之, 対話的旅行計画作成支援システムの実装と評価, 第 25 回日本観光研究学会全国大会, pp. 173-176, 2010.
- [30] 小西貞則, 多変量解析入門, 岩波書店, 2010.
- [31] K. W. Leung, D. L. Lee, “A Collaborative Location Recommendation Framework based on Co-Clustering,” Proceedings of the 34 th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 305-314, 2011, <http://hdl.handle.net/1783.1/45942>.
- [32] 國藤進, 中田豊久, 羽山徹彩, 知識基盤社会のための人工知能入門, 計測自動制御学会, コロナ社, 第 1 版, 2012.
- [33] 森毅, 現代数学入門 2: マトリックス, 明治図書出版, 1964.
- [34] ユーリイ・アナトリエヴィチ・ロザノフ (佐藤健一, 佐藤由身子: 訳), 確率過程入門, 産業図

- 書株式会社, 初版, 1980.
- [35] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp. 1-43, 2012, <https://doi.org/10.1109/TCIAIG.2012.2186810>.
- [36] T. M. Chan, "More Algorithms for All-Pairs Shortest Paths in Weighted Graphs," *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pp. 590-598, 2007, <https://doi.org/10.1145/1250790.1250877>.
- [37] D. H. Ackley, G. E. Hinton, "A Learning Algorithm for Boltzmann Machines," *Cognitive Science*, Vol. 1, No. 1, pp. 147-169, 1985, [https://doi.org/10.1016/S0364-0213\(85\)80012-4](https://doi.org/10.1016/S0364-0213(85)80012-4).
- [38] R. Salakhutdinov, G. Hinton, "Deep Boltzmann Machines," *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pp. 448-445, 2009.
- [39] N. T. Kuong, E. Uchino, N. Suetake, "IVUS Tissue Characterization of Coronary Plaque by Classification Restricted Boltzmann Machine," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 21, No. 1, pp. 67-73, 2017.
- [40] T. Tieleman, "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient," *Proceedings of the 25th International Conference on Machine Learning*, pp. 1064-1071, 2008, <https://doi.org/10.1145/1390156.1390290>.
- [41] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 598, pp. 671-680, 1983, <https://doi.org/10.1126/science.220.4598.671>.
- [42] Y. Nourani, B. Andresen, "A Comparison of Simulated Annealing Cooling Strategies," *Journal of Physics A: Mathematical and General*, Vol. 31, No. 41, pp. 8373-8385, 1998.
- [43] T. Kadowaki, H. Nishimori, "Quantum Annealing in the Transverse Ising Model," *Physical Review E*, Vol. 58, pp. 5355-5363, 1998, <https://doi.org/10.1103/PhysRevE.58.5355>.
- [44] G. E. Hinton, T. J. Sejnowski, "Optimal Perceptual Inference," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.448-453, 1983.
- [45] 小西健三, 瀧和男, 木村宏一, 温度並列シミュレーテッド・アニーリング法とその評価, *情報処理学会論文誌*, Vol. 36, No. 4, pp. 797-807, 1995.
- [46] Th, Kaiser, W.K. Benz, "Floating-Zone Growth of Silicon in Magnetic Fields III, Numerical Simulation," *Physical Review E*, Vol. 58, No. 5, pp. 5355-5363, 1998, [https://doi.org/10.1016/S0022-0248\(97\)00487-9](https://doi.org/10.1016/S0022-0248(97)00487-9)
- [47] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, <https://doi.org/10.1109/5.726791>
- [48] L. A. Gatys, A. S. Ecker, M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2413-2423, 2016, <https://doi.org/10.1109/CVPR.2016.265>.
- [49] C. Li, M. Wand, "Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis," *Proceedings of the IEEE Conference on Computer Vision and Pattern*

