

学位論文 博士(工学)

機械学習と量子コンピュータによる
分子シミュレーションを用いた研究開発の高効率化

2021 年度

慶應義塾大学大学院理工学研究科

遠藤 克浩

目次

第 1 章	序論	1
1.1	緒言	1
1.2	本研究の目的と本論文の構成	3
第 2 章	機械学習のための数学	4
2.1	数式の表記	4
2.2	線形代数	5
2.3	関数と微分積分	10
2.4	確率の基礎	15
2.5	統計の基礎	22
2.6	確率分布間の距離	30
第 3 章	機械学習・深層学習の基礎	32
3.1	機械学習の分類	32
3.2	線形回帰モデル	34
3.3	深層学習の基礎	41
3.4	深層生成モデルの基礎	47
第 4 章	分子シミュレーションの基礎	52
4.1	MC シミュレーション	52
4.2	MD シミュレーション	54
第 5 章	MD シミュレーションの高速な代理モデルの開発	60
5.1	本章の目的と概要	60
5.2	確率的時間発展を行う代理モデルの理論的整理	62
5.3	確率的時間発展を行う代理モデル MD-GAN の構築	65

5.4	MD-GAN によるポリエチレン溶融体中心座標の平均二乗変位の高速計算	76
5.5	結論と今後の展望	86
第 6 章	深層学習を用いた MD シミュレーション結果を自動解析する手法の開発	89
6.1	本章の目的と概要	89
6.2	ローカルダイナミクスアンサンブル (LDE) について	91
6.3	LDE 間の距離の測定	93
6.4	LDE 間の距離の埋め込み	95
6.5	LDE 間の距離の増減に関わるダイナミクスサンプルの特定	96
6.6	アミノ酸水溶液に対する解析例	97
6.7	結論と今後の展望	106
第 7 章	機械学習による MC シミュレーション高速化法の開発	111
7.1	本章の目的と概要	111
7.2	サンプリングアルゴリズムの問題点と先行研究	113
7.3	CNMF モデルによる同変性を備えた分布変換	116
7.4	MFMH 法による CNMF モデルに基づくサンプリングアルゴリズム	119
7.5	ISF 法による等圧縮かつ近接配置のない提案分布の実現	121
7.6	SL 法による系ごとに最適な提案分布の実現	126
7.7	LJ 粒子系を用いた MC シミュレーションの効率評価	129
7.8	結論と今後の展望	135
第 8 章	ゲート式量子コンピュータによる MC シミュレーションの高速化に向けた検討	137
8.1	本章の目的と概要	137
8.2	量子コンピュータを用いた汎用的な自己学習サンプリングアルゴリズムの定式化	139
8.3	1次元 QFT サンプラー	144
8.4	古典コンピュータによる Quantum-inspired QFT サンプラー実装	146
8.5	多段化による多次元 QFT サンプラー	147
8.6	QFT サンプラーの性能評価実験	150
8.7	結論と今後の展望	154
第 9 章	結論	157

謝辭	159
参考文献	162

第 1 章

序論

本論文は、機械学習と量子コンピュータによる分子シミュレーションを用いた研究開発の効率化についての研究成果をまとめたものである。本章では、先行研究における本研究の位置づけを簡潔にまとめ、本研究の目的と本論文の構成について述べる。

1.1 緒言

分子シミュレーションは、分子の振る舞いを調べるための汎用的かつ強力な手法である。ポリマーやタンパク質などの高分子化合物 [1, 2]、メタンハイドレードや水蒸気の核生成 [3, 4]、カーボンナノチューブやナノスリット内のコンファインド系 [5, 6] など、様々な物性を解明するために分子シミュレーションの応用が広がっている。

分子シミュレーションは計算方法の違いから分子動力学 (MD) シミュレーションとモンテカルロ (MC) シミュレーションの 2 つに大別される。MD シミュレーションは、Newton の運動方程式に従って原子・分子の時間発展を計算することによりシミュレーションを行う手法である。一方、MC シミュレーションは、温度や圧力などの指定したマクロ条件における尤もらしい原子・分子配置を、統計力学に基づいて確率的にサンプリングすることによりシミュレーションを行う手法である。

MD シミュレーションは、時間発展を計算するため、分子が時間発展に伴って辿る過程を調べることができるという利点がある。その一方、MC シミュレーションは、分子の時間発展とは無関係に系の取り得る状態を列挙できるため、MD シミュレーションより高速に実行できる可能性があるという利点がある。しかしながら、適用範囲の拡大により、MD シミュレーション・MC シミュレーション双方にて以下に述べる解決すべき課題がある。

まず、MD シミュレーションにおいては、近年になって巨大な系や長時間のシミュレーションが必要な系に対しても適用されており、系の大きさは約 10^9 粒子 [7]、シミュレーションの長さは約 10^{10} 時間ステップ [8] にも達している。このようなシミュレーションには膨大な計算資源が必要となってしまう。このため計算時間を可能な限り小さくするために、租視化による粒子数の削減 [9, 10] や MD シミュレーションを高速に計算する専用ハードウェアの開発 [11] など、今に至るまで多大な努力がなされている。特に、空間方向には領域分割によって並列計算が可能である [12] が、時間方向には演算の逐次性から単純な並列化ができないため、長時間のシミュレーションを実行することは、粒子数が多い系のシミュレーションと比べ難しい。

また、MD シミュレーション実行後に出力されるデータは単なる大量の分子配置の時系列であり、そこから有意義な知見を取り出すためには専門家の知識に基づく解析が必要である。専門家は、異なる系や、系内の性質が異なる部分の違いを、解析したい対象に合わせて相関係数やオーダーパラメータを設計することによって見つけ出す。これは分子のダイナミクスが持つ情報をそぎ落として見たいものだけを見ていることにほかならず、シミュレーションデータが本来持つ情報をほとんど活用できていないことになる。

さらに、MC シミュレーションにおいては、理論上時間発展とは無関係に系の取り得る状態を列挙できるとしたが、実用的には大きな問題がある。MC シミュレーションは統計力学に基づいた分子配置の確率分布であるボルツマン分布をサンプリングすることによってシミュレーションを行うが、ボルツマン分布は局所的にも大局的にも変化の激しい構造をしている分布である。そのため、ボルツマン分布の効率的なサンプリングが難しい。

一方、近年、上記のような分子シミュレーションの課題の解決策として、機械学習や量子コンピュータを応用した手法が注目を集めている。前者の機械学習は、与えられたデータから有用な情報を自動的に抽出し、何らかの値を予測するものである。機械学習を分子シミュレーションに適用した手法の例としては、MD シミュレーションの相互作用計算の際に、高コストだが正確な相互作用を計算できる量子化学計算を、高速に計算可能な機械学習モデルに代替する手法 [13, 14, 15] や、タンパク質等の反応経路を機械学習を用いて低次元に落とし込むことによって、複雑な反応経路を可視化する手法 [16, 17]、MC シミュレーション結果から自由エネルギー曲面を機械学習によって効率的に表現する手法 [18, 19]、MC シミュレーションがサンプリングする確率分布を機械学習を用いて潜在空間とよばれる低次元の空間に変換することにより、サンプリングしやすいフラットな分布に変換する手法 [20]、など様々な手法が提案されている。

また、後者の量子コンピュータは、量子力学から導かれる量子状態の重ね合わせや確率的測定といった性質を利用して、従来の古典コンピュータでは実現不可能な計算が可能と

するものである。分子シミュレーションに関連する量子コンピュータの応用例としては、主に量子化学計算 [21, 22, 23] が挙げられる。量子化学計算は分子シミュレーションの計算に用いられる原子間のポテンシャルエネルギーを決定する。量子化学計算においては、量子状態の重ね合わせという性質から、系のサイズに対して計算すべき相互作用の組み合わせが指数的に膨大してしまうため、古典コンピュータを用いた計算による計算が非現実的となってしまう。そこで、量子状態の計算には同じく量子状態を用いた計算を活用することで、計算を可能にする。また、量子科学計算の他にも、量子が本質的に確率論的な振る舞いを利用することを利用して、古典的にサンプリングが難しい分布を高速に実行する研究 [24] がされており、分子シミュレーションへの応用が期待される。

このように機械学習・量子コンピュータの分子シミュレーション分野への適用は、まさに今、応用範囲が広がっているものであり、機械学習・量子コンピュータを用いることで上述の課題解決に貢献できる可能性がある。

1.2 本研究の目的と本論文の構成

そこで本研究では、機械学習・量子コンピュータを分子シミュレーションに応用することにより

- MD シミュレーションの高速な代理モデルを開発する
- MD シミュレーション結果から物性の違いに寄与する有用な特徴量を自動抽出する
- MC シミュレーションを高速に実行するサンプリングアルゴリズムを開発する

ことを目的とする。

まず、2-4 章で本研究の基礎となる技術について説明する。2 章では本研究の理解に必要なとなる数学的知識・機械学習・量子コンピュータの基礎について、3 章では MD シミュレーションの基礎について、4 章では MC シミュレーションの基礎について述べている。その後、5-8 章では各目的を達成するために開発した技術とその結果についてを報告する。5 章では深層学習を用いた MD シミュレーションの高速な代理モデルの構築と性能評価について、6 章では深層学習を用いた MD シミュレーション結果を自動解析する枠組みの提案とその性能評価、7,8 章ではそれぞれ機械学習と量子コンピュータを用いた高速サンプリングアルゴリズムについての検討を報告する。最後に第 9 章で全体の結論を述べる。

第2章

機械学習のための数学

本章では、以降の章で用いる機械学習手法に必要となる基礎的な数学的知識について述べる。

2.1 数式の表記

a	スカラー
\mathbf{a}	ベクトル
$(\mathbf{a})_i, a_i$	ベクトル \mathbf{a} の i 番目の要素
\mathbf{A}	行列
$(\mathbf{A})_{ij}, a_{ij}$	行列 \mathbf{A} の i 行 j 列目の要素
$(\mathbf{A})_i, \mathbf{a}_i$	行列 \mathbf{A} の i 行目のベクトル
$\mathbf{A}_{n:m}$	行列 \mathbf{A} のうち n 行目から m 行目までを集めた行列
$(\mathbf{a}_n, \mathbf{a}_{n+1}, \dots, \mathbf{a}_m)$	
$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	自然数集合, 整数集合, 実数集合
A	定数または確率変数
\mathbf{a}^T	転置
\mathbf{a}^*	複素共役
\mathbf{a}^\dagger	転置複素共役 (\mathbf{a}^{T*})
$\ \mathbf{x}\ $	\mathbf{x} のノルム
$\delta(x)$	ディラックのデルタ関数
$\max(a, b)$	a と b のうち大きい値
$\max f(x)$	関数 $f(x)$ の最大値

$\min f(x)$	関数 $f(x)$ の最小値
$\operatorname{argmax}_x f(x)$	関数 f が最大になるような x あるいはその集合
$\sum_{i \in S} x_i$	集合 $\{x_i\}$ の添え字集合 S にわたる総和
$\sum_{i=a}^b x_i$	$a, b \in \mathbb{N}$ に対して, 総和 $x_a + x_{a+1} + \cdots + x_b$
$\prod_{i=a}^b x_i$	$a, b \in \mathbb{N}$ に対して, 総積 $x_a x_{a+1} \cdots x_b$
$(\text{数式}) _{a=b}$	(数式) 中の変数 a に b を代入した式

2.2 線形代数

ベクトル・行列などを含む, 本論文で必要となる線形代数の基礎的な知識について述べる.

2.2.1 ベクトルとその和・実数倍・内積・ノルム

次のように $K (\in \mathbb{N})$ 個の実数 $\{a_i \in \mathbb{R} | i = 1, \dots, K\}$ を並べたもの

$$\mathbf{a} = (a_1, a_2, \dots, a_K) \quad (2.1)$$

をベクトルという. 例えば, 空間上の任意の点の位置は, その空間の次元を K としたとき, 適切な座標系の下での座標 $\{x_i \in \mathbb{R} | i = 1, \dots, K\}$ を用いて

$$\mathbf{x} = (x_1, x_2, \dots, x_K) \quad (2.2)$$

と表すことができ, これを位置ベクトルという. 数直線上の点の座標は, 1次元空間であるから1個の実数で表現でき, 我々のすむ空間は3次元であるため, その空間上の点の座標は3個の実数で表現できる.

ベクトル \mathbf{a} に対してその i 番目の成分を $(\mathbf{a})_i$ あるいは a_i と書くことにする. ベクトルの和を次のように定義する. 任意の2つのベクトル $\mathbf{a} = (a_1, a_2, \dots, a_K)$, $\mathbf{b} = (b_1, b_2, \dots, b_K)$ について,

$$\mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \dots, a_K + b_K) \quad (2.3)$$

と定義する. 位置ベクトルを用いて考えると, 位置ベクトル \mathbf{a} と位置ベクトル \mathbf{b} で表される2点について, 2点の位置ベクトルの和によって得られる位置ベクトル $\mathbf{a} + \mathbf{b}$ が表す点とは, 2点の各次元の座標をそれぞれ加算した座標にある点を表すことになる.

ベクトルの実数倍を次のように定義する. 任意のベクトル $\mathbf{a} = (a_1, a_2, \dots, a_K)$ と $c \in \mathbb{R}$ について, \mathbf{a} の c 倍を

$$c\mathbf{a} = (ca_1, ca_2, \dots, ca_K). \quad (2.4)$$

と定義する. 位置ベクトルを用いて考えると, 位置ベクトル $c\mathbf{a}$ で表される点は, 位置ベクトル \mathbf{a} で表される点の各次元の座標をすべて c 倍した座標にある点を表すことになる.

ベクトルの和と実数倍から, ベクトル \mathbf{a} の -1 倍, すなわち $(-1)\mathbf{a}$ を省略して $-\mathbf{a}$ と書き, またベクトル \mathbf{a}, \mathbf{b} について, \mathbf{a} と $-\mathbf{b}$ の和, すなわち $\mathbf{a} + (-\mathbf{b})$ を省略して $\mathbf{a} - \mathbf{b}$ と書く.

ベクトルの内積を次のように定義する. 任意の 2 つのベクトル $\mathbf{a} = (a_1, a_2, \dots, a_K), \mathbf{b} = (b_1, b_2, \dots, b_K)$ について, 内積 $\mathbf{a} \cdot \mathbf{b}$ とは

$$\mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + \dots + a_Kb_K \quad (2.5)$$

である. 式からわかる通り, ベクトルの内積は実数になる. 内積は定義から交換法則を満たす. また, 内積を用いて, ベクトルのノルムを次のように定義する. 任意のベクトル \mathbf{a} について, ノルム $\|\mathbf{a}\|$ とは,

$$\|\mathbf{a}\| = \sqrt{\mathbf{a} \cdot \mathbf{a}} \quad (2.6)$$

である*1. ベクトルの内積は, 直交するとき 0 になる.

2 つのベクトルの内積とノルムの間には,

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta \quad (2.7)$$

という関係が成り立つ. ここで θ はベクトル \mathbf{a} と \mathbf{b} のなす角である.

2.2.2 行列とその和・実数倍・転置・積

次のような m 行 n 列の数の配列

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \quad (2.8)$$

*1 $\|\mathbf{a}\| = \sqrt{\mathbf{a} \cdot \mathbf{a}} = \sqrt{a_1^2 + a_2^2 + \dots + a_K^2}$ であるから, 三平方の定理より, ベクトルのノルムは, 原点との距離を表す.

を, (m 行 n 列の) 行列という. 行列 A に対してその (i, j) 成分を $(A)_{ij}$ あるいは a_{ij} と書くことにする. 行列の上から i 行目かつ左から j 列目の数 a_{ij} を, 行列の (i, j) 成分という. この行列は m 個の行を持ち, また同じく n 個の列を持つ.

たとえば, i 番目の行は

$$(a_{i1} \quad a_{i2} \quad \cdots \quad a_{in}) \quad (2.9)$$

であり, これを行ベクトルという. また, j 番目の行は

$$\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \quad (2.10)$$

であり, これを列ベクトルという.

行列 A と B

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} \quad (2.11)$$

に対して, A と B の行列の和は, 次のように定義される

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{pmatrix}. \quad (2.12)$$

また, 行列 A に対して, 行列の実数 c 倍は, 次のように定義される

$$c\mathbf{A} = \begin{pmatrix} ca_{11} & ca_{12} & \cdots & ca_{1n} \\ ca_{21} & ca_{22} & \cdots & ca_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ca_{m1} & ca_{m2} & \cdots & ca_{mn} \end{pmatrix}. \quad (2.13)$$

行列の行と列をとりかえる操作を, 行列の転置とよぶ. 行列 A に対して, 行列の転置を A^T と書き, 次のように定義される

$$\mathbf{A}^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{m2} & \cdots & a_{mn} \end{pmatrix}. \quad (2.14)$$

すなわち、転置とは対角線を軸として反転させる操作である。

m 行 n 列の行列 \mathbf{A} と n 行 s 列の行列 \mathbf{B}

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix}, \mathbf{B} = (\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_s), \quad (2.15)$$

に対して、行列の積は次のように定義される

$$\mathbf{AB} = \begin{pmatrix} \mathbf{a}_1 \cdot \mathbf{b}_1 & \mathbf{a}_1 \cdot \mathbf{b}_2 & \cdots & \mathbf{a}_1 \cdot \mathbf{b}_s \\ \mathbf{a}_2 \cdot \mathbf{b}_1 & \mathbf{a}_2 \cdot \mathbf{b}_2 & \cdots & \mathbf{a}_2 \cdot \mathbf{b}_s \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_m \cdot \mathbf{b}_1 & \mathbf{a}_m \cdot \mathbf{b}_2 & \cdots & \mathbf{a}_m \cdot \mathbf{b}_s \end{pmatrix}. \quad (2.16)$$

ここで、 \mathbf{a}_i は行列 \mathbf{A} の i 番目の列ベクトル、 \mathbf{b}_j は行列 \mathbf{B} の j 番目の行ベクトルであるとする。すなわち、 m 行 n 列の行列 \mathbf{A} と n 行 s 列の行列 \mathbf{B} の積は、 m 行 s 列の行列となる。行列の積は結合法則を満たす。

2.2.3 行列による連立一次方程式の表現

次の m 元 1 次方程式

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \quad (2.17)$$

は、次のように表現できる

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \quad (2.18)$$

ここで、 n 次元の縦ベクトルを n 行 1 列行列と見做していることに注意する。また、各行列・ベクトルを

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad (2.19)$$

と書くことにすれば,

$$Ax = b \quad (2.20)$$

と表記できる. よって, 連立一次方程式を行列に関する方程式として記述することができる.

2.2.4 特別な名前がついている行列

- 正方行列・・・行の数と列の数が同じ行列のこと
- 正規行列・・・逆行列(次節参照)をもつ行列のこと.
- 対角行列・・・対角成分以外はゼロの値を持つ正方行列のこと
- 単位行列・・・対角成分がすべて1である対角行列のこと. I と表記する.
- 零行列・・・すべての成分がゼロの値を持つ行列のこと. O と表記する.

2.2.5 逆行列と連立一次方程式の解法

2つの正方行列 A, B に対して, 次の式が成り立つ関係

$$AB = BA = I \quad (2.21)$$

にあるとき, B を A の逆行列という. 定義から, 明らかに A は B の逆行列である. 逆行列を簡単に表記するために, A に逆行列が存在するとき, A の逆行列を

$$A^{-1} \quad (2.22)$$

と書く. 従って, A に逆行列が存在するとき, 次が常に成り立つ

$$AA^{-1} = A^{-1}A = I. \quad (2.23)$$

さて, 連立一次方程式の行列を用いた表現

$$Ax = b \quad (2.24)$$

について, x を求めたい. ここで, A の逆行列 A^{-1} が存在するとき, 両辺に右から A^{-1} を掛ければ

$$A^{-1}Ax = A^{-1}b \quad (2.25)$$

$$Ix = A^{-1}b \quad (2.26)$$

$$x = A^{-1}b. \quad (2.27)$$

より, x は求まった.