- Recognition, pp. 2479-2486, 2016, <https://doi.org/10.1109/CVPR.2016.272>.
- [50] T. Mikolov, M. Karafiat, L. Burget, J. H. Cernoky, S. Khudanpur, "Recurrent Neural Network based Language Model," Proceedings of the 11th Annual Conference of the International Speech Communication Association, pp. 1045-1048, 2010.
- [51] N. Srivastava, et. al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, 2014.
- [52] N. Srivastava, R. Salakhutdinov, "Multimodal Learning with Deep Boltzmann Machines," Journal of Machine Learning Research, Vol. 15, pp. 2949-2980, 2014.
- [53] A. A. Efros, T. K. Leung, "Texture Synthesis by non-Parametric Sampling," Proceedings of the 7th IEEE International Conference on Computer Vision, Vol. 2, pp. 1033-1038, 1999, <https://doi.org/10.1109/ICCV.1999.790383>.
- [54] M. Matsumoto, T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator," ACM Transactions on Modeling and Computer Simulation, Vol.8, No.1, pp. 3-31, 1998, <https://doi.org/10.1145/272991.272995>.
- [55] N. Srivastava, R. Salakhutdinov, G. Hinton, "Modeling Documents with a Deep Boltzmann Machine," Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence, pp. 616-624, 2013.
- [56] X. D. Liu, S. Osher, T. Chan, "Weighted Essentially non-Oscillatory Schemes," Journal of Computational Pyhsiscs, Vol. 115, No. 1, pp. 200-212, 1994, <https://doi.org/10.1006/jcph.1994.1187>.
- [57] R. Salakhutdinov, A. Mnih, G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering," Proceedings of the 24th International Conference on Machine Learning, pp. 791-798, 2007, <https://doi.org/10.1145/1273496.1273596>.
- [58] D. Goldberg, D. Nichols, "Using Collaborative Filtering to Weave an Information Tapestry," Communications of the ACM 35.12, pp. 61-70, 1992, <https://doi.org/10.1145/138859.138867>.
- [59] Y. Koren, R. Bell, C. Volinsky, "Matrix Factorization Techniques For Recommender Systems," Computer, , Vol. 42, No. 8, pp. 30-37, 2009, <https://doi.org/10.1109/MC.2009.263>.
- [60] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," Technical Report UTML TR 2010-003, University of Toronto, 2010, https://doi.org/10.1007/978-3-642-35289-8_32.
- [61] C. Wang, D. M. Blei, "Collaborative Topic Modeling for Recommending Scientific Articles," Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining, pp. 448-456, 2011, <https://doi.org/10.1145/2020408.2020480>.
- [62] R. Salakhutdinov, A. Mnih, "Probabilistic Matrix Factorization," Advances Neural Information Processing Systems 20, pp. 1257-1264, 2007.
- [63] R. Salakhutdinov, A. Mnih, "Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo," Proceedings of the 25th International Conference on Machine Learning, pp. 880-887, 2008, <https://doi.org/10.1145/1390156.1390267>.