2.2.6 行列式とトレース

■**行列式** N 行 N 列の正方行列 \mathbf{A} について、行列 A の行列式 $\det(\mathbf{A})$ とは

$$\det(\mathbf{A}) = \sum_{\pi \in \Pi} \text{sgn}(\pi) \prod_{i=1}^N a_{i\pi(i)} \quad (2.28)$$

のことである。ここで Π は 1 から n までの全ての置換を含む集合であり、また $\text{sgn}(\pi)$ は π が奇置換のとき -1、偶置換のとき 1 となる関数である。

■**トレース** N 行 N 列の行列 A について、行列 A のトレース $\text{tr}(A)$ とは

$$\text{tr}(A) = \sum_{i=1}^N a_{ii} \quad (2.29)$$

のことである。すなわち、行列のトレースは行列の対角成分の和のことである。

2.3 関数と微分積分

関数の連続性や勾配などを含む、本論文で必要となる関数と微分積分の基礎的な知識について述べる。

2.3.1 関数とその極限

2つの集合 X, Y の元 x, y に対して、 x に対応する y がひとつだけ定まる関係があるとき、その対応関係を

$$y = f(x) \quad (2.30)$$

などと書き、この対応関係 f を関数とよぶ。また、このとき集合 X を関数 f の定義域、集合 Y を関数 f の値域と呼ぶ。

定義域、値域がともに実数であるような関数 $f(x)$ と実数 $a, \alpha \in \mathbb{R}$ があって、任意の正の実数 ϵ に対して、ある正の実数 δ が存在して、

$$0 < \|x - a\| < \delta \rightarrow \|f(x) - \alpha\| < \epsilon \quad (2.31)$$

を満たすとき、関数 $f(x)$ は $x = a$ で α に収束するといひ、

$$\lim_{x \rightarrow a} f(x) = \alpha \quad (2.32)$$

と表す。

2.3.2 連続関数

定義域、値域がともに実数であるような関数 $f(x)$ について、ある点 $x = a$ において、関数の極限が存在して、かつ

$$\lim_{x \rightarrow a} f(x) = f(a), \quad (2.33)$$

すなわち極限が $x = a$ における関数 f の値と一致するとき、関数 $f(x)$ は a で連続であるという。また、ある範囲内のすべての点 x において、関数 $f(x)$ が連続であるとき、その範囲において関数 $f(x)$ は連続関数であるという*2。

2.3.3 リプシッツ連続関数

定義域が実数または実数ベクトル、値域が実数または実数ベクトルであるような関数 f について、任意の実数ベクトルの組 $\mathbf{x}_1, \mathbf{x}_2$ について、ある実数 γ が存在して

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| < \gamma \|\mathbf{x}_1 - \mathbf{x}_2\| \quad (2.34)$$

が成り立つとき、関数 f は γ -リプシッツ連続関数であるといい、

$$\|f\|_L < \gamma \quad (2.35)$$

と書く。すなわち、 γ -リプシッツ連続関数は、入力する値の変化量に対して、出力が高々 γ 倍しか変化しないことを表している。また、 γ の値について気にしないとき、すなわち関数 f について、何らかの実数 γ が存在して $\|f\|_L < \gamma$ を満たすということだけを述べたいとき、単に f をリプシッツ連続関数であるということがある。

2.3.4 関数の微分・導関数

関数 $f(x)$ について、次の極限

$$\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (2.36)$$

が存在するとき、これを $x = a$ における $f(x)$ の微分係数とよび、

$$f'(a) \text{ や } \frac{d}{dx} f(a) \quad (2.37)$$

*2 連続関数同士の和、積、商は同じく連続関数になる。(ただし商については分母がゼロにならない関数であるとする)

などと書く。幾何学的には、 $f'(a)$ は $f(x)$ の $x = a$ における傾きと解釈できる。

また、 x を x における $f(x)$ の微分係数と対応づけるとき、すなわち関数

$$\frac{d}{dx}f(x) \quad (2.38)$$

を、 $f(x)$ の導関数または単に微分とよぶ。

2.3.5 多変数による偏微分と勾配・ヘッセ行列

定義域が複数の実数変数の組 (a, b, \dots) であって、値域が実数である関数 $y = f(a, b, \dots)$ について、関数 f の各変数に対する偏微分は、そのほかの変数を定数とみなしたときの微分、すなわち

$$\frac{\partial}{\partial a}f(a, b, \dots) = \lim_{x \rightarrow a} \frac{f(x, b, \dots) - f(a, b, \dots)}{x - a} \quad (2.39)$$

と定義される。

また、定義域が実数ベクトル $\mathbf{w} = (w_1, w_2, \dots)$ である関数 $y = f(\mathbf{w})$ について、関数 f の実数ベクトル \mathbf{w} における勾配ベクトル $\nabla_{\mathbf{w}}f(\mathbf{w})$ は、ベクトルの各成分による関数 f の偏微分からなるベクトル

$$\nabla_{\mathbf{w}}f(\mathbf{w}) = \begin{pmatrix} \frac{\partial}{\partial w_1}f(\mathbf{w}) \\ \frac{\partial}{\partial w_2}f(\mathbf{w}) \\ \vdots \end{pmatrix} \quad (2.40)$$

と定義される。勾配ベクトルは微分の変数への一般化であるといえる。つまり、勾配ベクトルは基本的には各成分ごとの微分を集めたものと考えて差し支えない。勾配ベクトル $\nabla_{\mathbf{w}}f(\mathbf{w})$ は、数式のわかりやすさのために

$$\frac{d}{d\mathbf{w}}f(\mathbf{w}) \quad (2.41)$$

と書く場合もある。

また、勾配ベクトルの各要素を構成する関数に対してさらに勾配ベクトル計算して、そ

れらを列ベクトルとみなして行方向に並べた行列

$$\nabla_{\mathbf{w}}^2 f(\mathbf{w}) = \begin{pmatrix} \nabla_{\mathbf{w}} \frac{\partial}{\partial w_1} f(\mathbf{w}) \\ \nabla_{\mathbf{w}} \frac{\partial}{\partial w_2} f(\mathbf{w}) \\ \vdots \end{pmatrix} \quad (2.42)$$

$$= \begin{pmatrix} \frac{\partial}{\partial w_1} \frac{\partial}{\partial w_1} f(\mathbf{w}) & \frac{\partial}{\partial w_2} \frac{\partial}{\partial w_1} f(\mathbf{w}) & \cdots \\ \frac{\partial}{\partial w_1} \frac{\partial}{\partial w_2} f(\mathbf{w}) & \frac{\partial}{\partial w_2} \frac{\partial}{\partial w_2} f(\mathbf{w}) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad (2.43)$$

を関数 f の実数ベクトル \mathbf{w} におけるヘッセ行列という。

2.3.6 1次近似と最急勾配

定義域が実数ベクトル \mathbf{w} である関数 $y = f(\mathbf{w})$ について、関数 $f(\mathbf{w})$ の点 \mathbf{w}_0 周りの一次近似（1次のテイラー展開）とは

$$f(\mathbf{w} + \mathbf{h}) = f(\mathbf{w}) + \nabla_{\mathbf{w}} f(\mathbf{w}) \cdot \mathbf{h} + R_2 \quad (2.44)$$

のことである。ただし R_2 について

$$R_2 = \frac{1}{2} \mathbf{h}^T \nabla_{\mathbf{w}}^2 f(\mathbf{w} + \theta \mathbf{h}) \mathbf{h} \quad (2.45)$$

となる $0 < \theta < 1$ が存在する（ラグランジュの剰余項）。

ある実数ベクトル \mathbf{w} が与えられ、かつその関数 $f(\mathbf{w})$ の値および勾配 $\nabla_{\mathbf{w}} f(\mathbf{w})$ 、ヘッセ行列 $\nabla_{\mathbf{w}}^2 f(\mathbf{w})$ が有限の値をとっているとする。ただし、勾配はゼロベクトルではないとする。このとき、関数 $f(\mathbf{w})$ がベクトル \mathbf{w} の近傍で最も増加する方向は、勾配 $\nabla_{\mathbf{w}} f(\mathbf{w})$ に一致する。なぜなら、単位ベクトル \mathbf{n} と十分に小さい微小量 ϵ に対して、単位ベクトル \mathbf{n} の方向に微小に進んだときの変化量は

$$f(\mathbf{w} + \epsilon \mathbf{n}) - f(\mathbf{w}) \simeq \epsilon \nabla_{\mathbf{w}} f(\mathbf{w}) \cdot \mathbf{n} \quad (2.46)$$

$$= \epsilon \|\nabla_{\mathbf{w}} f(\mathbf{w})\| \|\mathbf{n}\| \cos \theta \quad (2.47)$$

$$= \epsilon \|\nabla_{\mathbf{w}} f(\mathbf{w})\| \cos \theta \quad (2.48)$$

となり、勾配の向きと単位ベクトルの向きが一致しているとき ($\cos \theta = 1$) にこの差が最大となるからである。ここで θ は勾配と単位ベクトルのなす角である。関数 $f(\mathbf{w})$ がベクトル \mathbf{w} の近傍で最も減少する方向は勾配の逆向き $-\nabla_{\mathbf{w}} f(\mathbf{w})$ に一致することも明らかである。

2.3.7 関数の定積分とガウス積分

閉区間 $[a, b]$ に対して, これを N 分割して,

$$a = x_0 < x_1 < \cdots < x_N = b \quad (2.49)$$

とする. このとき, 連続な関数 $f(x)$ に対して, 次の和

$$\sum_{i=0}^{N-1} f(x_i)(x_{i+1} - x_i) \quad (2.50)$$

をリーマン和と呼ぶ. このとき, この分割の間隔の最大値

$$\Delta = \max_i (x_{i+1} - x_i) \quad (2.51)$$

に対して, $\Delta \rightarrow 0$ のときのリーマン和の極限

$$\lim_{\Delta \rightarrow 0} \sum_{i=0}^{N-1} f(x_i)(x_{i+1} - x_i) \quad (2.52)$$

を, 関数 $f(x)$ の閉区間 $[a, b]$ での定積分とよび,

$$\int_a^b f(x) dx \quad (2.53)$$

と書く. 定積分は, リーマン和との関係性から, 幾何学的には定積分は (符号付) 面積を表すと考えられる.

関数の積分の例として, 確率統計に頻出であり以降の章でも用いられているガウス積分について記しておく. 次の関係式

$$\int_{-\infty}^{\infty} e^{-a(x-p)^2} dx = \sqrt{\frac{\pi}{a}} \quad (2.54)$$

はガウス積分と呼ばれ (ただし a, p は定数とする), ガウス関数 e^{-x^2} の実数全体 $(-\infty, \infty)$ での積分となっている*3.

*3 余談ではあるが, 微分がガウス関数となる関数は初等関数の範囲には存在しないことが知られているから, 一般の範囲において初等的な範囲でガウス関数の積分を求めることはできないが, 実数全体に渡る積分についてはその値を求めることができる (式中の値).

2.4 確率の基礎

ベイズの定理などを含む，本論文で必要となる確率論の基礎的な知識について述べる．

2.4.1 確率（真/偽）

確率とは，不確かなことがらについて，特定の事象が真である確かさの度合いを表したものである．ある不確かなことがら A について， A が真である不確実性（不確かさ）を $p(A)$ と書き， A が真である確率とよぶ．ここで，

$$0 \leq p(A) \leq 1 \quad (2.55)$$

を満たす． $p(A) = 1$ は A が厳密に真であることを表し， $p(A) = 0$ は A が厳密に偽であることを表す．また， A が偽である確率を

$$p(\bar{A}) \quad (2.56)$$

と書き， $p(\bar{A}) = 1 - p(A)$ と定義される． A が真であることを $A = 1$ ， A が偽であることを $A = 0$ と表記すれば， $p(A)$ は

$$p(A = 1) \text{ または単に } p(1) \quad (2.57)$$

と書ける．また $p(\bar{A})$ も同様に $p(A = 0)$ または単に $p(0)$ と書ける．

2.4.2 確率（離散確率変数）

前頁のことがら A は真偽の2値（0か1）をとるものであったが，これを複数の離散値をとることがらについて拡張できる．あることがら X がとりうるものをすべて集めた集合を \mathcal{X} とする*4．ここで， X が $x \in \mathcal{X}$ である不確かさ（確率）を

$$p(X = x) \text{ または単に } p(x) \quad (2.58)$$

と書く． $p(x)$ は x に関して関数となっていると解釈できる．

確率論では，次の用語が使われる．

- X … 確率変数
- $p(X = x)$ … 確率質量関数
- \mathcal{X} … 状態空間

*4 \mathcal{X} は無限離散集合でもよい（整数全体 \mathbb{Z} など）

2.4.3 複数の状態に対する確率

前頁の確率質量関数は、状態空間の要素それぞれに確率を定義した。要素ひとつひとつではなく、全状態のうちいくつかの状態いずれかが真であることを表せると便利である。状態空間 \mathcal{X} のうちの一部の状態集合 $S \subset \mathcal{X}$ に対する確率を

$$p(S) \tag{2.59}$$

と表す。

状態空間全体のうちいずれかが真であるはずであるから、状態空間全体に対しての確率は 1 でなければならない。すなわち $p(\mathcal{X}) = 1$ が常に成り立つ。

2.4.4 和集合の確率

状態空間 \mathcal{X} の部分集合 A, B がある。このとき A, B の和集合の確率 $p(A \cup B)$ は次のように表せる

$$p(A \cup B) = p(A) + p(B) - p(A \cap B). \tag{2.60}$$

ただし、 $p(A \cap B)$ は A と B の積集合の確率を表す。とくに、 A と B が相互排他的、すなわち $A \cap B = \emptyset$ であるとき、

$$p(A \cup B) = p(A) + p(B) \tag{2.61}$$

となり、これを確率の加法定理という。

状態空間全体に対しての確率は 1 であるから、確率の加法定理より、 $\sum_{x \in \mathcal{X}} p(X = x) = p(\mathcal{X}) = 1$ が成り立つ。

2.4.5 条件付確率と乗法定理

状態空間 \mathcal{X} の部分集合 A, B がある。このとき、 B が確実であるときの A の確からしさを

$$p(A|B) \tag{2.62}$$

と書き、 B のもとでの A の条件付確率という。条件付確率と和集合の確率の間には

$$p(A \cap B) = p(A|B)p(B) \tag{2.63}$$

という関係があり、これを確率の乗法定理とよぶ。

2.4.6 同時確率 (2 変数の確率)

状態空間 \mathcal{X} において、状態が 2 つの状態空間 \mathcal{Y} , \mathcal{Z} の状態の組み合わせで記述される場合、すなわち $\mathcal{X} = \{(y, z) | y \in \mathcal{Y}, z \in \mathcal{Z}\}$ となっている場合、 $Y = y$ であってかつ $Z = z$ であるような確率を

$$p(Y = y, Z = z) \quad (2.64)$$

と書き、これを同時確率という (ただし、 Y , Z はそれぞれ状態空間 \mathcal{Y} , \mathcal{Z} に対応する確率変数であるとした)。また、 $A \subset \mathcal{Y}$ かつ $B \subset \mathcal{Z}$ であるような状態のいずれかの確からしさを表す確率を、1 変数の場合と同様に

$$p(A, B) \quad (2.65)$$

と書く*5。

2.4.7 周辺確率 (2 変数の確率)

状態空間 \mathcal{X} において、状態が 2 つの状態空間 \mathcal{Y} , \mathcal{Z} の状態の組み合わせで記述される場合、すなわち $\mathcal{X} = \{(y, z) | y \in \mathcal{Y}, z \in \mathcal{Z}\}$ となっている場合、 $Y = y$ であってかつ Z はなんでもよい状態を集めた部分集合の確率を

$$p(Y = y) \quad (2.66)$$

と書き、これを周辺確率という。定義から、周辺確率と同時確率には次の関係が成り立つ

$$p(Y = y) = \sum_{z \in \mathcal{Z}} p(Y = y, Z = z). \quad (2.67)$$

2.4.8 条件付確率と乗法定理 (2 変数の場合)

状態空間 \mathcal{X} において、状態が 2 つの状態空間 \mathcal{Y} , \mathcal{Z} の状態の組み合わせで記述される場合、すなわち $\mathcal{X} = \{(y, z) | y \in \mathcal{Y}, z \in \mathcal{Z}\}$ となっている場合に、 $Z = z$ であることが確実であるときの $Y = y$ の条件付確率を

$$p(Y = y | Z = z) \quad (2.68)$$

*5 3 つ以上の組み合わせであっても同じように考えることができる。

と書く*6. このとき, 乗法定理から条件付確率, 同時確率, 周辺確率の間には次の関係

$$p(y, z) = p(y|z)p(z) \quad (2.69)$$

があることがわかる.

2.4.9 ベイズの定理

乗法定理と周辺確率の式から, 次の関係を導くことができる

$$p(Y = y|Z = z) = \frac{p(Y = y, Z = z)}{p(Z = z)} = \frac{p(Z = z|Y = y)p(Y = y)}{\sum_{y' \in \mathcal{Y}} p(Z = z|Y = y')p(Y = y')}. \quad (2.70)$$

これをベイズの定理という. $p(Y = z|Z = z)$ を y のみの関数と見なすと, 分母はすべての y について足し合わせると 1 になるようにするための正規化定数とみなすことができるから,

$$p(Y = y|Z = z) \propto p(Z = z|Y = y)p(Y = y) \quad (2.71)$$

をベイズの定理ということもある. ベイズの定理の文脈では, この式の左辺の確率 $p(Y = y|Z = z)$ を事後確率, 右辺左の確率 $p(Z = z|Y = y)$ を尤度, 右辺右の確率 $p(Y = y)$ を事前分布という.

2.4.10 (確率論的) 独立

状態空間 \mathcal{X} において, 状態が 2 つの状態空間 \mathcal{Y} , \mathcal{Z} の状態の組み合わせで記述される場合, すなわち $\mathcal{X} = \{(y, z)|y \in \mathcal{Y}, z \in \mathcal{Z}\}$ となっている場合に, 同時分布と周辺分布の間に, 次の関係

$$p(Y = y, Z = z) = p(Y = y)p(Z = z) \quad (2.72)$$

が常に成り立つとき, 確率変数 Y と Z は (確率論的) 独立であるという*7. 2 つの確率変数が独立であるとき, 乗法定理から, (どちらから見ても) 条件付確率は周辺確率に一致する. すなわち, どちらか一方の確率変数の情報を得たとしてももう一方の情報は全く得られないということである.

*6 今までと同様に, 各引数が部分集合であってもよい.

*7 3 変数以上の場合も同様に定義する.