- [64] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 721-741, 1984, <https://doi.org/10.1109/TPAMI.1984.4767596>
- [65] J. Kawale, H. Bui, B. Kveton, L. Tran-Thanh, S. Chawla, "Efficient Thompson Sampling for Online Matrix-Factorization Recommendation," *Advances in Neural Information Processing Systems* 28, pp. 1297-1305, 2015.
- [66] M. D. Vose, "Modeling Simple Genetic Algorithms," *Journal of Evolutionary Computation*, Vol. 3, No. 4, pp. 453-472, 1995, <https://doi.org/10.1162/evco.1995.3.4.453>.
- [67] V. Nair, G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on Machine Learning*, pp. 807-814, 2010.
- [68] R. Salakhutdinov, G. Hinton, "Replicated Softmax: and Undirected Topic Model," *Advances in Neural Information Processing Systems* 22, pp. 1607-1614, 2009.
- [69] D. M. Blei, A.Y. Ng, M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, Vol. 3, pp. 993-1022, 2003.
- [70] 宇高義郎, 湯浅栄二, 大竹尚登, 藤本浩司, 京極秀樹, 品川一成, 磯西和夫, 三浦秀士, 酒井潤一, 松岡信一, 松尾湯太郎, 川田宏之, 吉田一也, 原田幸明, 塩谷義, *機械材料学*, 第4版, JSME テキストシリーズ, 日本機械学会, 2008年.
- [71] 北山智大, 田川俊夫, 軸方向磁場下における振動マランゴニ対流におよぼす上下ロッドの導電性およびゼーバック効果の影響, *日本機械学会論文集*, Vol. 83, No. 852, p. 17-00011, 2017, <https://doi.org/10.1299/transjsme.17-00011>.
- [72] S. Parl, "Principal of Sigma-Delta Modulation for Analog-to-Digital Converters," *MOTOROLA*, 1999.
- [73] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, Vol. 57, No 1, pp. 97-109, 1970, <https://doi.org/10.1093/biomet/57.1.97>.
- [74] B. Buchberger, G. E. Collins, R. Loos, R. Albrecht, "Computer Algebra Symbolic and Algebraic Computation" *ACM Special Interest Group in Symbolic and Algebraic Manipulation*, Vol. 16, No. 4, pp. 5-5, 1982, <https://doi.org/10.1145/1089310.1089312>.
- [75] A. Einstein, "The Foundation of the General Theory of Relativity," *Annalen der Physik*, Vol. 49, No. 7, pp. 769-822, 1916, <https://doi.org/10.1002/andp.19163540702>.
- [76] R. Nane, V. M. Sima, C. Pilato, J. Choi, B. Fort, A. Canis, Y. T. Chen, H. Hsiao, S. Brown, F. Ferrandi, J. Anderson, K. Bertels, "A Survey and Evaluation of FPGA High-Level Synthesis Tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 35, No. 10, pp. 1591-1604, 2016, <https://doi.org/10.1109/TCAD.2015.2513673>.
- [77] M. Yannakakis, "Expressing Combinatorial Optimization Problems by Linear Programs," *Journal of Computer and System Sciences*, Vol. 43, No. 3, pp. 441-466, 1991, [https://doi.org/10.1016/0022-0000\(91\)90024-Y](https://doi.org/10.1016/0022-0000(91)90024-Y).
- [78] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, A. J. Hudspeth, (金沢一郎, 宮下保司: 監修, 岡野栄之和田圭司, 加藤総夫, 入末篤史, 藤田一郎, 伊佐正, 定藤規弘, 大隅典子, 笠井清登: 訳), *カンデル神経科学, メディカル・サイエンス・インターナショナル*, 第3版,

2014.

発表文献

- [1] 柴田祐樹, 高間康史, 辺ベクトルを用いた観光経路推薦問題の定式化と焼きなまし法による解法の提案, 知能と情報, Vol.31, No.1, 2019 (掲載決定).
- [2] H. Shibata, H. Ishikawa, Y. Takama, “Crystalizing Effect of Simulated Annealing on Boltzmann Machine,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 23, No. 3, 2019 (Accepted).
- [3] H. Shibata, Y. Takama, “Index-based Notation for Random Variable and Probability Space,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 23, No. 4, 2019 (Accepted).
- [4] H. Shibata, Y. Takama, “Behavior Analysis of RBM for Estimating Latent Factor Vector from Rating Matrix,” *Proceedings of the 6th International Conference on Informatics, Electronics & Vision & 7th International Symposium in Computational Medical and Health Technology*, No. 15, 2017, <https://doi.org/10.1109/ICIEV.2017.8338568>.
- [5] H. Shibata, Y. Takama, “Consideration on Crystalizing Simulated Annealing for Boltzmann Machine,” *Proceedings of the 10th International Conference on Management of Gigital EcoSystems*, pp. 87-93, 2018, <https://doi.org/10.1145/3281375.3281401>.
- [6] 柴田祐樹, 高間康史, 評価値行列からの潜在因子推定における制限ボルツマンマシンの挙動解析, 第10回 Web Intelligene and Interaction (WI2) 研究会, WI2-2017-2, pp. 7-12, 2017.
- [7] 柴田祐樹, 高間康史, スポット推薦を伴う経路推薦手法の提案, 第18回 Interactive Information Access and Visual Mining (SIGAM) 研究会, 日本人工知能学会, SIG-AM-18-07, pp. 40-46, 2018.
- [8] 柴田祐樹, 高間康史, 辺ベクトルを用いた定式化による経路推薦手法の提案, 第32回人工知能学会全国大会, IE2-02, 4 pages, 2018.