2.4.11 条件付独立

状態空間 \mathcal{X} において、状態が3つの状態空間 \mathcal{Y} , \mathcal{Z} , \mathcal{W} の状態の組み合わせで記述される場合、すなわち $\mathcal{X} = \{(y, z, w) | y \in \mathcal{Y}, z \in \mathcal{Z}, w \in \mathcal{W}\}$ となっている場合に、 $W = w$ の下での条件付の同時分布と周辺分布の間に、次の関係

$$p(Y = y, Z = z | W = w) = p(Y = y | W = w)p(Z = z | W = w) \quad (2.73)$$

が成り立つとき、確率変数 Y と Z はの W もとで条件付独立であるという。これは、確率変数 W について確定した情報を得た場合に Y と Z が独立となることを意味する*⁸。

2.4.12 連続変数に対する確率

状態空間 \mathcal{X} が実数であるときを考える。実数のように、状態空間が無限個の状態を含む場合において、1つ1つの状態に対する確率質量関数を考えることに意味はない。なぜなら、無限個の状態からある1つの状態に丁度ぴったり一致する確率を考えると、とりうる実際の状態が有限個であるといった特殊ケースを除いて、確率はゼロになるためである。例えば、「体重が55 kgである確率」といったとき、体重は厳密に

$$55.0000000000 \dots \text{kg} \quad (2.74)$$

であるような確率はどれくらいであるかを述べていることになる。体重が55.0000000000 kgから55.0000000001 kgであるような人は、ほとんど滅多にいない（確率が非常に小さい）。このように、状態空間が連続であるときは、状態集合のとりうる幅が小さければ小さいほど、それに対する確率は小さくなっていく。よって、体重が55 kgに完全な一致を示すという幅ゼロの確率は、ゼロとなってしまふ。このようなおかしなことが発生しないように、状態空間が実数であるときは、「体重が55kg以下である確率」のように考える。

状態 $x \in \mathcal{X}$ が実数であるとき、確率変数 X が x 以下である確率 $p(X \leq x)$ を累積分布関数とよび、

$$F(x) = p(X \leq x) \quad (2.75)$$

と書く。累積分布関数を用いると、 X がある範囲 $[a, b]$ にある確率を、次のように計算できる

$$p(a \leq x \leq b) = F(b) - F(a). \quad (2.76)$$

*⁸ 4変数以上の場合も同様に定義する。

2.4.13 確率密度関数

しかしながら、特定の状態に対する確率を計算できないのは、「体重が 55 kg であることは、60 kg であることに比べてどれくらい確からしいか」などの直観的な操作ができず、不便そうである。そこで、先ほどのように「体重が 55 kg から 55.001 kg である確率」のように、微小に幅を持たせた範囲を考えてみる。これは、累積分布関数 $F(x)$ を用いて、

$$F(x+h) - F(x) \quad (2.77)$$

と表せる（ただし h は微小量）。上式は

$$h \frac{F(x+h) - F(x)}{(x+h) - x} \quad (2.78)$$

と変形できる。上式をよく見ると、 h が 0 に限りなく近づくと、微分の定義から

$$h \frac{d}{dx} F(x) \quad (2.79)$$

となることが分かる。ここで、微小範囲 h を定数と考えれば、「体重が 55 から 55+ h kg である確率」と「体重が 60 から 60+ h kg である確率」は、それぞれ

$$h \frac{d}{dx} F(55 \text{ kg}), h \frac{d}{dx} F(60 \text{ kg}) \quad (2.80)$$

であることが分かる。したがって、微小に幅を持たせた範囲において、体重が 55 kg である確率と体重が 60 kg である確率の比について、微小量 h に関係なく

$$\frac{\frac{d}{dx} F(55 \text{ kg})}{\frac{d}{dx} F(60 \text{ kg})} \quad (2.81)$$

と計算できる。このことから、累積分布関数 $F(x)$ の微分

$$\frac{d}{dx} F(x) \quad (2.82)$$

を、特定の値の確からしさの表し方であると解釈でき、これを確率密度関数という。以降、誤解の恐れがない場合は、単純に $p(X = x)$ を確率密度関数として表すことにする。

2.4.14 確率密度関数の性質

同時分布、条件付分布、周辺分布とその関係性は、確率密度関数についても同様に適用できる。

例) 2変数の場合

- 同時確率密度関数 $\dots f(Y = y, Z = z)$
- 周辺確率密度関数 $\dots f(Y = y) = \int_{-\infty}^{\infty} f(Y = y, Z = z) dz$
- 条件付き確率密度関数 $\dots f(Y = y|Z = z) = \frac{f(Y=y, Z=z)}{f(Z=z)}$

ベイズの定理も同様に適用できる。

2.4.15 確率変数の和・積・定数倍

通常の変数と同様に、確率変数に和・積・定数倍などの演算を定義することができる。確率変数 X, Y について（そのとりうる状態に加算乗算が定義されているとき）、 X と Y の和で表される確率変数 $Z = X + Y$ の確率質量関数は、

$$p(Z = z) = \sum_{z=x+y} p(X = x, Y = y) \quad (2.83)$$

によって表される。すなわち、 $Z = z$ となる確率とは、 x と y の和が z となるような組み合わせをすべて集めた集合の確率である*⁹。

同様に、確率変数 X, Y について（そのとりうる状態に加算乗算が定義されているとき）、 X と Y の積で表される確率変数 $Z = XY$ の確率質量関数は、

$$p(Z = z) = \sum_{z=xy} p(X = x, Y = y) \quad (2.84)$$

によって表される。

また同様に、確率変数 X について（そのとりうる状態に加算乗算が定義されているとき）、 X の定数 c 倍で表される確率変数 $Z = cY$ の確率質量関数は、

$$p(Z = z) = \sum_{z=cx} p(X = x) \quad (2.85)$$

によって表される。

連続変数についても確率密度関数に対して同様の議論ができ、例えば確率変数 $Z = X + Y$ の確率密度関数は

$$p(Z = z) = \int_{-\infty}^{\infty} p(X = x, Y = z - x) dx \quad (2.86)$$

と計算できる。

*⁹ よって、 $p(Z = z) = \sum_{x \in \mathcal{X}} p(X = x, Y = z - x)$ とも書ける。

2.5 統計の基礎

期待値や最尤推定などを含む、本論文で必要となる統計学の基礎的な知識について述べる。

2.5.1 平均（期待値）

（取りうる状態に加算乗算が定義されている）離散確率変数 X について、

$$\sum_{x \in \mathcal{X}} xp(X = x) \quad (2.87)$$

を X の平均または期待値と言い、

$$\mathbb{E}_X[X] \quad \text{または} \quad \langle X \rangle_X \quad (2.88)$$

と書く。誤解を招く恐れがなければ $\mathbb{E}[X]$ や $\langle X \rangle$ 、あるいは単に μ と書く場合もある。連続変数に対しても、 $\int_{-\infty}^{\infty} xp(X = x)dx$ を平均とする。

2.5.2 確率変数の和の期待値

（そのとりうる状態に加算乗算が定義されている）確率変数 X, Y について、 X と Y の和で表される確率変数 Z

$$p(Z = z) = \sum_{z=x+y} p(X = x, Y = y) \quad (2.89)$$

がある。このとき、 Z の期待値は X の期待値と Y の期待値の和に等しい。すなわち、

$$\mathbb{E}_Z[Z] = \mathbb{E}_X[X] + \mathbb{E}_Y[Y] \quad (2.90)$$

である。これを期待値の線形性という。この関係式は X と Y が独立かどうかにかかわらず、どんなときでも成り立つ^{*10}。

^{*10} 期待値の線形性より、 $\mathbb{E}[cX] = c\mathbb{E}[X]$ が成り立つ。一方、 $\mathbb{E}_{XY}[XY] = \mathbb{E}_X[X]\mathbb{E}_Y[Y]$ は一般には成り立たないことに注意せよ。これは X と Y が独立の時のみ成り立つ。

証明)

$$\mathbb{E}_Z[Z] = \sum_{z \in \mathcal{Z}} zp(Z = z) \quad (2.91)$$

$$= \sum_{z \in \mathcal{Z}} z \sum_{z=x+y} p(X = x, Y = y) \quad (2.92)$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} (x + y)p(X = x, Y = y) \quad (2.93)$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} xp(X = x, Y = y) + \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} yp(X = x, Y = y) \quad (2.94)$$

$$= \sum_{x \in \mathcal{X}} x \sum_{y \in \mathcal{Y}} p(X = x, Y = y) + \sum_{y \in \mathcal{Y}} y \sum_{x \in \mathcal{X}} p(X = x, Y = y) \quad (2.95)$$

$$= \sum_{x \in \mathcal{X}} xp(X = x) + \sum_{y \in \mathcal{Y}} yp(Y = y) \quad (2.96)$$

$$= \mathbb{E}_X[X] + \mathbb{E}_Y[Y]. \quad (2.97)$$

2.5.3 分散・標準偏差

(取りうる状態に加算乗算が定義されている) 離散確率変数 X について,

$$\mathbb{E}_X[(X - \mu)^2] = \sum_{x \in \mathcal{X}} (x - \mu)p(X = x) \quad (2.98)$$

を X の分散とよび,

$$\text{var}_X[X] \quad (2.99)$$

と書く. $\sigma^2(X)$ や単に σ^2 と表記される場合もある.

連続変数に対しても, $\int_{-\infty}^{\infty} (x - \mu)p(X = x)$ を分散とする. 期待値の線形性より, 次が成り立つことが分かる

$$\sigma^2 = \mathbb{E}[(X - \mu)^2] = \mathbb{E}[X^2] - 2\mu\mathbb{E}[X] + \mu^2 = \mathbb{E}[X^2] - \mu^2. \quad (2.100)$$

さらに, 分散の平方根を標準偏差 (Standard deviation)

$$\text{std}[x] = \sqrt{\sigma^2} = \sigma \quad (2.101)$$

という.

2.5.4 共分散

(取りうる状態に加算乗算が定義されている) 確率変数 X, Y について,

$$\mathbb{E}_X[\mathbb{E}_Y[(X - \mu_X)(Y - \mu_Y)]] \quad (2.102)$$

を X と Y の共分散といい,

$$\text{cov}[X, Y] \quad (2.103)$$

と書く. 定義から, $\text{cov}[X, Y] = \text{cov}[Y, X]$ である (対称性). また, $\text{cov}[X, X] = \text{var}[X]$ が常に成り立つ.

期待値の線形性より, 共分散は次のように展開できる

$$\text{cov}[X, Y] = \mathbb{E}_X[\mathbb{E}_Y[(X - \mu_X)(Y - \mu_Y)]] \quad (2.104)$$

$$= \mathbb{E}_X[\mathbb{E}_Y[XY - \mu_X Y - \mu_Y X + \mu_X \mu_Y]] \quad (2.105)$$

$$= \mathbb{E}_X[\mathbb{E}_Y[XY]] - \mu_X \mathbb{E}_Y[Y] - \mu_Y \mathbb{E}_X[X] + \mu_X \mu_Y \quad (2.106)$$

$$= \mathbb{E}_X[\mathbb{E}_Y[XY]] - \mu_X \mu_Y. \quad (2.107)$$

したがって, 共分散とは積の期待値から期待値の積を引いたものである. よって X と Y が独立であれば共分散はゼロである.

2.5.5 正規分布・多次元正規分布

統計学では様々な確率分布が用いられる, そのうち, 本論文で使用される一様分布・正規分布について説明する.

■**一様分布** 確率変数 X について, X が一様分布に従うとは, X の確率質量関数, あるいは確率密度関数が

$$p(x) \propto \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases} \quad (2.108)$$

となることである. ここで, S は一様分布の台であり, この分布を範囲 S 上の一様分布という.

■**正規分布** 実数集合 \mathbb{R} が状態空間である確率変数 X について, X が正規分布に従うとは, X の確率密度関数が

$$p(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x - \mu)^2\right] \quad (2.109)$$

となることであり，この確率密度関数を特別に

$$\mathcal{N}(x|\mu, \sigma^2) \quad (2.110)$$

と書く．ただし， μ, σ^2 は正規分布のパラメータ^{*11}であり，それぞれ確率変数 X の平均と分散に等しい．とくに，平均が 0 で分散が 1 の正規分布 $\mathcal{N}(x|0, 1^2)$ を標準正規分布という．

■**多次元正規分布** K 次元実数ベクトルが状態空間 (\mathbb{R}^K) である確率変数 X について， X が多次元正規分布に従うとは， X の確率密度関数が

$$p(x, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \quad (2.111)$$

となることであり，この確率密度関数を特別に

$$\mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (2.112)$$

と書く．ただし， $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ は多次元正規分布のパラメータであり， $\mathbb{E}_{X_i}[X_i] = \mu_i, \text{cov}[X_i, X_j] = \Sigma_{ij}$ が成り立っている．すなわち，多次元正規分布のパラメータは確率変数 X の平均および共分散を設定している．

2.5.6 経験分布と分布収束

■**経験分布** 離散値のみをとる N 点からなるサンプル集合 $\mathcal{D} = \{x_1, x_2, \dots, x_N\}$ があるとす。このとき \mathcal{D} に対する経験分布を

$$p(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(x = x_i) \quad (2.113)$$

と定義する．ここで $\mathbf{1}(C)$ は指示関数であり，条件 C が真のときに 1 をとり偽のときに 0 をとる．定義から明らかな通り，経験分布はサンプルの出現割合をそのまま確率分布と設定したものである^{*12}。

サンプル集合 \mathcal{D} が連続値をとるような場合においても， \mathcal{D} に対する経験分布関数を

$$p(x) = \frac{1}{N} \sum_{i=1}^N \delta(x - x_i) \quad (2.114)$$

^{*11} μ, σ^2 が与えられたときの条件付き確率として考えることもできる。

^{*12} サンプル集合の要素がベクトルであってもよい。

と定義する。ただし $\delta(x)$ はディラックのデルタ関数である。経験分布に対する平均・分散をとくに標本平均・標本分散などという。以降、サンプル集合に対する平均・分散とは、サンプル集合に対応する経験分布の平均・分散であるとする。

サンプル集合 $\mathcal{D} = \{x_1, x_2, \dots, x_N\}$ についての標本平均を、データ集合のインデックス i を用いて

$$\langle x_i \rangle_i \quad (2.115)$$

と書くことがある。インデックスが2つあるとき、1つのインデックスにのみ平均をとることもできる。

■経験分布の分布収束 あるデータ源があって、データ源からいくらかでもサンプルを収集できるとする。いま、データ源から得られるサンプルを順に集めたサンプル集合を $\mathcal{D}_N = \{x_1, x_2, \dots, x_N\}$ とする。つまり、 \mathcal{D}_N があってデータ源からさらに1つサンプルを得たとき、そのサンプルを \mathcal{D}_N に追加した集合が \mathcal{D}_{N+1} であるとする。このとき、サンプル集合 \mathcal{D}_N に対する経験分布 $p_N(x)$ を N が小さい順に並べた経験分布の列

$$p_1(x), p_2(x), \dots \quad (2.116)$$

がある分布 $p(x)$ に分布収束するとき、データ源のサンプルは分布 $p(x)$ に従うという。

ここで、分布の列 $p_1(x), p_2(x), \dots$ が分布 $p(x)$ に分布収束するとは、分布の列に対応する累積分布の列を $F_1(x), F_2(x), \dots, p(x)$ に対応する累積分布関数を $F(x)$ であるとしたとき、任意の x について

$$\lim_{n \rightarrow \infty} F_n(x) = F(x) \quad (2.117)$$

を満たすことである。

2.5.7 相関

(取りうる状態に加算乗算が定義されている) 確率変数 X, Y について、次の値

$$\frac{\text{cov}[X, Y]}{\sqrt{\text{var}[X]\text{var}[Y]}} \quad (2.118)$$

を数 X と Y の相関という。相関は、確率変数の線形的な関係の強さを表しており、 X, Y に1次関数の関係があるとき相関は ± 1 に、 X と Y が独立であるとき相関は 0 になる。したがって、相関は X と Y の関係性の情報を定量的に表した統計量であると言える。図

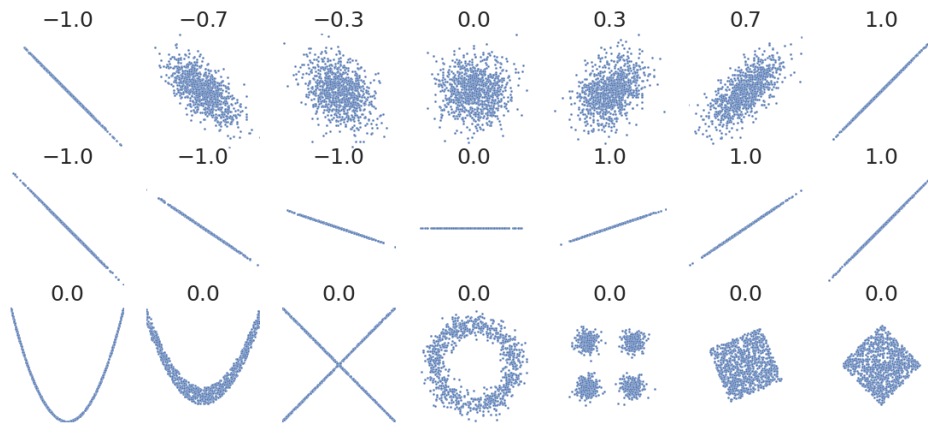


図 2.1 様々な分布とその相関係数. 1 段目は相関係数の異なる正規分布を示している. 2 段目は相関係数と分布の傾きは無関係であることを示している. 3 段目は, 相関係数が 0 であることばそれらが独立であるとは限らないことを示している.

表 2.1 測定結果

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1.28	1.86	1.73	1.73	-0.48	0.94	0.92	0.54

2.1 はいくつかの経験分布に対してとその分布の相関の値を示したものである. 線形的な関係が強いほど相関の絶対値が大きくなっていることが分かる. また, 図 2.1 の 3 段目のプロットからわかる通り, 無相関であるからといって独立であるとは限らないことに注意する. 相関はあくまで線形的な関係の強さを表す量である.

2.5.8 ベイズ推定・最尤推定・MAP 推定

上記の確率統計の知識を用いて, 何らかの量を推定することを考える. 分かりやすさのために, 例題を使って話を進める. ある物体 T の位置 (1 次元) を測定器 A で測定しようとしている. 測定器 A は平均的には正しい位置を測定できるが, 分散が 1^2 の誤差が発生することが分かっている. そこで, 正しい位置 μ が与えられたときの測定値 c の尤度を $p(c|\mu) = \mathcal{N}(c|\mu, 1^2)$ と考え, 複数回の測定によって正しい位置 p を推定しようと考えた. さて, 実際に 8 回の測定を行ったところ, 表 2.5.7 の値が測定された. このとき, 正しい位置 μ は何と推定するのが妥当かを考えたい. ただし, 各測定値は正しい位置 μ が与え

られた際に条件付き独立であるとする^{*13}.

ベイズの定理に基づいて正しい位置 μ の事後確率を計算する. 測定された値を並べた集合を $\mathcal{D} = \{x_i\}$ と書くことにする. 位置 μ の事前確率 $p(\mu)$, 位置 μ のもとでの測定された結果 \mathcal{D} が測定される尤度 $p(\mathcal{D}|\mu)$, 測定された結果 \mathcal{D} のもとでの位置 μ の事後確率 $p(\mu|\mathcal{D})$ の間には, ベイズの定理から次の関係

$$p(\mu|\mathcal{D}) \propto p(\mathcal{D}|\mu)p(\mu) \quad (2.119)$$

が成り立っている. よって, 事前確率 $p(\mu)$ と尤度 $p(\mathcal{D}|\mu)$ が分かれば位置 μ についての事後確率がわかる. いま, 測定された値は互いに独立であるから,

$$p(\mathcal{D}|\mu) = \prod_{i=1}^8 p(x_i|\mu) \quad (2.120)$$

と計算できる. また, 各測定の尤度は平均 μ 分散 1^2 の正規分布に従うから, 各 i について

$$p(x_i|\mu) = \mathcal{N}(x_i|\mu, 1^2) \quad (2.121)$$

である. よって, 位置 μ のもとでの測定された結果 \mathcal{D} が測定される尤度は,

$$p(\mathcal{D}|\mu) = \prod_{i=1}^8 \mathcal{N}(x_i|\mu, 1^2) \quad (2.122)$$

である. したがって, μ の事後確率 $p(\mu|\mathcal{D})$ は

$$p(\mu|\mathcal{D}) = p(\mu) \prod_{i=1}^8 \mathcal{N}(x_i|\mu, 1^2) \quad (2.123)$$

と求まった. 事前確率 $p(\mu)$ については, 測定者が, 測定前の位置 μ に対しての確からしさを反映すればよい.

さて, 上記により位置 μ の事後確率 $p(\mu|\mathcal{D})$ が分かったが, では具体的に位置 μ として妥当な値は何だろうか. このように最も妥当な推定量をひとつ挙げることを点推定という. 点推定のなかでも, 特に事後確率を最大化する量を求めることを最大事後確率推定 (MAP 推定) という. MAP 推定では事前分布を与える必要があるが, 事前分布に対する全く情報がない場合, 無情報事前分布とよばれる確率分布が用いられる. あるいはもう一つの点推定法として, 事前確率を考慮せず尤度を最大化する量を推定量とする最尤推定が

^{*13} この条件付き独立の仮定はサンプル集合が与えられたときの仮定としてよく用いられるものであり, 「独立なサンプル」, 「互いに独立」, 「サンプルに関連がない」などと表記される.

ある。最尤推定は形式的には事前確率をすべて 1 とした MAP 推定に一致する。最尤推定と MAP 推定どちらの推定法が優れているか、あるいは事前分布として何を選ぶべきかについては、情報量基準とよばれる統計的モデルのよさを選択する手法を用いることができる。モデル選択の詳細については、書籍 [25] を参照されたい。

まず、最尤推定量を求めてみる。この問題設定では、最尤推定とは尤度 $p(\mathcal{D}|\mu)$ を最大化するような位置 μ を決定するということである。ここで尤度

$$p(\mathcal{D}|\mu) = \prod_{i=1}^8 \mathcal{N}(x_i|\mu, 1^2) \quad (2.124)$$

は積の形になっているため取り扱いにくい。そこで対数尤度

$$\log p(\mathcal{D}|\mu) = \sum_{i=1}^8 \log \mathcal{N}(x_i|\mu, 1^2) \quad (2.125)$$

を最大化することに置き換える。このように置き換えてよい理由は対数関数が単調増加関数であるからである。計算を進めると、

$$\log p(\mathcal{D}|\mu) = \sum_{i=1}^8 \log \mathcal{N}(x_i|\mu, 1^2) \quad (2.126)$$

$$= \sum_{i=1}^8 \frac{1}{2 \times 1^2} (x_i - \mu)^2 + C \quad (2.127)$$

$$= -\frac{8}{2} \left(\mu - \frac{\sum_{i=1}^8 x_i}{8} \right)^2 + C \quad (2.128)$$

$$(2.129)$$

となる。ただし μ に関係ない定数項は C として詳細表示を省いた。よって、最尤推定量は

$$\operatorname{argmax}_{\mu} \log p(\mathcal{D}|\mu) = \frac{1}{8} \sum_{i=1}^8 x_i = 1.065 \quad (2.130)$$

と求まった。これは測定結果の標本平均である。

次に、MAP 推定量を求めてみる。MAP 推定のためには事前分布を設定する必要があるが、ここでは事前分布に対して特に事前情報をもっていないと仮定して、無情報事前分布とよばれる確率分布が用いられる。無情報事前分布の中でも、解析的な計算がしやすいように、尤度分布の計上にあわせた共役事前分布が用いられる場合がある。この問題設定

では、尤度分布は正規分布であり、これに対応する共役事前分布もまた正規分布である。よってここでは位置 μ の事前確率として平均 0・分散 1^2 の正規分布

$$p(\mu) = \mathcal{N}(x_i|0, 1^2) \quad (2.131)$$

を設定する。計算を進めると、

$$\log p(\mu|\mathcal{D}) = \log \mathcal{N}(x_i|0, 1^2) + \sum_{i=1}^8 \log \mathcal{N}(x_i|\mu, 1^2) \quad (2.132)$$

$$= \frac{1}{2 \times 1^2}(\mu - 0)^2 + \sum_{i=1}^8 \frac{1}{2 \times 1^2}(x_i - \mu)^2 + C \quad (2.133)$$

$$= -\frac{9}{2} \left(\mu - \frac{\sum_{i=1}^8 x_i}{9} \right)^2 + C \quad (2.134)$$

$$(2.135)$$

よって、MAP 推定量は

$$\operatorname{argmax}_{\mu} \log p(\mathcal{D}|\mu) = \frac{1}{9} \sum_{i=1}^8 x_i \simeq 0.946 \quad (2.136)$$

と求まった。最尤推定量と比較すると、分母が 1 増加することによって、推定量が少しだけ小さくなっていることが分かる。これは、事前分布として位置 μ が 0 に近いほど確からしいという情報を反映したものである。

2.6 確率分布間の距離

以降の章で用いる、代表的な 2 つの確率分布間の距離について述べる。

2.6.1 Kullback - Leibler(KL) ダイバージェンス

ある空間 \mathcal{X} を共通の状態空間とする 2 つの確率変数 P, Q があり、その確率分布をそれぞれを $p(\mathbf{x})$ および $q(\mathbf{x})$ と書くとする。このとき、確率分布 $p(\mathbf{x})$ に対する $q(\mathbf{x})$ の KL ダイバージェンス $D_{KL}(p||q)$ とは

$$D_{KL}[p||q] = \sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \log q(\mathbf{x}) \quad (2.137)$$

のことである。

2.6.2 Wasserstein 距離

ある空間 \mathcal{X} を共通の状態空間とする 2 つの確率変数 P, Q があり, その確率分布をそれぞれを $p(\mathbf{x})$ および $q(\mathbf{x})$ と書くとする. このとき, 確率分布 $p(\mathbf{x})$ と $q(\mathbf{x})$ の Wasserstein 距離 $W[p, q]$ とは

$$W[p, q] = \min_{\gamma \in \Pi} \sum_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) \quad (2.138)$$

のことである. ただし Π は $\sum_{\mathbf{y} \in \mathcal{X}} \gamma(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})$ かつ $\sum_{\mathbf{x} \in \mathcal{X}} \gamma(\mathbf{x}, \mathbf{y}) = q(\mathbf{y})$ を満たすすべての非負関数 γ の集合である. 関数 γ は確率変数 P から Q への輸送方法と呼ばれ, 輸送コスト $\sum_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y})$ が最も小さくなるように輸送方法を選んだ時の値を Wasserstein 距離と呼んでいる.

上式の Wasserstein 距離の定式化から, Wasserstein 距離を求める問題は輸送方法 $\gamma(\mathbf{x}, \mathbf{y})$ に対する線形計画問題となっている. 線形計画問題には必ず対応する双対形式があることが知られている. Wasserstein 距離を求める線形計画問題の双対形式は次のようになる [26]

$$W[p_A, p_B] = \max_{\|f\|_L \leq 1} \mathbb{E}_P[f(P)] - \mathbb{E}_Q[f(Q)]. \quad (2.139)$$

ただし, 関数 f は空間 \mathcal{X} を定義域としてかつ実数 \mathbb{R} が値域であるとし, $\|f\|_L \leq 1$ は関数 f が 1-リプシッツ連続であることを表している. すなわち, 1-リプシッツ連続関数 f の中で $(\mathbb{E}_P[f(P)] - \mathbb{E}_Q[f(Q)])$ が最大となるような関数 f を選んだときの値が Wasserstein 距離である.

第3章

機械学習・深層学習の基礎

3.1 機械学習の分類

機械学習手法の分類について述べる。

3.1.1 機械学習の大きな分類

機械学習は、大きく分けて教師あり学習，教師無し学習，強化学習の3つに大別される[27]。この3種類の概要は以下の通りである。

教師あり学習

入力 \mathbf{x} から出力 \mathbf{y} への対応関係を学習することが目標となる機械学習である。教師あり学習では，入出力の複数のサンプル集合 $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ が与えられる。 \mathcal{D} は訓練集合ともよばれ，訓練集合をもとにして未知の入力に対しても，正しい出力になるべく近い出力を達成する対応関係を学習する。 N は訓練集合の要素の数である*¹。

教師なし学習

与えられたデータから新しい知識となる「興味深いパターン」を抽出することが目標となる機械学習である。教師なし学習では入力のみで訓練集合 $\mathcal{D} = \{(\mathbf{x}_1), (\mathbf{x}_2), \dots, (\mathbf{x}_N)\}$ が与えられる。教師なし学習は，学習の目的が明確に定まっていないことが多く，自明な誤り指標がないために，教師あり学習に比べて多

*¹ 一般に，入力 \mathbf{x} から出力 \mathbf{y} の取り得る組み合わせすべてが訓練集合として与えられていることはなく，その組み合わせのうちいくつかを訓練集合として収集してきたと考える。この訓練集合からもとの取り得る組み合わせの情報を復元することを試みているのである。

様な問題設定がある。

強化学習

エージェント（例：ロボット）が試行錯誤を通じて報酬を最大化することが目標となる機械学習である。教師あり学習とは異なり、エージェントの個々の挙動 x に対して報酬 y が与えられる訳ではなく、連続的な試行錯誤の過程で報酬が与えられる。

3.1.2 教師あり学習の主な分類

教師あり学習は、主に回帰と分類の2つに大別される。

回帰 (Regression)

入力 x から出力 y への対応関係の予測において、出力 y が連続変数であるとき、これを回帰という。

分類 (Classification)

入力 x から出力 y への対応関係の予測において、出力 y がカテゴリ（離散変数）である場合、これを分類という。 y が2種類のカテゴリのいずれかに属するとき、これを2値分類という。また、それ以上の複数のカテゴリに属す時、これを多クラス分類という。

3.1.3 教師なし学習の主な例

教師なし学習には様々な種類があるが、その代表例として以下を挙げる。

クラスタリング

訓練集合 D の似た者同士を集めていくつかの種類にグループ分けを行う。

異常検出

訓練集合 D のの中から、平均的な傾向から逸脱したデータを発見する。

潜在的な要素の推定

訓練集合 D が高次元であるとき、データを低次元空間に落とし込み、最低限の情報で元データの特徴を表現する。

深層生成モデル

訓練集合 D を低次元の潜在変数と関連づけ、潜在変数から実データを確率的に生

成する。VAE[28] や GAN[29] などが該当する。

3.1.4 その他の分類方法 1 (パラメトリックモデルとノンパラメトリックモデル)

機械学習モデルが、訓練データ集合のサイズに依らず、固定の数の学習パラメータを持っているとき、これをパラメトリックモデルと呼ぶ。一方、訓練データのサイズに依存して学習すべきパラメータの数が増加する機械学習モデルをノンパラメトリックモデルと呼ぶ。パラメトリックモデル・ノンパラメトリックモデルの代表例を以下に述べる。

パラメトリックモデルの例

線形回帰モデル, ロジスティック回帰, 深層学習モデル, ナイーブベイズ

ノンパラメトリックモデルの例

カーネル SVM, 決定木, 近傍法

3.1.5 その他の分類方法 2 (生成モデルと識別モデル)

入力 x から出力 y への対応関係の予測において、 x と y の同時分布 $p(x, y)$ を学習するモデルを生成モデル、 x の下での y の条件付確率 $p(y|x)$ を学習するモデルを識別モデルという。識別モデル・生成モデルの代表例を以下に述べる

生成モデルの例

ナイーブベイズ, ベイジアンネットワーク, GAN や VAE などの深層学習モデル

識別モデルの例

ロジスティック回帰, 近傍法, 決定木, (普通の) 深層学習モデル

3.2 線形回帰モデル

機械学習手法のうち最も基本的なモデルである線形回帰モデルについて述べる。

3.2.1 線形回帰モデルとは

線形回帰とは、入力 \mathbf{x} と出力 y の間に線形的な関係があると仮定して、訓練集合 $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ から、線形モデル

$$y_i = \mathbf{w}^T \mathbf{x}_i = w_1(\mathbf{x}_i)_1 + w_2(\mathbf{x}_i)_2 + \dots + w_d(\mathbf{x}_i)_d \quad (3.1)$$

の未知パラメータ \mathbf{w} を推定し、未知入力に対する予測を行うモデルである。ここで、入力 \mathbf{x} は $d (\in \mathbb{N})$ 次元ベクトル、出力 y はスカラーである*2。

3.2.2 線形回帰モデル（特徴量を入れる）

線形回帰の入力として、直接訓練集合の入力ベクトル \mathbf{x} を入力するのではなく、入力ベクトル \mathbf{x} を特徴量ベクトル $\phi(\mathbf{x})$ に変換し、特徴量ベクトルに対する線形モデル

$$y_i = \mathbf{w}^T \phi(\mathbf{x}) \quad (3.2)$$

を考えることで、非線形的な関係性も表現することができる。ここで $\phi(\mathbf{x})$ はベクトル \mathbf{x} を入力にとってベクトルを出力する関数であり、入力に対して（一般に）非線形的な変換を施す。入力ベクトル \mathbf{x} が非線形変換されようが、パラメータ \mathbf{w} に対して線形になっていれば、線形回帰モデルである。

特徴量の例をいくつかあげると、関数

$$\phi(x) = (1, x) \quad (3.3)$$

を特徴量として用いた線形変換 $y = (0.5, 0.5)^T \phi(x)$ が挙げられる。これは単にパラメータとして入力に無関係な定数項*3を追加したものであり、積を展開すると

$$y = 0.5x + 0.5 \quad (3.4)$$

となっている。一般に線形回帰モデルといったときにはこの定数項は初めから導入されていることが多い。本論文では数式展開のしやすさから、必要であれば定数項は特徴量関数に含むとする。

もう一つの例として

$$\phi(x) = (1, x, x^2) \quad (3.5)$$

*2 式中の表記について、 $(\mathbf{x}_i)_j$ は、訓練集合 i 番目の組の入力ベクトルの j 番目の要素である。

*3 1 次関数の切片に相当する。

を特徴量として用いた線形変換 $y = (1, 2, 3)^T \phi(x) = 3x^2 + 2x + 1$ が挙げる。これは 2 次関数を含むが、パラメータに対しては線形なので線形回帰モデルを用いて表現できる。

3.2.3 線形回帰モデルの最適パラメータとは

簡単のため、当面の間入力 x を 1 次元ベクトル (= スカラー) とする。訓練集合 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ が与えられたとき、線形回帰モデル

$$y_i = wx_i \quad (3.6)$$

のパラメータ w を求めたい。最適なパラメータ w^* は、訓練集合の各データを精度良く予測しなければならない。仮に訓練集合も未知入力に対しても、厳密に誤差なく線形モデルを用いて表現できるならば、訓練集合のうちひとつを取ってきて

$$w^* = \frac{y_i}{x_i} \quad (3.7)$$

と計算すればよい^{*4}。しかしながら実際の訓練集合は、線形モデルでは厳密に表現することはできず、誤差を含んでいることが多い。これを数式で表すと、

$$y_i = w^* x_i + \epsilon_i \quad (3.8)$$

となる。ただし ϵ_i は i 番目の訓練集合の組が、線形モデルが想定する厳密な予測値 $y_i = w^* x_i$ からどれだけ外れているかの誤差を表す。誤差項を含む線形モデルは、パラメータ w をどのように選んでも誤差項 ϵ_i を

$$\epsilon_i = y_i - wx_i \quad (3.9)$$

と選ぶことで訓練集合に対する予測を一致させることができる。しかしながら、当然、未知入力に対して誤差項を知ることはできない。このような未知のデータセットに対する予測能力を汎化性能という。汎化性能を高めるためには、誤差項がないときの予測精度を高める必要がある。したがって、線形回帰モデルの最適パラメータ w^* としては、訓練集合における誤差項が総合的になるべく小さいものを選ぶことが望ましいと言える^{*5}。

^{*4} ただし $x_i \neq 0$ であるものに限る。

^{*5} もちろん、本当は未知入力に対しての誤差項を小さくしたいということであるが、未知入力に対しての誤差を知ることはできないため、訓練集合の誤差を代用するということである。

3.2.4 最小二乗法：線形モデルの最適パラメータの求め方

訓練集合に対するの予測値との誤差をなるべく小さくするような最適パラメータ w^* の求め方として、各訓練データの予測誤差 (x_i を入力としたときの線形回帰モデルの予測値 wx_i と訓練集合の値 y_i との差)

$$\epsilon_i = y_i - wx_i \quad (3.10)$$

の訓練集合全体での二乗和

$$\sum_{i=0}^N \epsilon_i^2 = \sum_{i=1}^N (y_i - wx_i)^2 \quad (3.11)$$

を最小化するように選んだパラメータ w^* を最適なパラメータとする求め方がある。これを最小二乗法という。数式で書けば、

$$w^* = \operatorname{argmin}_w \sum_{i=1}^N (y_i - wx_i)^2 \quad (3.12)$$

となる。

ここで、誤差の二乗和をパラメータ w の関数と考えれば、

$$L(w) = \sum_{i=1}^N (y_i - wx_i)^2 \quad (3.13)$$

という関数 $L(w)$ を考えることができる。このようなパラメータに対してその良さの指標を与える関数を、機械学習の文脈では損失関数という。訓練集合への予測誤差が小さいほど、損失関数の値は小さくなる。

では、損失関数 $L(w)$ を最小化する w はどのようになるだろうか。線形回帰モデルにおいて最小二乗法に基づく損失関数 $L(w)$ は、パラメータ w に対して下に凸の凸関数であるから、損失関数 $L(w)$ の微分がゼロとなるときに $L(w)$ が最小となり、そのときの w が最適パラメータである。すなわち、

$$\frac{d}{dw} L(w^*) = 0 \quad (3.14)$$

が成り立つ。これを解くと、

$$0 = \frac{d}{dw} L(w^*) \quad (3.15)$$

$$= \left[\frac{d}{dw} \sum_{i=1}^N (y_i - wx_i)^2 \right] \Big|_{w=w^*} \quad (3.16)$$

$$= - \sum_{i=1}^N 2x_i (y_i - w^* x_i) \quad (3.17)$$

$$\Leftrightarrow w^* = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2} \quad (3.18)$$

となり線形回帰モデルの最小二乗法による最適パラメータの解析解が求まった。

3.2.5 最尤推定

線形回帰モデルの学習には、最小二乗法を使用するのが一般的である。誤差の評価法としては最小二乗法でなくても、誤差の絶対値の和や誤差の4乗などでも学習はできるはずである。最小二乗法が選ばれる理由の一つとして、線形回帰モデルの最尤推定が最小二乗法による解に一致することがある。最尤推定は線形回帰モデルに限らずモデルパラメータを決定する一般的な手法であるから、最小二乗法を用いることに一定の根拠を与える。そこで、最尤推定から最小二乗法を導出する。

線形回帰モデルのパラメータを最尤推定によって決定するためには、前章で見た通り、尤度が最大となるようなパラメータ w を最適パラメータと決定する。すなわち、

$$w^* = \underset{w}{\operatorname{argmax}} \log p(\mathcal{D}|w) \quad (3.19)$$

である。ここで線形回帰モデルは x と y を

$$y_i = wx_i + \epsilon_i \quad (3.20)$$

と関連づけるものであった。ここで誤差 ϵ_i について、データ間で独立な平均ゼロ・分散 σ^2 のガウス分布を仮定すると

$$p(y_i|x_i, w) = \mathcal{N}(y_i|wx_i, \sigma^2) \quad (3.21)$$

かつ

$$p(y_1, y_2, \dots, y_N|x_1, x_2, \dots, x_N, w) = \prod_{i=1}^N p(y_i|x_i, w) \quad (3.22)$$

と定式化することができる。前者は線形回帰モデルの識別モデルとしての条件付確率を定義しており、後者はデータ間の独立性を表現している。したがって、パラメータ w の下での訓練集合 D の尤度は、

$$p(\mathcal{D}|w) = p(y_1, y_2, \dots, y_N, x_1, x_2, \dots, x_N|w) \quad (3.23)$$

$$= p(y_1, y_2, \dots, y_N|x_1, x_2, \dots, x_N, w)p(x_1, x_2, \dots, x_N|w) \quad (3.24)$$

$$= p(x_1, x_2, \dots, x_N|w) \prod_{i=1}^N p(y_i|x_i, w) \quad (3.25)$$

$$= p(x_1, x_2, \dots, x_N|w) \prod_{i=1}^N \mathcal{N}(y_i|wx_i, \sigma^2) \quad (3.26)$$

となる。よって対数尤度は

$$\log p(\mathcal{D}|w) = \sum_{i=1}^N \mathcal{N}(y_i|wx_i, \sigma^2) + C \quad (3.27)$$

$$= \sum_{i=1}^N -\frac{1}{2\sigma^2}(y_i - wx_i)^2 + C \quad (3.28)$$

となる。ただし、 w に関係のない定数項は C として省いた。ここで最小二乗法による損失関数（再掲）

$$L(w) = \sum_{i=1}^N (y_i - wx_i)^2 \quad (3.29)$$

と比較すると、定数項および定数倍の違いしか見られないことが分かる。すなわち、

$$w^* = \operatorname{argmax}_w \log p(\mathcal{D}|w) = \operatorname{argmin}_w L(w) \quad (3.30)$$

を導く。これで線形回帰モデルの最尤推定が最小二乗法による解に一致することが証明された。

3.2.6 多次元の線形回帰モデル

前節まで、入力が1次元の線形回帰モデルについて説明した。入力が多次元ベクトルである線形回帰モデルについても同様な議論ができる。すなわち、高次元の線形回帰モデル

$$y_i = \mathbf{w}^T \mathbf{x}_i \quad (3.31)$$

に対して，訓練集合 $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ が与えられたとき，予測誤差の二乗和

$$L(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \quad (3.32)$$

を損失関数として，損失関数が最小となるような最適パラメータ \mathbf{w}^* を求めればよい．入力が多次元であっても，最尤推定は上記の最小二乗法を導くことが知られている．

最小二乗法による最適パラメータ \mathbf{w}^* は，訓練集合から行列

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad (3.33)$$

を構成したとき，方程式

$$\mathbf{X}^T \mathbf{X} \mathbf{w}^* = \mathbf{X}^T \mathbf{y} \quad (3.34)$$

を満たす．この方程式は正規方程式という．この方程式は明らかに， $(\mathbf{X}^T \mathbf{X})$ が正規行列であれば $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ と解ける．