謝辞

私は修士まで航空宇宙工学を専門としていました。そこで教わった理論と、研究に対する教授陣の熱意は、今の私が研究を進めていくうえでの基礎となり、今後そう変わることは無いと思えるもので、学部3年までの間に、博士後期過程へ進む決意を与えてくれるに十分なものでした。このような先生方に出会えたことは幸いでした。

修士では田川俊夫先生のもとで数値流体力学を学びました。田川先生は物理学に精通する方で、その豊富な知識を、多くの時間を割き私に与えてくださいました。先生のもとで学んだ電磁流体、熱流体に関する理論は多くの分野で共通するものがあり、情報科学における今でも私の研究にアイデアを与えてくれます。ファイマンの残した物理学書を読んだ輪講の日々は、文化的で、満たされるものでした。その後、期待に応えることができず、私は専門を変えようと言いついた時にも、暖かく送り出していただきましたこと、感謝いたします。

修士時代、私に研究費を提供してくださった新日鐵住友金属社の方々、特に私の研究に目を止めてくださった高谷幸司様、岡田信宏様に感謝いたします。

私は人工知能の特異点に賭けるため、博士後期において専攻を変えました。

高間康史先生に出会ったのは修士2年の夏で、ふらっと、現れたと言う印象だそうですが、実のところ私は保守的な性格で、外を調べるのが苦手であり、わらにすぎる思いでした。とにかく人工知能と検索し、田川先生の部屋の4つ程隣、高間先生の部屋に向かう結果となりました。高間先生にはその後いくらかの研究室を推薦していただきましたが、手続きに戸惑っている間に、時間的余裕がなくなり、親身に話を聞いてくださった高間先生のところへ行こうと思い、12月の願書締め切り寸前に再度連絡をすることになりました。このような事態にもかかわらず、2つ返事で入学を許可してくれた先生の懐の広さにこの3年間憧れていました。なかなか研究が進まず、期限を守れないことが多い私を高間先生は厳しく指導してくださいました。また、論文の書き方についても、航空時代には真面目に取り組んでおらず、まともな文書を書けませんでした。ここに来て再度丁寧に面倒を見てもらえ、今では多少自信を持って書くことができるようになりました。私はこの研究室に入れてよかったです。いかにして卒業するか、ギリギリの状態で作戦会議を重ねる日々は、辛くも楽しいもので、良い思い出となりました。博士課程というものは、超高等教育課程であり、哲学、紳士の立ち振舞いの教育すら受けるものでした。今後も、様々なご指導をよろしく願います。

本論文を査読して頂いた、山口亨先生、小町守先生、川本一彦先生には、貴重なお時間を割いていただき、私の論述の弱い部分を補強可能な指摘をしてくださったことに深く感謝いたします。この御恩は今後の研究成果をもって返します。

台湾に留学した際、陳履恆先生には大変お世話になりました。暖かく迎えてくれ、研究のきっか

けを与えてくださったことは忘れません。洪さんには台湾で様々な世話をさせていただき、不安なく過ごせたことを感謝します。

外から入ってきた私と仲良くしてくれた高間研究室の方々に感謝します。特に 2016 年度の同期の方々には、入ったばかりの頃、会議の発表で満足な進捗を示せない私を見て気まずい思いをしていたかもしれません。申し訳なかったのと、それでも仲良くしてくれたことに感謝します。坂元陽亮くん、白石雄也くんとはよく遊んでもらいました。楽しかったです。清水勝太氏は同じ境遇の人間でもあり、輪講、雑用、添削等いろいろなことを協力してもらいました。研究者を志望する君とは今後も長い付き合いになると思います。

首都大学東京様には、学部時代より私の授業料を 9 年間に渡り減免していただきました。この支援がなければ私は学業に集中することができず、早期に退学していたと思います。

ディライト社には博士 1 年の時、生活の支援をしていただきました。ここに感謝を述べます。

その他にも、様々な事情から名前を上げることができませんが、私に金銭的援助を与えてくださいました方々に感謝いたします。

姉、叔母、祖母には一人暮らしにおける家具や雑貨等の多くを送っていただきました。毎日快適に暮らせたこと感謝いたします。

学術とは無縁であったはずの母が、私を幼少より学者として育てることを諦めなかった理由は未だわかりませんが、自らの厳しい生い立ちや、私を育てている間においても貧しい日々の中、ここまで支えてくれたことに感謝します。

その他、助けていただきました方について詳細を書けないこと、お許しください。私は運良く助けられて生きてきました。これに対し、真面目に生きて恩を返さなければなりません。この命題を恒真とするため、善処していきます。