正規方程式の導出を次に示す．多次元入力の線形回帰モデルの損失関数 $L(\mathbf{w})$ は下に凸であるから，損失関数が最小となるのは，

$$\nabla_{\mathbf{w}} L(\mathbf{w}^*) = 0 \quad (3.35)$$

である．よって，

$$0 = \nabla_{\mathbf{w}} L(\mathbf{w}^*) \quad (3.36)$$

$$\Leftrightarrow 0 = \nabla_{\mathbf{w}} \left(\sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \right) \Big|_{\mathbf{w}=\mathbf{w}^*} \quad (3.37)$$

$$\Leftrightarrow 0 = \sum_{i=1}^N \nabla_{\mathbf{w}} ((y_i - \mathbf{w}^T \mathbf{x}_i)^2) \Big|_{\mathbf{w}=\mathbf{w}^*} \quad (3.38)$$

$$\Leftrightarrow 0 = \sum_{i=1}^N -\mathbf{x}_i (y_i - \mathbf{w}^{*T} \mathbf{x}_i) \quad (3.39)$$

$$\Leftrightarrow \sum_{i=1}^N \mathbf{x}_i (\mathbf{w}^{*T} \mathbf{x}_i) = \sum_{i=1}^N y_i \mathbf{x}_i \quad (3.40)$$

$$\Leftrightarrow \mathbf{X}^T \mathbf{X} \mathbf{w}^* = \mathbf{X}^T \mathbf{y} \quad (3.41)$$

$$(3.42)$$

が成り立ち証明された。最後のベクトル和表現から行列積表現への変換は一見ややこしいが、要素ごとに展開すれば一致していることが確認できる。

3.3 深層学習の基礎

以降に用いる深層学習モデルの基礎について述べる。

3.3.1 深層順伝播型ニューラルネットワーク

深層順伝播型ニューラルネットワークとは、入力 \mathbf{x} と出力 \mathbf{y} の間の関係

$$\mathbf{y} = f(\mathbf{x}; \theta) \quad (3.43)$$

を予測するモデルのひとつである [30]。ニューラルネットワークは、予測モデルが複数の関数 f_1, f_2, f_3 を用いて

$$\mathbf{y} = f_3(f_2(f_1(\mathbf{x}))) \quad (3.44)$$

のようにさまざまな関数を組み合わせる形で構成されており、これがネットワークと呼ばれる理由である。これらの関数は必ず入力 \mathbf{x} から出力 \mathbf{y} へ一方向へ流れるように接続されており（有向非巡回グラフ）、これが「順伝播」の意味である。図 3.1(a) のような基本形ニューラルネットワークにおいて、一番初めの関数 f_1 を第 1 層（第 1 レイヤー）、つづけて f_2 を第 2 層（第 2 レイヤー）、 f_3 を第 3 層（第 3 レイヤー）のように呼ぶ。層の数を層の「深さ」とよび、線形回帰などの他のモデルとくらべて、ニューラルネットワークは層を重ねて深くしていくため、これが「深層」の意味である。有向非巡回グラフであれば図 3.1(b) のようなつなぎ方であってもよい。図 3.1(c) は、有向であるが非巡回ではない（巡回グラフ）であるので、不適格である。

ニューラルネットワークの各関数による出力を分解してみると、

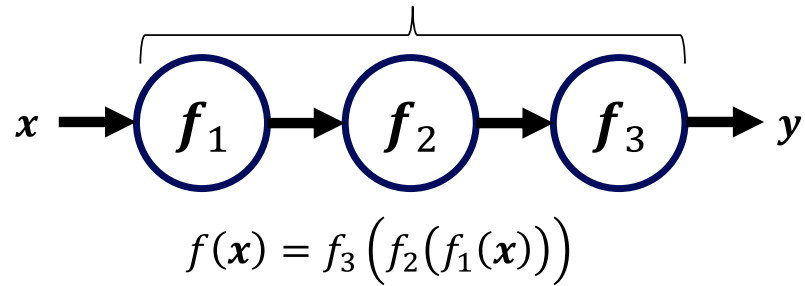
$$\mathbf{h}_1 = f_1(\mathbf{x}) \quad (3.45)$$

$$\mathbf{h}_2 = f_2(\mathbf{h}_1) \quad \mathbf{y} = f_3(\mathbf{h}_2) \quad (3.46)$$

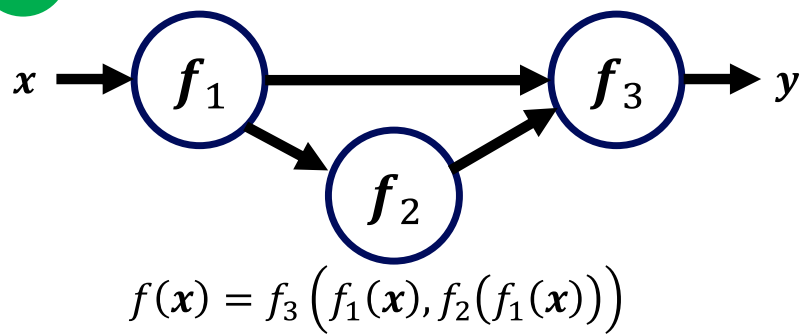
となる。上式の通り、最後の層は出力 \mathbf{y} の予測に直接的に貢献しており、出力層と呼ばれる。一方、出力層より手前の残り 2 つの層は、出力層への入力を変化させることによって出力に対して間接的に影響をおよぼしており、隠れ層と呼ばれる。隠れ層の出力 $\mathbf{h}_1, \mathbf{h}_2$ を隠れ変数とよぶ。第一層への入力は \mathbf{x} であり、出力層の出力は出力 \mathbf{y} であるから第一層の入力変数の次元と出力層の出力変数の次元は訓練集合 \mathcal{D} によって決定される。一方、


(a) 

3 layers



(b) 



(c) 

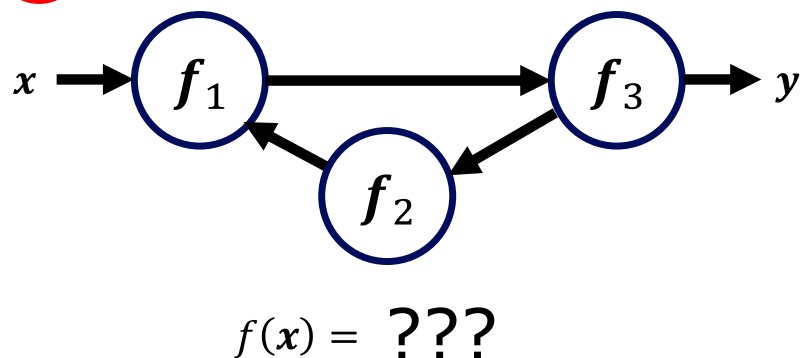


図 3.1 ニューラルネットワークの構造を表すグラフ. (a) 順に層を適用していくニューラルネットワーク (b) 分岐をもつニューラルネットワーク (c) 巡回する構造を持っているため、ニューラルネットワークではないグラフ

隠れ層の出力 $\mathbf{h}_1, \mathbf{h}_2$ の次元は、訓練集合 \mathcal{D} とは無関係である。したがって、ニューラルネットワークのモデルアーキテクチャの自由度として、モデルの使用者が事前に決める必要がある。隠れ層の出力の次元を、「深さ」との対比で）層の幅と呼ぶ。

3.3.2 特徴量の学習：線形回帰モデルとの違い

線形回帰モデルは、

$$y = \mathbf{w}^T \mathbf{x} \quad (3.47)$$

のように入力 \mathbf{x} を線形変換することで出力 y を予測するモデルであった。また、線形でない訓練集合 \mathcal{D} にも対応するために、入力 \mathbf{x} を非線形変換しておいて、非線形変換された特徴量 $\phi(\mathbf{x})$ を基底にとる

$$y = \mathbf{w}^T \phi(\mathbf{x}) \quad (3.48)$$

を考えることにより線形回帰モデルの表現力を高めたのであった。言い換えれば、訓練集合 \mathcal{D} の性質に応じて、特徴量 $\phi(\mathbf{x})$ を設計する必要があるということである。この設計方法として、次のように考えることができる。

- 非常に沢山の多様な特徴量を設定して、どんな訓練集合が与えられても有用な特徴量を含むように構成すること。実際に、カーネル回帰とよばれる方法では無限個の特徴量を用意して表現力を高めている。
- 訓練集合に合わせて手動で有用な特徴量を設計すること。このやり方は分野ごとの深い専門知識をもつ専門家が長い時間をかけて設計する必要があるが、深層学習モデル登場以前では主要な方法であった。
- そして3つ目は、特徴量そのものを学習によって獲得する方法である。ニューラルネットワークは、隠れ層が特徴量を学習する。各層の関数はそれぞれ学習によって更新できるパラメータを持ち、訓練集合が与えられることによって出力層のみならず隠れ層（特徴量）も学習によって自動的に獲得される。たとえば、図 3.1(a) のニューラルネットワークにおいて $f_3(\mathbf{h}_2; \mathbf{w}) = \mathbf{w}^T \mathbf{h}_2$, $f_2(f_1(\mathbf{x}; \theta_2); \theta_1) = \phi(\mathbf{x}, \theta_2, \theta_1)$ とおけば、次のようになる。

$$f(\mathbf{x}) = f_3(f_2(f_1(\mathbf{x}, \theta_1), \theta_2), \theta_3) = \mathbf{w}^T \phi(\mathbf{x}, \theta_2, \theta_1) \quad (3.49)$$

線形回帰モデル（式 3.1）と比較すると、このニューラルネットワークモデルには特徴量 $\phi(\mathbf{x})$ にも学習可能なパラメータ θ_2, θ_1 があることが分かる。深層学習で

は、特徴量もパラメータをもち、訓練集合に基づいて適切な特徴量の獲得を目指すものであることが分かる。

3.3.3 ニューラルネットワークに使用される具体的な関数

ニューラルネットワークの予測モデルを具体的に与えるためには、各層 f_1, f_2, f_3 が具体的に与えられている必要がある。最も基本的な層のひとつとして全結合層がある。全結合層は、

$$f(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{W}\mathbf{x} + \mathbf{b} \quad (3.50)$$

と与えられる。学習パラメータとして $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}$ をもつ。全結合層は特徴量を用いない線形回帰モデルと同じ形式をしている。ただし、バイアス項 \mathbf{b} が明示的に与えられている。

全結合層の中身は、上記のとおりであるが、この層を重ねてニューラルネットワークを実現するには、ひとつだけ問題点がある。それは、各層が全結合層であるとき、各層は線形変換であるため、層をいくつ重ねても線形変換であることには変わりはないから、線形的な関係しか表現することができないという問題があることである。線形回帰モデルは特徴量として様々な非線形変換を用いることで表現力を上げたわけであるから、隠れ層は非線形関係を表現できる必要がある。

隠れ層が非線形変換を獲得するために、出力層を除すべての全結合層には線形変換後に活性化関数と呼ばれる非線形関数 $\sigma(\cdot)$ を適用し、

$$f(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (3.51)$$

とする。この非線形関数 $\sigma(\cdot)$ として近年汎用的に用いられているのが、図 3.2 に示す ReLU(Rectified Linear Unit) 関数

$$\text{ReLU}(x) = \max(0, x) \quad (3.52)$$

である。図 3.2 からわかる通り、この関数はこの関数は入力が正ならそのままの値を返し、ゼロ以下ならゼロとする非線形関数である。ただし、活性化関数への入力が多次元ベクトルの場合は、ベクトルの要素ごとに活性化関数を適用したものを出力とする^{*6}。ReLU 関数は非線形変換としては非常に頼りなく思えるが、普遍性定理とよばれる数学的証明に

^{*6} 深層学習において、1 変数の活性化関数をまず与えて、多変数への拡張として要素ごとに活性化関数を適用することが一般的である

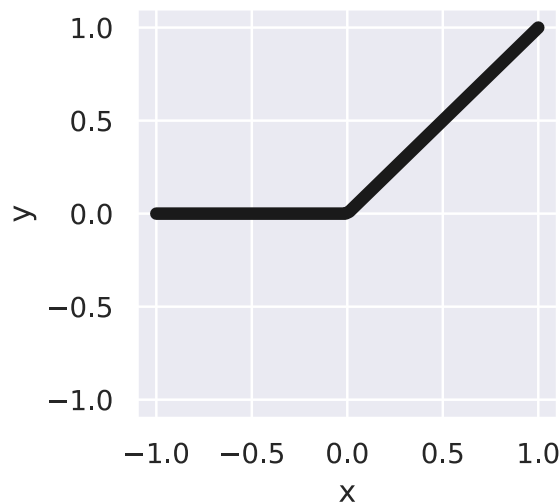


図 3.2 ReLU 関数.

よって、隠れ変数の次元が十分に大きい場合、2 層以上のニューラルネットワークはどんな関数も表現できることが知られている。

3.3.4 ニューラルネットワークの学習：最急降下法

ニューラルネットワークの最適パラメータは、線形回帰モデルと同様に最小二乗法により求めることができる。すなわち、ニューラルネットワークによる出力を

$$f(\mathbf{x}; \boldsymbol{\theta}) \tag{3.53}$$

としたとき、訓練集合に対する予測誤差の二乗和

$$L(\boldsymbol{\theta}) = \sum_{i=1}^N \|\mathbf{y}_i - f(\mathbf{x}_i; \boldsymbol{\theta})\|^2 \tag{3.54}$$

を損失関数として、損失関数が最小となるようなパラメータを見つけたい。ここでニューラルネットワークの全パラメータを $\boldsymbol{\theta}$ としてまとめた。

損失関数 $L(\boldsymbol{\theta})$ を最小化するパラメータ $\boldsymbol{\theta}$ を求めるために、線形回帰モデルでは最適パラメータの解析解を求めることができたが、ニューラルネットワークは損失関数がパラメータに対して非線形であり、一般に解析的に最適パラメータ求めることはできない。そこでニューラルネットワークでは、最急降下法を用いて損失関数が小さくなる方向を探索する。

最急降下法は、(最適でない)パラメータの初期値 θ_0 から始めて、損失関数の値を少し低下させるパラメータに更新していくプロセスを繰り返し行うことで最終的に最適なパラメータに到達する反復法の一つであり、更新の方法として勾配を用いる手法である。いま、 k 回目の反復プロセスにおいて、パラメータ θ_k を θ_{k+1} に更新することを考える。最急降下法の更新の計算式は

$$\theta_{k+1} = \theta_k - \mu \nabla_{\theta} L(\theta_k) \quad (3.55)$$

となる。すなわち最急降下法は、損失関数の勾配の逆方向へ少しずつ進むことによって学習するアルゴリズムである。ここで、式中の変数 μ は学習率とよばれ、最適パラメータへの収束の速さや計算の安定性を変化させる。

最急降下法を用いた場合、適切に学習率を設定すれば、損失関数が凸関数である限り、最適値に1次収束することが知られている。ただし、ニューラルネットワークの損失関数は一般に非凸関数となり、最急降下法による最適値への到達は保証されず、局所最適解への到達のみ保証される。ところが、近年の研究によれば、過剰なパラメータをもつニューラルネットワークは大域最適解に到達できることが発見されている。

加えて、最急降下法では損失関数のパラメータに対する勾配 $\nabla_{\theta} L(\theta)$ を計算する必要がある。ニューラルネットワークは関数が複雑に合成されているため、コンピュータによる自動微分を用いることが一般的である。

3.3.5 ニューラルネットワークの学習：オンライン学習・ミニバッチ学習・バッチ学習

最急降下法では、訓練集合すべての組に対する二乗誤差を総和した損失関数を用いて、最適化を行った。すなわち、

$$\theta_{k+1} = \theta_k - \mu \nabla_{\theta} \sum_{i=1}^N \|y_i - f(x_i; \theta)\|^2 \quad (3.56)$$

である。このように1度のパラメータ更新のために訓練集合すべてを用いる手法をバッチ学習とよぶ。一方、訓練集合すべてを用いるのではなく(あるいはすべてを用いる準備ができないとき)、データの組を一つ取り出して1度のパラメータ更新を行う手法をオンライン学習とよぶ。すなわち、オンライン学習ではある i について

$$\theta_{k+1} = \theta_k - \mu \nabla_{\theta} \|y_i - f(x_i; \theta)\|^2 \quad (3.57)$$

のように更新する。また、その中間として、訓練集合のうちランダムに一部だけを取り出して、勾配を計算しパラメータを更新するミニバッチ学習がある。

一度の更新で用いるデータの組が少ないとき、勾配の計算コストが小さくなるおかげで高速に学習が進行する一方、勾配のゆらぎが大きくなることにより学習が不安定になる(揺らぎが大きい)。反対に一度の更新で用いるデータの組が多いとき、揺らぎが小さくなることによって学習が安定するが、計算コストの増大により学習が遅くなる。ただし、大規模並列計算によって、計算時間を大幅に減少させることが可能である。

3.3.6 ニューラルネットワークの学習：ハイパーパラメータ

ニューラルネットワークには、訓練集合を用いた学習によって更新可能なパラメータに加えて、学習開始前に設計者が設定しなければならないハイパーパラメータがある。ハイパーパラメータの例としては、層の数、各隠れ層の幅(隠れ変数の次元)、学習率などがある。

ハイパーパラメータの選び方によって、ニューラルネットワークの汎化性能は大きく異なることがある。これらのハイパーパラメータは手動でいくつかの条件を試し適切なもの決定することもできるが、ランダムサーチ・グリッドサーチによる自動的なハイパーパラメータの探索や、他のハイパーパラメータ最適化ライブラリを用いることもできる。

3.4 深層生成モデルの基礎

以降に用いる2つの主要な深層生成モデルについて述べる。

3.4.1 敵対的生成ネットワーク (GANs)

GAN[29]は深層生成モデルの1種である。深層生成モデルは、訓練集合が与えられたとき、訓練集合を生成したと考えられるデータ源の従う確率分布を推定して、推定されたデータ源からサンプルを生成することを目的とする。GANは、この目的を達成するために、サンプルを生成するニューラルネットワーク G と、 G によって生成されたサンプルがデータ源の従う確率分布と一致しているかを比較するニューラルネットワーク D の2つのネットワークを用いる。前者のニューラルネットワーク G はジェネレータと呼ばれ、後者のニューラルネットワーク D はディスクリミネータと呼ばれている。これら2つのネットワークに対して、ディスクリミネータを、ジェネレータの生成したサンプル集合の従う確率分布をと訓練集合の従う確率分布がどれだけ異なっているかをより精度良く識別

する方向に、ジェネレータを、ディスクリミネータができるだけ識別しにくくなるように学習する。学習の結果、最終的にはジェネレータから得られるサンプル集合がデータ源の従う確率分布と一致することが期待される。

Goodfellow によって提案された最も初期の GAN は、次のように構築される。まず、潜在変数 z を入力にとって訓練集合と同じ次元を持つ変数 x を出力するニューラルネットワークを作り、ジェネレータ $x = G(z)$ とする。ここで、潜在変数 z 上の確率変数 Z を考えて、確率変数 Z の確率分布 $p(z)$ に従うサンプル集合を、それぞれニューラルネットワーク $x = G(z)$ に適用して得られるサンプル集合を、ジェネレータによって生成されたサンプル集合と考える。潜在変数上の確率分布 $p(z)$ は、通常無相関正規分布が用いられる。一方、訓練集合と同じ次元を持つ変数 x を入力にとって、実数 y を出力するニューラルネットワークを作り、ディスクリミネータ $y = D(x)$ とする。ディスクリミネータ D には、予め用意した訓練集合のサンプルか、あるいはジェネレータによって生成されたサンプルのどちらが入力され、出力として入力されたサンプルが訓練集合のサンプルであると判定する確率を出力する。ジェネレータはディスクリミネータが訓練集合のサンプルであるとジェネレータによるサンプルを誤判定する方向に、ディスクリミネータは各サンプルを正しく判定する方向に学習を進めるため、クロスエントロピー誤差関数に基づく次の損失関数

$$L(\theta_D, \theta_G) = -\mathbb{E}_T[\log D(T)] - \mathbb{E}_Z[\log(1 - D(G(Z)))] \quad (3.58)$$

を、ジェネレーターは最小化、ディスクリミネータは最大化するように学習する。ただし T は訓練集合の経験分布に従う確率変数である（訓練集合からランダムにひとつを選ぶような確率）。学習は通常のニューラルネットワークと同じく勾配法を用いるが、2つのニューラルネットワークが真逆の損失関数を用いて学習するところが特別に異なっている。

しかしながら、上記の定式化による GAN は、学習が不安定になることが指摘されている [31, 32]。このため、上記の GAN の安定性を改善する様々な先行研究が提案されている [33, 34]。特に、文献 [31] では、ジェネレーターによるサンプル集合と訓練集合の台が一致しないとき、ジェネレーターへ損失関数の勾配がゼロとなり学習が進行しない状態になることが理論的に指摘されている。本研究では、GAN の安定性を向上するために、文献 [31] で提案された Wasserstein GAN とよばれる種類の GAN を用いることにする。Wasserstein GAN では、ディスクリミネータの損失関数としてクロスエントロピー誤差を最小化するの代わりに、ジェネレータによって生成されたサンプル集合と訓練集合間の Wasserstein 距離を求めるように変更する。

Wasserstein 距離を求める線形計画問題の双対形式 (2 章を参照) から, Wasserstein 距離は次のように計算できる.

$$W = \max_{\|D\|_L \leq 1} \mathbb{E}_T[D(T)] - \mathbb{E}_Z[D(G(Z))]. \quad (3.59)$$

$\|D\|_L \leq 1$ は関数 D が 1-リプシッツ連続であることを表しているから, 関数 D をニューラルネットワークで表現したとき, 関数 D が 1-リプシッツ連続となるように制約して, 上式を最大化するように学習すれば, Wasserstein 距離を求めることができる. 関数 D が 1-リプシッツ連続となるような制約は, 次の One-Centered Gradient Penalty[35]

$$L_{\text{PEN}}(\boldsymbol{\theta}_D) = \mathbb{E}_U[(\|\nabla_U D(U)\| - 1)^2] \quad (3.60)$$

を最小化するように学習することで達成できる. ここで U は 0 から 1 までの一様分布を確率分布にもつ実数確率変数 τ を用いて

$$U = \tau T + (1 - \tau)G(Z) \quad (3.61)$$

と計算される確率変数であり, ジェネレータによって生成されたサンプルと訓練集合 (のうちの 1 つのランダムサンプル) を一様ランダムに内分することで得られるサンプルである. 従って結局, Wasserstein GAN の学習は, ディスクリミネータの損失関数として

$$L_D(\boldsymbol{\theta}_D) = -\mathbb{E}_T[D(T)] + \mathbb{E}_Z[D(G(Z))] + \lambda L_{\text{PEN}}(\boldsymbol{\theta}_D) \quad (3.62)$$

を最小化するように, またジェネレータの損失関数として

$$L_G(\boldsymbol{\theta}_G) = -\mathbb{E}_Z[D(G(Z))] \quad (3.63)$$

を最小化するように進めることになる. ただし λ は One-Centered Gradient Penalty の強さを決定する正の実数であり, ハイパーパラメータの一つになる.

3.4.2 Continuous Normalizing Flow(CNF)

CNF[36] は, 深層生成モデルの一種である. CNF は, ある 1 階微分方程式が与えられ, 入力となる確率分布に従うサンプルを微分方程式の初期値 $x(t=0)$ として, これを時間積分した結果 $x(t=T)$ を出力サンプルとするものである. 前述の GAN と異なる点として, CNF は入力となるサンプルの従う確率分布の値が既知であれば, 出力となるサンプルの従う確率分布の値を計算できるという性質を持つ. GAN はジェネレータのサンプルが, 訓練集合の従うサンプルの確率分布になるべく近い確率分布に従うように学

習をしたが、ジェネレーターのサンプルが厳密にどのような確率分布に従うかどうかについては計算しないのであった。従って、CNF では、出力サンプルが従う確率分布の値を厳密に計算する必要がある場合に有効である。

数式を用いて定式化すると、次のようになる。あるリプシッツ連続関数 \mathbf{f} があって、次の 1 階微分方程式

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) \quad (3.64)$$

を考える。この微分方程式の $t = 0$ における初期値を、ある確率分布 $p(\mathbf{x}(t = 0))$ に従うサンプルとして、それぞれのサンプルを $t = T$ まで時間積分する。このとき、得られるサンプル集合が従う確率分布 $p(\mathbf{x}(t))$ は、

$$\frac{\partial \log p(\mathbf{x}(t))}{\partial t} = -\text{tr} \left(\frac{d\mathbf{f}}{d\mathbf{x}(t)} \right) \quad (3.65)$$

に従う時間変化をすることが示せる。すなわち、確率分布 $p(\mathbf{x}(t = 0))$ に従うサンプルを $t = T$ まで時間積分したときの確率分布を陽に表示すれば、

$$p(\mathbf{x}(t)) = \exp \left[- \int_{t=0}^T \text{tr} \left(\frac{d}{d\mathbf{x}(t)} \mathbf{f} \right) dt \right] p(\mathbf{x}(0)) \quad (3.66)$$

となる。従って、式 3.64 の時間積分と同時に上式も数値積分することによって、得たサンプルが従う確率分布の値を求めることができる。

式 3.65 の確率分布の時間変化は、次の通り証明できる。まず、ある確率分布 $p(\mathbf{x})$ に従う確率変数を、逆関数が存在する関数 $\mathbf{y} = \mathbf{g}(\mathbf{x})$ によって変換したとき、確率分布 $p(\mathbf{y})$ は

$$p(\mathbf{y}) = \det \left(\frac{d}{d\mathbf{x}} \mathbf{f} \right)^{-1} p(\mathbf{x}) \quad (3.67)$$

に従う。いま、CNF における時間 t から $t + dt$ への変化を考えると、状態 $\mathbf{x}(t)$ は

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + dt \mathbf{f}(\mathbf{x}) \quad (3.68)$$

に従う変換を受けているから、その確率変化は

$$p(\mathbf{x}(t + dt)) = \det \left(\frac{d}{d\mathbf{x}(t)} \mathbf{x}(t + dt) \right)^{-1} p(\mathbf{x}(t)) \quad (3.69)$$

$$= \det \left(I + dt \frac{d}{d\mathbf{x}(t)} \mathbf{f} \right)^{-1} p(\mathbf{x}(t)) \quad (3.70)$$

$$= \text{tr} \left(\frac{d}{d\mathbf{x}(t)} \mathbf{f} \right)^{-1} p(\mathbf{x}(t)) dt \quad (3.71)$$

と計算できる. ただし dt^2 以降の項は微小量として無視しており, かつ f がリプシッツ連続であるから, この状態変化には逆変換が存在することを利用している. これは直ちに式 3.65 を導く. よって証明できた.

第 4 章

分子シミュレーションの基礎

分子シミュレーションは、大きく分けて MC シミュレーションと MD シミュレーションに大別される。それぞれの手法についての基礎的な知識を述べる。

4.1 MC シミュレーション

分子シミュレーション手法の 1 つである MC シミュレーションについて述べる。

4.1.1 ボルツマン分布

統計力学の理論では、平衡状態にある系について、任意の状態が実現する確率は圧力や温度などの巨視的な物理量のみによって決定され、この確率分布をボルツマン分布という [37]。MC シミュレーションは、ボルツマン分布に従うように状態をサンプリングすることで、系が平衡状態であるときの状態の統計的性質を調べるシミュレーションを行う手法である。

たとえば粒子数 (N)・温度 (T)・体積 (V) が一定となるカノニカルアンサンブルでは、ある状態 \mathbf{r} が実現される確率 $p_B(\mathbf{r})$ は次のボルツマン分布

$$p_B(\mathbf{r}) \propto \exp[-\beta U(\mathbf{r})] \quad (4.1)$$

に従う。ここで、 β は系の逆温度、 $U(\mathbf{r})$ は状態 \mathbf{r} が与えられたときの総ポテンシャルエネルギーである。式からわかる通り、ある状態が実現される確率は、その状態が持つエネルギーの増加に対して指数的に小さな確率が与えられる。また、温度は高ければ高いほど確率の指数的な減少を低減していることになるから、温度が高いということはその分だけポテンシャルエネルギーが高い状態も許容されるということである。一般に、ポテンシ

ルエネルギーが小さい状態の数よりも、それよりエネルギーが高い状態の数の方が多いから、温度が高い状態では温度がそれより低いときに良く出現していた状態は、比率的に出現しなくなるだろう。

また、粒子数の代わりにその共役量である化学ポテンシャルを一定とする、化学ポテンシャル (μ) ・温度 (T) ・体積 (V) が一定となるグランドカノニカルアンサンブルでは、ある状態 \mathbf{r} が実現される確率 $p(\mathbf{r})$ は次のボルツマン分布

$$p_B(\mathbf{r}) \propto \exp[-\beta U(\mathbf{r}) + \mu N] \quad (4.2)$$

に従う。カノニカルアンサンブルのボルツマン分布と比較して、粒子数に関するエネルギーが指数部に追加されている。

4.1.2 メトロポリス・ヘイスティング (MH) 法

MC シミュレーションにおいてボルツマン分布をサンプリングする場合、ボルツマン分布は分布の式が解析的に与えられているだけであり、シミュレーションにはボルツマン分布からサンプリングを実行するアルゴリズムが必要である。そのようなアルゴリズムとして基礎的かつ一般的に用いられている手法がメトロポリス・ヘイスティング (MH) 法 [38] である。MH 法は、マルコフ過程に従う状態列を考えて、その状態列の定常状態が目的の分布へ一致するようにマルコフ過程を設定し、そのマルコフ過程をシミュレーションすることでサンプリングを行う。

マルコフ過程に基づくサンプリングの理論的な背景について述べる。状態 x にもとづき次の状態 x' をサンプリングする (1 階斉時) マルコフ過程 $p(x'|x)$ について

$$p(x'|x)p_B(x) = p(x|x')p_B(x') \quad (4.3)$$

が成り立つとき、このマルコフ過程は定常状態がボルツマン分布 p_B に一致する。この式は詳細釣り合い条件と呼ばれており、マルコフ過程が詳細釣り合い条件を満たすように設計することで、このマルコフ過程から得られるサンプル列はボルツマン分布 p_B に従うようになる。ただし、詳細釣り合い条件はマルコフ過程は定常状態がボルツマン分布に一致するための十分条件であって必要条件ではないという点に注意する。詳細釣り合い条件を満たさないマルコフ過程の設計法としてはいくつかの研究が知られている [39, 40, 41]。

MH 法では、上記詳細釣り合い条件を満たすマルコフ過程 $p(x'|x)$ を構成するために、提案分布による候補状態のサンプリングと、候補状態の棄却採択という 2 つのプロセスを導入している。まず、提案分布は、最新の状態 x に依存して (あるいは依存してなくて

もよい) 候補状態 x^* を生成する. この確率分布を $q(x^*|x)$ と書くことにする. また, 提案分布は目的の分布とは全く異なってもよい. 次に, 得た候補状態 x^* を採択率

$$A(x, x^*) = \min \left(\frac{p(x^*) q(x|x^*)}{p(x) q(x^*|x)}, 1 \right) \quad (4.4)$$

で採択して, マルコフ過程の次状態 x' とする. ただし採択されなかった場合は現状態 x を次状態 x' とする. 候補状態のサンプリングと棄却採択を統合的にみると, 詳細つり合い条件を満たしているため, MH 法は目的の分布をサンプリングできる. MH 法の効率は, 提案分布の選び方に大きく依存している. ここで, MC シミュレーションの効率とは, いかにか高速にボルツマン分布からの独立なサンプリングを集められるかどうかということである. 詳しくは MC シミュレーションに関する研究の 7・8 章にて述べている.

4.2 MD シミュレーション

分子シミュレーション手法の 1 つである MD シミュレーションについて述べる.

4.2.1 運動方程式と数値積分

MD シミュレーションは, 分子間に働く力に従ってニュートンの運動方程式を数値的に解くことで, 分子の運動をシミュレーションする方法である. MC シミュレーションは統計学的に平衡状態にある系の状態を調べることができたが, MD シミュレーションは系のダイナミクスを直接シミュレーションすることによって, 系の動力的な性質や非平衡状態における挙動を調べることができる. 具体的には, 状態として系の全粒子の位置 $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ について次の 2 階微分方程式

$$m_i \frac{\partial^2}{\partial t^2} \mathbf{r}_i = \mathbf{f}_i \quad (4.5)$$

を数値的に解く. ここで N は系の全粒子数, m_i は粒子 i の質量, \mathbf{f}_i は粒子 i に働く力 (合力) である. 粒子 i に働く力は, これが保存力であれば, 保存力に対応する状態 \mathbf{r} に対するポテンシャルエネルギー $U(\mathbf{r})$ を用いて

$$\mathbf{f}_i(\mathbf{r}) = -\nabla_{\mathbf{r}_i} U(\mathbf{r}) \quad (4.6)$$

と計算できる.

本研究では, ニュートン運動方程式の数値積分のために専用のソフトウェアを用いているが, これらのソフトウェアがニュートンの運動方程式の数値積分を実行するのに, 蛙飛

び法や速度ベルレ法が用いられる。これは、蛙飛び法や速度ベルレ法がシンプレティック性と呼ばれる性質を持つことによる。シンプレティック性を持つ時間積分アルゴリズムでは、時間発展に伴う総エネルギーの誤差が時間に依存しないため、長期シミュレーションに都合が良い。蛙飛び法による数値積分の1サイクルは次のように計算される。

$$\mathbf{r}_i(t) = \mathbf{r}_i(t - \Delta t) + \mathbf{v}_i(t - \Delta t/2)\Delta t \quad (4.7)$$

$$\mathbf{v}_i(t + \Delta t/2) = \mathbf{v}_i(t - \Delta t/2) + \mathbf{f}_i(\mathbf{r}(t))\Delta t/m_i \quad (4.8)$$

ここで、 $\mathbf{r}_i(t)$, $\mathbf{v}_i(t)$, Δt はそれぞれ時刻 t における粒子 i の位置、時刻 t における粒子 i の速度、シミュレーションの時間刻み幅である。

4.2.2 分子モデル

粒子に働く力 $\mathbf{f}_i(\mathbf{r})$ あるいは対応する保存力のポテンシャルは、使用する分子モデルによって異なる。この節では、MD シミュレーションの際に一般的に用いられるポテンシャルについて説明する。なお、下記に述べているポテンシャルは個々の粒子ペアが相互作用するときのポテンシャルを述べているものであり、すべての相互作用粒子ペアのポテンシャルを総和したものが系全体の総ポテンシャルエネルギーとなる。

分子内結合を表現するための方法として、結合長、結合角、ねじれ角を一定の範囲に制約する方法が用いられる。結合長を制約する方法のひとつとして、次の結合長ポテンシャル

$$U_b(r_{ij}) = \frac{1}{2}k_b(r_{ij} - r_0)^2 \quad (4.9)$$

は、結合している粒子同士をばねポテンシャルによって近似している。ここで r_{ij} , k_b , r_0 は結合している粒子 $i - j$ 間の距離、ばねポテンシャルのばね係数、ばねポテンシャルの自然長である。ただし、ばねポテンシャルによる結合長の制約を採用すると、シミュレーション実行時に結合間距離の振動の時間スケールが非常に小さくなることから、数値積分の時間刻み幅をかなり小さくする必要がある場合がある。そこで、観察したい現象が結合間距離の振動に依存しないと推察される場合は、ばねポテンシャルの代わりに結合長を直接固定して拘束する SHAKE 法 [42] や LINCS 法 [43] が用いられることがある。

結合角を制約する方法としては、次の結合角ポテンシャル

$$U_a(\theta_{ijk}) = \frac{1}{2}k_a(\theta_{ijk} - \theta_0)^2 \quad (4.10)$$

がある。ここで、 θ_{ijk} , k_a , θ_0 はそれぞれ、粒子 $i - j - k$ がなす角度、結合角ポテンシャルのばね係数、結合角ポテンシャルの自然結合角である。

ねじれ角を制約する方法としては、次のねじれ角ポテンシャル

$$U_t(\phi_{ijkl}) = \frac{1}{2}k_t(1 + \cos(m\phi_{ijkl} - \phi_0)) \quad (4.11)$$

がある。ここで、 ϕ_{ijkl} , k_t , m , ϕ_0 はそれぞれ、粒子 $i - j - k$ がなす平面と粒子 $j - k - l$ がなす平面がなす角度（ねじれ角）、ねじれ角ポテンシャルの強さを表す係数、ねじれ角ポテンシャルの周期を設定する整数、ねじれ角の自然角を表す。ねじれ角ポテンシャルの周期を設定する整数 m は、ねじれ角が安定となる角度（エネルギーが最小となるねじれ角）が 360 度回転に対して唯一でなく、周期的に複数ある場合に 2 以上の整数に設定される。また、分子モデルによっては、1 つのねじれ角に対して異なる m を持つ複数のねじれ角ポテンシャルの総和をポテンシャルとして設定する場合がある。

加えて、分子間相互作用や上記の結合相互作用によって直接相互作用しない分子内粒子同士の相互作用として、次の Lennard-Jones ポテンシャル [44]

$$U_{LJ}(r_{ij}) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (4.12)$$

が用いられることがある。ここで、 r_{ij} は粒子 $i - j$ 間の距離であり、 σ_{ij} , ϵ_{ij} は粒子ペアごとに決定される固有の係数である。ただし、粒子の種類の数に対して粒子ペアの種類数は 2 乗で増加するから、粒子ペア同士の係数をそれぞれを決定するのが実験コスト上難しい場合がある*1。このため、同種粒子の係数だけ決定して、異種粒子の係数は次の Lorentz-Berthelot 混合則 [45, 46]

$$\epsilon_{ij} = \sqrt{\epsilon_{ii}\epsilon_{jj}} \quad (4.13)$$

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2} \quad (4.14)$$

によって決定することがある。

4.2.3 自由境界条件と周期境界条件

シミュレーション系の中に有限個の粒子があり、それらの粒子は無限遠点まで自由に移動でき、かつそれらの粒子以外には相互作用する対象が存在しないとき、そのシミュレーションは自由境界条件下にあるという。自由境界条件を用いる MD シミュレーションでは、コンピュータの性能上（何もない空間上で）粒子数が多くとも数千万粒子しかないよ

*1 全ての組み合わせに対して決定されていたとしても、計算のために全ペアの情報を RAM 上に置いておくより計算効率が低い。

うな系の計算が限界である。しかしながら、実用的なシミュレーションにおいては、例えば1滴の水滴にもアボガドロ数個オーダーの水分子が含まれており、物質のバルク的な性質を調べることは到底不可能である。そこで、周期境界条件では、シミュレーションの対象となる粒子をあるセル内の領域に限って、セルの外側にはセルが周期的にコピーされていると考える (図 4.1)。コピーされたセル内の粒子との相互作用を考えることで、疑似的に可算無限個の粒子数を持つ系の計算を行う。周期境界条件では、粒子がセルの境界をまたぐ際に、セルの反対側から戻ってくることになる。ただし、周期境界条件の適用によって、シミュレーション結果がセルの大きさによって変化してしまう場合がある。周期境界条件は計算量の削減のために便宜的に導入したものであり、もともとの計算対象には周期境界という人工的に導入した条件は存在しないことに注意しなければならない。よって、観察したい物理現象が正しく表現されるように、セルの大きさを適切に選択する必要がある。

4.2.4 アンサンブル

MD シミュレーションにおいて、単に前述のポテンシャルエネルギーをもとにニュートンの運動方程式を解くと、総エネルギー (E)・体積 (V)・粒子数 (N) が一定の NVE (ミクロカノニカル) アンサンブルを設定してシミュレーションを実行していることになる。この場合、系の温度や圧力は、シミュレーションを通して間接的に決定される。しかしながら、MC シミュレーションで見たように、温度や圧力を設定して、これらが一定となるように制御したい場合がある。以下に温度や圧力の制御を実現する手法について述べる。

温度を制御する方法について最も簡単な方法として、目的の温度を達成するように分子の速度を一律にスケールする速度スケールリング法がある。速度スケールリング法では、現在の温度 T と目標温度 T_0 が与えられたとき、速度を

$$\mathbf{v}_i^{\text{new}} = \sqrt{\frac{T_0}{T}} \mathbf{v}_i^{\text{old}} \quad (4.15)$$

と設定する。ただし、 $\mathbf{v}_i^{\text{old}}$ は時間積分によって速度を更新した直後の値であり、 $\mathbf{v}_i^{\text{new}}$ は速度スケールリング法によって修正された速度の値である。また、系の温度は

$$T = \frac{1}{g} \sum_{i=1}^N m_i \|\mathbf{v}_i\|^2 \quad (4.16)$$

と計算される。ここで g は系の自由度である。よって、速度スケールリング法では、毎ステップごとに温度が強制的に T_0 へ更新される。

しかしながら、速度スケールリング法による温度制御は、平衡状態において統計力学が要請する NVT アンサンブルを再現することが保証されていないという問題がある。そこ

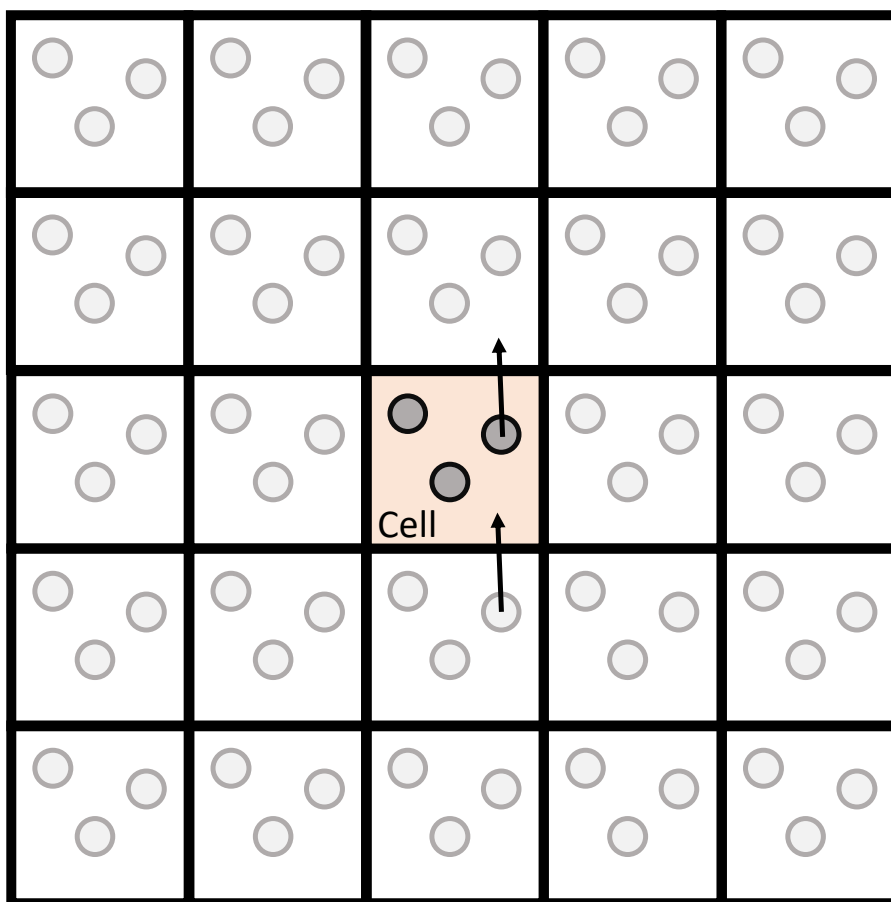


図 4.1 周期境界条件の模式図 (2 次元). 周期境界条件では, セルが周期的にコピーされており, コピーされたセル内の粒子からも相互作用を受ける. セルの境界をまたぐ粒子は, その反対側から出てくることになる.

で, NVT アンサンブルを再現する温度制御手法として Nosé-Hoover 熱浴 [47, 48] が用いられる. Nosé-Hoover 熱浴では, 温度制御のために系の状態として新たに熱浴の状態変数 ζ を導入し, 解くべき運動方程式を次の通り書き換える

$$\frac{\partial \mathbf{r}_i}{\partial t} = \mathbf{v}_i \quad (4.17)$$

$$\frac{\partial \mathbf{v}_i}{\partial t} = \frac{\mathbf{f}_i}{m_i} - \zeta \mathbf{v}_i \quad (4.18)$$

$$\frac{\partial \zeta}{\partial t} = \frac{1}{Q} \left(\sum_{i=1}^N m_i \|\mathbf{v}_i\|^2 - gT_0 \right). \quad (4.19)$$

ここで, Q は熱浴変数 ζ の質量に相当する量であり, 熱浴との熱の交換速度を決定する.

一方、圧力を制御するために、平衡状態において統計力学が要請する NPE アンサンブル再現するもっとも基本的な方法としては、Andersen の方法 [49] がある。Andersen の方法では、体積変化を表現するために、位置 \mathbf{r}_i をセルの体積 $V = L^3$ とセルサイズに依存しないスケール座標系 \mathbf{s}_i を用いて

$$\mathbf{r}_i = L\mathbf{s}_i \quad (4.20)$$

と表現し、セルサイズ L とスケール座標系 \mathbf{s}_i の双方についての運動方程式を考えることによってシミュレーションを実行する。このとき、運動方程式は次の通り書き換えられる

$$\frac{\partial^2 \mathbf{s}_i}{\partial t^2} = \frac{\mathbf{f}_i}{Lm_i} - \frac{2}{3V} \frac{\partial V}{\partial t} \frac{\partial \mathbf{s}_i}{\partial t} \quad (4.21)$$

$$\frac{\partial^2 V}{\partial t^2} = \frac{1}{M} \left(\frac{1}{3V} \left[\sum_{i=1}^N m_i L^2 \left\| \frac{\partial \mathbf{s}_i}{\partial t} \right\|^2 + \sum_{i=1}^N \mathbf{f}_i \cdot \mathbf{r}_i \right] - P_0 \right). \quad (4.22)$$

ここで、 M はターゲットとなる圧力 P_0 と現在の圧力の差に応じてセルサイズ V をフィードバック変形させる強さを決定するパラメータである。Andersen の方法は、体積を全方向一様に変形させるが、各軸別々に変形する必要がある場合の拡張として Parrinello-Rahman の方法 [50] が用いられることがある。

第5章

MD シミュレーションの高速な代理モデルの開発

本章では、MD シミュレーションにおいて部分系の確率的時間発展による計算高速化について調べた研究を報告する。

5.1 本章の目的と概要

序論で述べた通り、近年の MD シミュレーションの空間的・時間的スケールは非常に大規模化している。空間的に大きい系、すなわち膨大な粒子数を含む系については、計算領域を空間分割して同時に並列計算することによって高速に計算できる [12] ことから、GPU や専用計算デバイスなどの並列計算機による並列化や、複数の計算機にまたがった並列化がなされている。一方、時間方向における並列化については、計算が原理的に逐次的であることから並列計算は難しい。時間方向に並列化を行う手法としては時空間法や Parareal 法 [51] が知られているが、その応用例は小規模な問題にとどまる。また、複数の粒子をまとめてひとつの粒子とみなしてシミュレーションを行う租視化法を用いることによって、系に含まれる粒子数を減らし、かつ大きな時間刻み幅を用いることができることを利用して、空間的・時間的スケールが大きい現象のシミュレーションを達成することもできる。

一方、近年機械学習分野においては、機械学習に基づく代理モデルを構築することでシミュレーションの高速化に成功した例がある。流体シミュレーション分野では、時間刻み幅が大きくとることのできる代理モデルを機械学習によって構築することにより計算を高速化した例がある [52, 53]。また、分子シミュレーションにおいては、租視化モデルを機

機械学習によって獲得する研究 [54] や、第一原理 MD シミュレーションによる力場を深層学習による代理モデルによって高速化する研究 [55] が提案されている。

そこで、本研究では、訓練データとして用意した短時間の MD シミュレーションデータを用いて、系の一部の局所的な原子分子の時間発展予測モデルを獲得し、局所的な状態の長時間ダイナミクスを予測する手法を提案する。局所的な状態のみを時間発展させることによって、計算量を大幅に削減することが期待される。これにより得られる計算結果は、部分系の時間発展の様子だけであるが、求めたい統計量が部分系のみ依存して計算できるような場合には、十分であると言える。

この代理モデルは、系全体ではなく局所的な状態の時間発展を行うため、時間発展はもはや決定論的ではなく、代理モデルの構成には確率論的時間発展を考える必要がある。したがって、時間発展予測をおこなう機械学習モデルも時間発展の確率的な振る舞いをとらえる必要がある。

そこで、まず本章の初めに、時間発展の確率的な振る舞いを予測するための理論について述べ、その中で MD シミュレーションデータ（訓練集合）が持つべき性質について整理した。例えば、対象となるシミュレーションが非定常的であるときは、訓練集合として用意するシミュレーションデータからその先を予測できる正当性がないために、確率論的時間発展に関する理論的な保証をすることはできない。

理論的な整理の後に、確率的な振る舞いを予測するために提案する機械学習モデル MD-GAN について述べている。MD-GAN は、深層生成モデルの一つである GAN をベースとする代理モデルであり、GAN により確率的な振る舞いを獲得している。GAN についての基礎的な事項は機械学習・深層学習の章を参照されたい。MD-GAN の特徴的な点は、短いシミュレーションデータからなるべく予測精度の高いモデルを構築するための工夫が導入されている点である。特に、短いシミュレーションデータから長時間の予測をする際には、長時間予測の際に誤差が指数関数的に累積するエクスポージャー・バイアス (Exposure bias) [56] と呼ばれる現象が発生する可能性があり、エクスポージャー・バイアスを回避するいくつかの工夫を提案している。これらの工夫によるエクスポージャー・バイアスの低減効果について、簡易データを用いて調べている。

さらに、ポリエチレン溶融体シミュレーションを実験対象として、MD-GAN による代理モデルを学習し、その精度と高速化について検証している。局所的な状態のみで時間発展を行うことができるという MD-GAN の利点を生かすために、ポリエチレン分子の中心粒子だけを局所的な状態として選択して MD-GAN による予測を行う。精度については、ポリエチレン分子の中心粒子の平均二乗変位が、元の MD シミュレーションを基準として、MD-GAN による予測の一致度合いを見る。また、高速化については、同じ時間長さ

の平均二乗変位を計算する場合について、MD シミュレーションが必要とする計算時間と比べて MD-GAN による予測に必要な計算時間を比較する。MD-GAN による予測に必要な計算時間とは、訓練集合を用意するために必要な短時間 MD シミュレーションの計算時間、MD-GAN の学習に必要な計算時間および MD-GAN による予測に必要な計算時間の 3 つの合計であることに注意する必要がある。最後に、結果の考察と今後の展望について述べている。

5.2 確率的時間発展を行う代理モデルの理論的整理

MD シミュレーションデータを元にした、時間発展の確率的な振る舞いを予測するための理論と、その中で MD シミュレーションデータ（訓練集合）が持つべき性質について整理する。

5.2.1 局所系の確率的ダイナミクスとその長期予測可能性

MD シミュレーションを含む決定論的な動力学シミュレーションでは、一般に系の初期状態 \mathbf{x}_0 とその時間発展演算子 $F(\cdot)$ が与えられ、次の演算

$$\mathbf{x}_{i+1} = F(\mathbf{x}_i) \quad (5.1)$$

を K 回逐次的に繰り返し適用することで、系列長さ $K + 1$ （時間長さ K ）のシミュレーションデータを $\mathbf{X}_{0:K} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K)$ を得る。ここで、 \mathbf{x}_i は i 回目の逐次時間発展により得る系の状態であり、便宜的に時刻 i （の状態）と表記する。このような決定論的なダイナミクスでは、ある時刻においての状態が定まれば、以降の状態は厳密に確定し、確率的な振る舞いをするのではない。

ここで、決定論的な動力学シミュレーションにおける系の状態のうち一部分とってきて、これを部分系の状態と呼ぶことにする。ただし、部分系の状態とは、系のとりうる状態の集合を定義域とした関数 $h(\mathbf{x})$ によって次式

$$\mathbf{y}_i = h(\mathbf{x}_i) \quad (5.2)$$

のように系の状態から変換（抽出）される値のことであるとする。元のダイナミクスから関数 h により抽出されるダイナミクス、すなわち部分系の状態 \mathbf{y} の時間発展 $\mathbf{Y}_{0:K} = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_K)$ を考える。部分系を取り出す関数 h は、入力された元の状態のうち一部だけを切り取ったものであるから、一般に逆写像が存在せず、同じ \mathbf{y} に対応する状態 \mathbf{x} が複数存在することがあり得る。よって、部分系の状態の時間発展はもはや一意

に決定できない。そこで、代わりに部分系 \mathbf{y} の確率的なふるまいを考えることで、局所的なダイナミクスを予測することを考える。

いま、ある部分系の統計量を調べたいとして、そのためには系列長さ L のダイナミクスを知る必要があるとする。このとき、時刻 0 から時刻 $L-1$ までの部分系のダイナミクスの同時分布

$$p(\mathbf{Y}_{0:L-1}) = p(\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{L-1}) \quad (5.3)$$

に従うサンプルを得ることができれば、そのような統計量を計算することができる。そこで、本研究では、系列長さ L より十分に短い系列長さ K のダイナミクスから、系列長さ L のダイナミクスを得ることを考える。この際に重要となるのが、ダイナミクスの長期予測可能性である。長期予測とは、時刻 0 から時刻 $K-1$ までの系列長さ $K (< L)$ の部分系のダイナミクスの同時分布 $p(\mathbf{Y}_{0:K-1})$ が与えられたとき、時刻 K より先の部分系のダイナミクスの条件付同時分布を予測することであると定義する。一般には、このようなことはできない。なぜなら、時刻 $L-1$ までの条件付同時分布 $p(\mathbf{Y}_{0:L-1})$ について、これを展開すると

$$p(\mathbf{Y}_{0:L-1}) = p(\mathbf{Y}_{0:K-1}, \mathbf{Y}_{K:L-1}) \quad (5.4)$$

$$= p(\mathbf{Y}_{K:L-1} | \mathbf{Y}_{0:K-1}) p(\mathbf{Y}_{0:K-1}) \quad (5.5)$$

であるが、 $p(\mathbf{Y}_{0:K-1})$ は既知であっても $p(\mathbf{Y}_{K:L-1} | \mathbf{Y}_{0:K-1})$ が未知であるためである。このことから、長期予測が可能であるためには何かしらの条件を課して $p(\mathbf{Y}_{K:L-1} | \mathbf{Y}_{0:K-1})$ を推定する必要がある。本論文では、部分系のダイナミクスが斉時的 K 階マルコフ性を持つと仮定することで長期予測を行う。ただし、 K 階マルコフ性を持つダイナミクスとは、次の通り定義される性質を持つことである。

K 階マルコフ性 とは、任意の時刻 $L > K$ に対して

$$p(\mathbf{y}_L | \mathbf{Y}_{0:L-1}) = p(\mathbf{y}_L | \mathbf{Y}_{L-K:L-1}) \quad (5.6)$$

が成り立つことである。

すなわち、 K 階マルコフ性とは、状態の確率的時間発展が、直近の過去 K 個の状態の確率分布のみに依存しているということである。さらに、この条件付確率が時刻 L の変化に対して不変であるとき、これを斉時的 K 階マルコフ性を持つという。

斉時的 (K 階) マルコフ性を利用することで、長期予測が可能であることを示す。時刻

0 から時刻 $L - 1$ までの同時分布 $p(\mathbf{Y}_{0:L-1})$ について

$$p(\mathbf{Y}_{0:L-1}) = p(\mathbf{y}_{L-1} | \mathbf{Y}_{0:L-2}) p(\mathbf{Y}_{0:L-2}) \quad (5.7)$$

$$= p(\mathbf{y}_{L-1} | \mathbf{Y}_{0:L-2}) p(\mathbf{y}_{L-2} | \mathbf{Y}_{0:L-3}) p(\mathbf{Y}_{0:L-3}) \quad (5.8)$$

$$\vdots \quad (5.9)$$

$$= \left(\prod_{i=K}^{L-1} p(\mathbf{y}_i | \mathbf{Y}_{1:i-1}) \right) p(\mathbf{Y}_{0:K-1}) \quad (5.10)$$

$$= \left(\prod_{i=K}^{L-1} p(\mathbf{y}_i | \mathbf{Y}_{i-k:i-1}) \right) p(\mathbf{Y}_{0:K-1}) \quad (5.11)$$

が成り立つ。ここで時刻 $i \geq K$ の状態を導く条件付確率 $p(\mathbf{y}_i | \mathbf{Y}_{i-k-1:i-1})$ は時間に不変であるから、ただひとつの i についてこの条件付確率分布が既知であれば、任意の時刻 i について時刻 i の状態を導く条件付確率が既知となる。 $i = K$ の場合について考えると、確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ は、既知としていた確率分布 $p(\mathbf{Y}_{0:K})$ から求められる。したがって、局所的な状態が従うダイナミクスが斉時的 (K 階) マルコフ性を持つとき、時刻 $L > K$ までの条件付同時分布 $p(\mathbf{Y}_{0:L})$ は予測可能であるといえる。

ところで、局所的な状態が斉時的マルコフ性を持つかどうか、あるいは K の値はどの程度であるかについては、個々の (部分) 系によって異なるだろう。本論文では、斉時的 K 階マルコフ性をもつかについては、予備実験により実験的に確認する方法を取ることになっている。すなわち、対象とする部分系のダイナミクスについて、ある K について斉時的 K 階マルコフ性があると仮定して、長期予測を行い、長期予測結果に基づく統計量が本来の MD シミュレーションによる統計量と一致していれば、斉時的 K 階マルコフ性を満たしていると判断する。ただし、MD シミュレーションのダイナミクスの性質から斉時性および十分に大きな K について K 階マルコフ性が成り立っていることが妥当であると判断できる場合が多いだろうと考える。まず、斉時性の成立については、元の全体系の時間発展演算子 F はニュートンの運動方程式に従っており、運動方程式が途中で変化しない。また、部分系を取り出す関数 h についても、ここでは関数 h に対する時間変化を考えていない。よって、部分系の確率的時間発展も時間に対して不変であると考えられる。

また、 K 階マルコフ性についても、局所状態が定常状態 (局所的な平衡を含む) に達する系であれば、十分に長い時間スケールで近似的に成り立っていると考えることができるだろう。 K 階マルコフ性とは、直近の過去 K 個の状態の確率分布のみにみに依存しているということであった。系が定常状態に達するということは、いまある時刻に状態がある確率分布に従っていたとしても、十分に長い時間が経過するとその情報はほとんど失われ

て定常分布に従うということになる。これは、後ろの時刻からの視点から見れば、ある時刻の確率分布は、「十分に長い時間」より前の確率分布に（直接的に）依存していないということである。したがって、この「十分に長い時間」の系列長さを K と設定すれば、局所状態が定常状態に達する系では K 階マルコフ性が成り立つと考えられる。

さらに、 K の大きさについて議論する。本来の目的であるシミュレーションの高速化の観点から考えると、高速化の効率が変わる可能性があるという意味で、抽出する部分系の選び方による K の変化を理解することは非常に重要である。次節で詳しく述べる通り、 K が大きくなるほどそれに比例して必要な MD シミュレーションデータの時間ステップ数が長くなるため、MD シミュレーションは系全体をシミュレーションする高コストな計算であるから、 K が小さいことに越したことはない。 K の大きさは上述の通り、局所状態のある恣意的な分布から始まって定常分布に到達するまでの「十分に長い時間」に相当する。従って、これは対象となる系や抽出する部分系の選び方によって大きく変動すると考えられる。抽出する部分系の選び方によって K を減少させることができると考えられる根拠の 1 つとして、部分系が系全体と等しいという極端な例を考える。このとき、全体系が時間発展演算子（ここではニュートンの運動方程式）により確定的に時間発展することから明らかである。つまり、ある時刻の系全体の状態は、そのひとつ前の時刻の位置・速度だけから決定されていて、それ以前の位置・速度に直接依存していない。すなわち、このダイナミクスは状態を位置と速度とすれば $K = 1$ 階マルコフ性を持つ。状態を位置だけに限っても、2 つ前までの状態から速度が推定できることになるため、 $K = 2$ である。この事実は、部分系を大きく選べば選ぶほど K を小さくできるのではないかという仮説をもたらす。もちろん、部分系が系全体と等しいということは、機械学習によって MD シミュレーションの 1 ステップを予測することになってしまうことから、部分系が大きすぎれば代理モデルの計算量が増大してしまうため、逆に効率が悪くなってしまう。しかしながら、この仮説に基づけば、部分系を丁度良い大きさに設定することで、 K も小さく計算量も小さいような効率の良い代理モデルを構成できる可能性がある。この仮説については、本研究に続く共著の研究にて、一部の範囲で実験的に確かめられている。その論文の詳細は章末に参考として記載している。

5.3 確率的時間発展を行う代理モデル MD-GAN の構築

短時間 MD シミュレーションをもとに長期予測を行う、精度の高い機械学習モデルを提案する。

5.3.1 MD シミュレーションデータを用いた確率的ダイナミクスの推定とサンプリング

前節では、局所状態の時間発展が斉時的 K 階マルコフ性に従うとすれば、時刻 0 から時刻 $K - 1$ までの部分系のダイナミクスの同時分布さえわかれば、時刻 K 以降の長期予測が可能であることを理論的に示した。これを実際に適用して長期予測をおこなうためには、MD シミュレーションデータから時刻 0 から時刻 $K - 1$ までの部分系のダイナミクスの同時分布についての情報を抽出して、さらにそれを長期予測に利用する手法が必要である。これらの手法について、まず最低限の手法を構成する。

実用的な観点から言えば、長期予測のためにはダイナミクスの同時分布 $p(\mathbf{Y}_{0:L-1})$ の確率の値を出力できることはあまり重要ではなく、ダイナミクスの同時分布 $p(\mathbf{Y}_{0:L-1})$ から高速にサンプリングできることが重要である。これは、MC シミュレーションの節で説明したように、ボルツマン分布の解析的な形式が求まっても統計量の計算に有用ではなく、ボルツマン分布に従うサンプルを大量に集めることで統計量のサンプル平均を求めるのと同じである。よって、ダイナミクスの同時分布 $p(\mathbf{Y}_{0:L-1})$ をサンプリングできる手法を考えるべきである。

ダイナミクスの同時分布 $p(\mathbf{Y}_{0:L-1})$ をサンプリングするために、 K 階マルコフ性があることを踏まえ、時刻が小さい順から逐次的にサンプリングすることを考えると、次のサンプリングアルゴリズムを導く。

サンプリングアルゴリズム 1

時刻 $K (\leq L)$ までダイナミクスの $p(\mathbf{Y}_{0:K-1})$ のサンプリング値が与えられたとする。まず確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ から \mathbf{y}_K の値をサンプリングする。続けて、確率分布 $p(\mathbf{y}_{K+1} | \mathbf{Y}_{1:K})$ から \mathbf{y}_{K+1} をサンプリングする。これを繰り返して、最終的に時刻 $L - 1$ までのサンプリング値 $\mathbf{Y}_{0:L-1}$ を得る。

このサンプリングアルゴリズムを実行するには、時刻 K までの短時間のダイナミクス $p(\mathbf{Y}_{0:K-1})$ のサンプリングおよび、直近 K 個分の状態から確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ に従って次の状態をサンプリングする手法が必要である。

短時間のダイナミクス $p(\mathbf{Y}_{0:K})$ のサンプリングについては、時刻 K までの MD シミュレーションを行い、そのシミュレーションによって得られる全体系のダイナミクスから、部分系を抽出することで達成できる。これは部分系のサンプリング 1 回につき 1 回の MD シミュレーションを実行するということであるが、高速化の観点から 2 つ改善方法があ

る。1つ目は、MD シミュレーションを並列化することである。長時間の MD シミュレーションは逐次性があるために並列化できないが、短時間の MD シミュレーションの複数回実行は独立に実行できるために容易に並列化可能である。従って、系列長さ K のシミュレーションを並列化して実行することで所要時間を探索することができる。2つ目は、部分系の取り方によっては全体系から複数の部分系を抽出できることである。例えば、液体の水のバルク系では、各水分子は統計的に同じ振る舞いをしているから、部分系としてある特定の水分子1つを追いかけるような抽出は、統計的性質から考えると別の水分子ひとつを選んだのと何ら変わりはない。このような場合、1つの全体系ダイナミクスから複数の部分系ダイナミクスサンプルを計算できていることになる（例えば、水分子 1024 個の系であれば、1度の短時間シミュレーションで 1024 個の部分系のサンプリングが得られる）。従って、全体系自体が巨大で並列化が難しい場合でも、1度の全体系のシミュレーションによって複数の部分系のダイナミクスのサンプリングが可能な場合がある。本論文にて実験で用いているポリマー溶融体の系は、同種のポリマー分子が相互作用しながら拡散していく系であり、ポリマー 1 分子（のうちの特定の原子）を部分系としているため、1度のシミュレーションでポリマー分子の数だけサンプルを集めている。

一方、直近 K 個分の状態から次の状態をサンプリングする方法については、部分系の状態だけでは情報が不足しているため、MD シミュレーションから直接得ることはできない。よって、確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ をサンプリングするモデルを構成する必要がある。このモデルには、(1) 直近 K 個分の状態を入力にとって、その次の時刻の値を確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ に従って出力する機能、および (2) その確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ を、複数の時刻 $K-1$ までの部分系ダイナミクスのサンプルから学習する機能、の2つの機能が必要である。(1) については上記アルゴリズムの要求から自明な機能であるが、(2) については、MD シミュレーションから直接、部分系の時間発展法を得ることが難しいため、時刻 K までの MD シミュレーションから得られるの部分系ダイナミクスが確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ に従うことを利用して、複数の短時間 MD シミュレーションから得られるの部分系ダイナミクスデータから確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ を学習するというわけである。

本研究では、このようなことが実行できるモデルとして GAN を用いる。GAN は深層生成モデルの一種であり、単なる自己回帰モデルでは扱うことのできない確率的な振る舞いを予測することができる。GAN 以外の深層生成モデルでも同様のことを達成できると考えられるが、GAN は深層生成モデルの中でも最も多くの適用研究例があり、精度良く確率モデルを学習できることが知られているため、GAN を選択した。

機械学習の基礎の章で説明した通り、GAN は、与えられたデータの従う分布と（可能な限り）同じ分布に従ってサンプルを生成できる。従って、求める2つの機能を達成で

きる。サンプリングアルゴリズム 1 を実行する GAN のモデル構造について図 5.1(a) に示す。

以上より、長期予測を行う代理モデルを構成できた。まとめると、以下のようになる。

1. MD シミュレーションを実行して、時刻 K までの部分系のダイナミクスのサンプルを十分に集める。
2. 集めたサンプルを用いて、直近 K 個分の状態を入力にとって、その次の時刻の値を確率分布 $p(\mathbf{y}_K | \mathbf{Y}_{0:K-1})$ に従って出力する GAN を学習する。
3. 集めたサンプルをランダムに選んで初期値とし、学習した GAN を逐次的に $L - K$ 回適用し、部分系状態の系列長さ L の長時間予測を得る。
4. この長時間予測を何度も試行することで、長時間の部分系のダイナミクス $p(\mathbf{Y}_{0:L})$ からのサンプリングがいくらかでも実行できるので、得たサンプル集合から目的の統計量を算出する。

5.3.2 長期予測を行う代理モデルの問題点とその改善としての MD-GAN

前節で導入したサンプリングアルゴリズムには、実用上大きな問題がある。それは、長時間予測のために次の時刻の値を予測することを繰り返すことによって、誤差が予測回数に対して指数的に累積してしまうことである。この現象はエクスポージャー・バイアスと呼ばれ広く知られている。実際、上記のアルゴリズムを実行すると長時間予測のほとんどの部分は誤差の影響が大きく全く有用なデータを得られない。直近の情報だけから次の状態を逐次的に予測するものであり、長期的な誤差を保障する方法^{*1}もないので、原理的に誤差が累積してしまうのである。従って、目的の長時間予測スケールにおいて、誤差の累積がほとんど無視できる程度になるように誤差を低減させる必要がある。

本研究では、複数時刻の一括時間発展の採用および潜在変数と分布安定化機構の導入により、エクスポージャー・バイアスの問題を低減させるサンプリングアルゴリズムを構築した。潜在変数の導入と分布安定化機構について、順を追って詳細を説明する。

まず、複数時刻の一括時間発展については、逐次的な予測の回数を大幅に減らす目的として、先述の「サンプリングアルゴリズム 1」から次のように改善する。

^{*1} たとえば、正準力学系の運動方程式においては、シンプレティック数値積分によって長期的に誤差が拡大しないように計算できる。

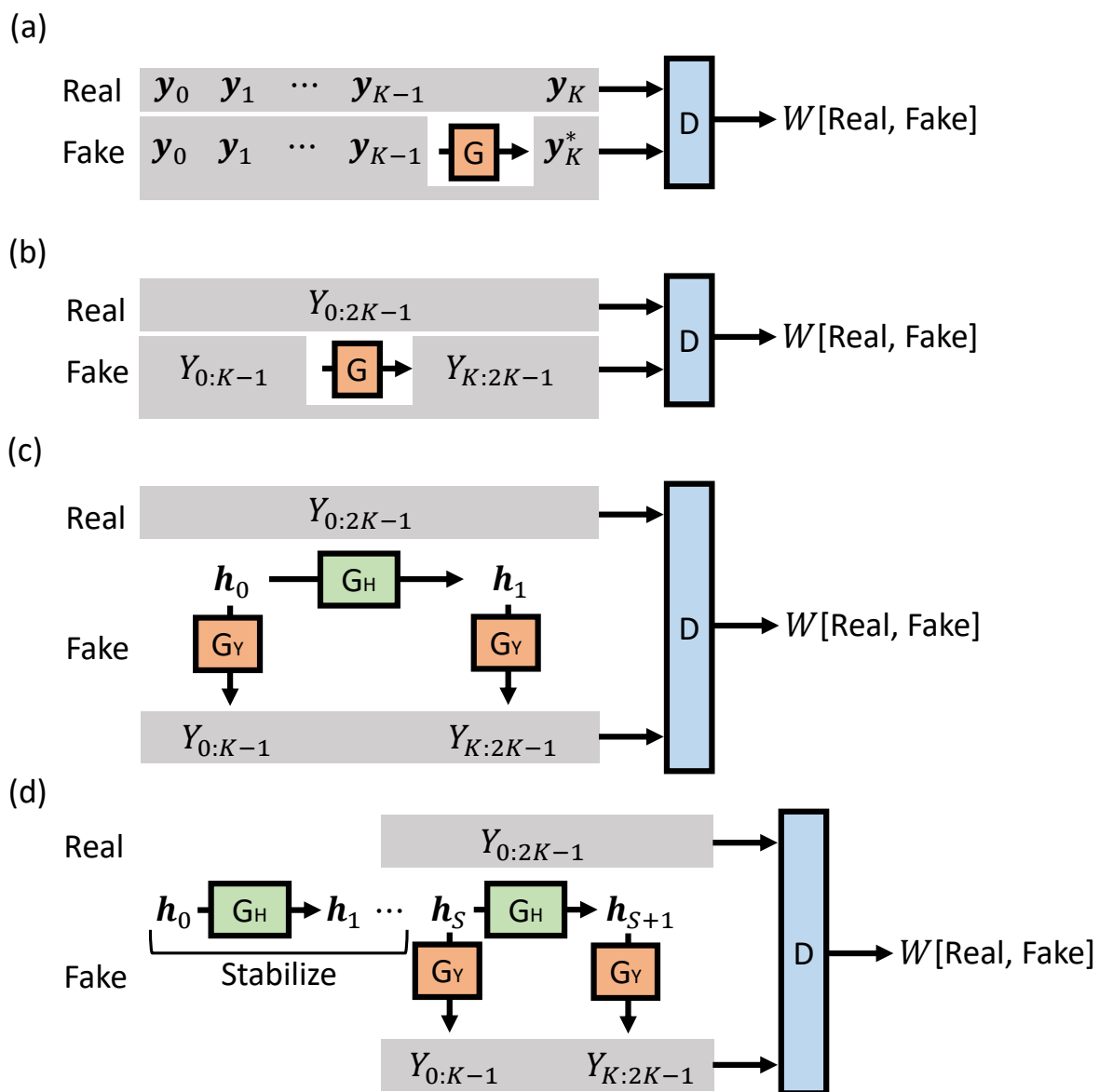


図 5.1 長期予測を行う GAN モデルの構造. (a) サンプルングアルゴリズム 1 で用いるモデル, (b) サンプルングアルゴリズム 2 で用いるモデル, (c) サンプルングアルゴリズム 3 で用いるモデル, (d) サンプルングアルゴリズム 4 で用いるモデル (MD-GAN).

サンプリングアルゴリズム 2

時刻 K までダイナミクスの $p(\mathbf{Y}_{0:K-1})$ のサンプリング値が与えられたとする。まず確率分布 $p(\mathbf{Y}_{K:2K-1}|\mathbf{Y}_{0:K-1})$ から $\mathbf{Y}_{K:2K-1}$ の値をサンプリングする。続けて、確率分布 $p(\mathbf{Y}_{2K:3K-1}|\mathbf{Y}_{K:2K-1})$ から $\mathbf{Y}_{2K:3K-1}$ をサンプリングする。これを繰り返して、系列長さ L までのサンプリング値 $\mathbf{Y}_{0:L-1}$ を得る。ただし L は K の倍数とする。

このアルゴリズムに基づき、予測モデルは図 5.1(b) のように改善される。このサンプリングアルゴリズムは、次の時刻の状態を 1 つずつ生成していくのではなく、 K 状態をひとまとまりとして次の K 状態を予測する。この変更によって、GAN は条件付確率分布 $p(\mathbf{Y}_{K:2K-1}|\mathbf{Y}_{0:K-1})$ に基づくサンプリングを学習することになるため、必要な短時間 MD シミュレーションの系列長さは $K + 1$ から $2K$ と約 2 倍に増加する。その代わりに、系列長さ L の長時間予測に必要な逐次的予測の回数は $L - K$ 回から L/K 回へと約 $1/K$ 倍に減少させることができる。これは、MD シミュレーションの計算時間が少し増えてしまう代わりに、予測回数に対して指数的に増加するエクスポージャー・バイアスの抑制が期待できる。

複数時刻の一括時間発展を導入したこのアルゴリズムによって、大幅に誤差を減少させることができるが、後述の簡易データによる実験結果が示す通り、サンプリングアルゴリズム 2 では不十分な結果となったため、さらなるエクスポージャー・バイアスの低減手法として、潜在変数と分布安定化機構を導入する。

潜在変数の導入については、サンプリングアルゴリズム 2 の部分系の時間発展に関して、部分系の状態 \mathbf{Y} とは別に、直接観測されない潜在状態 \mathbf{h} がある隠れマルコフモデルを考える。提案するモデルでは、時刻 i の潜在状態 \mathbf{h}_i は、時刻 iK からの複数時刻の部分系の状態 $\mathbf{Y}_{iK:iK+K-1}$ に対応しており、潜在状態から部分系の状態が生成されると考える。また、時間発展は潜在変数の下で行われ、時刻 i の潜在状態 \mathbf{h}_i から次の時刻 $i + 1$ の潜在状態 \mathbf{h}_{i+1} がサンプリングされる。部分系の状態 $\mathbf{Y}_{iK:iK+K-1}$ は直接時間発展に関わらず、時間発展する潜在状態 \mathbf{h}_i から変換されることによって間接的に時間発展が実行されると考える。このような考えのもと、サンプリングアルゴリズム 2 から次のように改善する。

サンプリングアルゴリズム 3

まず初めに、一様分布から隠れ変数の初期状態 \mathbf{h}_0 を $p(\mathbf{h}_0)$ からサンプリングする。次に、時刻 0 の潜在状態 \mathbf{h}_0 をもとにして、確率分布 $p(\mathbf{Y}_{0:K-1}|\mathbf{h}_0)$ から部分系の状態 $\mathbf{Y}_{0:K-1}$ をサンプリングする。また同時に、時刻 0 の潜在状態 \mathbf{h}_0 をもとにして、確率分布 $p(\mathbf{h}_1|\mathbf{h}_0)$ から時刻 1 の潜在状態 \mathbf{h}_1 をサンプリングする。続けて、時刻 1 の潜在状態 \mathbf{h}_1 をもとにして、確率分布 $p(\mathbf{Y}_{K:2K-1}|\mathbf{h}_1)$ から部分系の状態 $\mathbf{Y}_{K:2K-1}$ をサンプリングする。また同時に、時刻 1 の潜在状態 \mathbf{h}_1 をもとにして、確率分布 $p(\mathbf{h}_2|\mathbf{h}_1)$ から時刻 2 の潜在状態 \mathbf{h}_2 をサンプリングする。これを繰り返して、最終的に時刻 $L-1$ までのサンプリング値 $\mathbf{Y}_{0:L-1}$ を得る。ただし L は K の倍数とする。

サンプリングアルゴリズム 3 では、潜在変数を導入したため、確率分布 $p(\mathbf{Y}_{0:K-1})$ や $p(\mathbf{Y}_{K:2K-1}|\mathbf{Y}_{0:K-1})$ の代わりに、潜在変数に関する確率分布 $p(\mathbf{h}_0)$ や $p(\mathbf{Y}_{0:K-1}|\mathbf{h}_0)$ や $p(\mathbf{h}_1|\mathbf{h}_0)$ が導入されている。当然のことながら、部分系の状態 Y のサンプリング結果は、もとのサンプリング結果と一致する必要がある。GAN は識別器によって、与えられたサンプル集合が比較となるサンプル集合と分布をなるべく一致させるようにするものであるから、隠れ変数に関する分布を直接取り扱わず、隠れ変数を通してサンプリングされた部分系の状態を $p(\mathbf{Y}_{0:2K-1})$ に従うように学習することができる。もちろん、長期予測のためには斉時性をもつ必要があるから、隠れ変数の時間発展 $p(\mathbf{h}_1|\mathbf{h}_0)$ や、隠れ変数から部分系の状態のサンプリング $p(\mathbf{Y}_{0:K-1}|\mathbf{h}_0)$ は時不変であるとする。以上より、予測モデルは図 5.1(c) と改善される。

潜在変数の導入によって、さらなる累積の誤差の低減を達成することができる。このことを説明する。エクスポージャー・バイアスの大きな要因として、予測を繰り返すうちに、本来の疎な台からどんどん離れていってしまうということが考えられる。

簡単な例を使って説明しよう。状態空間を 2 次元とする（この例では、複数時刻の一括時間発展については考えず、1 時刻の状態を考えている）。初期状態は $\mathbf{p}_0 = (\cos \theta_0, \sin \theta_0)$ であり、かつこの状態の真の時間発展は

$$\mathbf{p}_{i+1} = \begin{pmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{pmatrix} \mathbf{p}_i \quad (5.12)$$

と定まっているとする。ただし、初期状態 θ_0 は $[0, 2\pi]$ から一様にランダムに選ばれとし、 $\Delta\theta$ は 1 より十分に小さな実数であるとする。いま、GAN は、真の時間発展の方法はわからない状態で、この初期状態とその時間発展した次の状態のペアだけがいくつかデータとして与えられて、この時間発展を予測しようとする*2。

*2 これは確率的な挙動を含まないが、一般に時間発展に確率的な挙動を含むものを考えている。

すると、データの量が十分にあれば、与えられた状態から小さい誤差で次の時刻の状態を予測できるが、完全に正しい時間発展を獲得できるわけではなく、微妙に異なったものになるだろう。今、GAN は多少の誤りを含んで、時間発展を

$$\mathbf{p}'_{i+1} = 1.001 \begin{pmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{pmatrix} \mathbf{p}_i \quad (5.13)$$

と学習してしまったとしよう*³。このとき、ある入力状態 $\mathbf{p}_0 = (\cos \theta_0, \sin \theta_0)$ から次の時刻の状態を予測した時の真の状態との差は

$$\mathbf{p}'_1 - \mathbf{p}_1 = 0.001\mathbf{p}_0 \quad (5.14)$$

しかなく、この大きさは 0.001 であり、非常に小さい量である。GAN がこのような小さな予測誤差を持ってしまっても不思議ではない。

しかしながら、これをもとに長期予測を行うと事態は変わってくる。いま、GAN により得られた時間発展を $L (\in \mathbb{N})$ 回適用したとする。すると、真の状態 \mathbf{p}_L および予測された状態 \mathbf{p}'_L の差は

$$\mathbf{p}'_L - \mathbf{p}_L = (1.001^L - 1)\mathbf{p}_0 \quad (5.15)$$

となる。明らかに、この誤差の大きさは時間発展の回数 L に対して指数的に増加する。この様子を図に示したのが図 5.2 である。この図からわかる通り、とりうる状態の分布は状態空間全体に対して疎な空間となっており、微小な誤差が累積することによって、分布の外側の本来とりえない状態へと遷移してしまっている。今回は線形的な時間発展のみを考えているが、本来は時間発展はニューラルネットワークに基づく GAN によって実装されており、学習の際に見たことのない状態が入ってきたときの挙動については未知である。このようなことが起きてしまうのは、状態の確率分布の台が疎な空間になっているためであり、状態がより密な状態であればこのような事態が発生するのを避けられる可能性がある。

潜在変数を使って、時間発展の際の状態空間をより密にすることで、このような問題を回避することができる可能性がある。いま、状態 $\mathbf{p} = (\cos \theta, \sin \theta)$ が潜在空間 h 上の状態 $h = \theta$ と 1 対 1 対応している、すなわち状態の偏角が潜在状態となっており、潜在変数の値域が $[-2\pi, 2\pi]$ の範囲に制限されているとする。このとき、潜在状態が取り得る状態は潜在空間を埋め尽くしている。すると、ニューラルネットワークが学習時に見たこと

*³ もちろん、これは著者が恣意的に選んだものであって、実際には非線形的でノイジーな時間発展が獲得されるだろう。

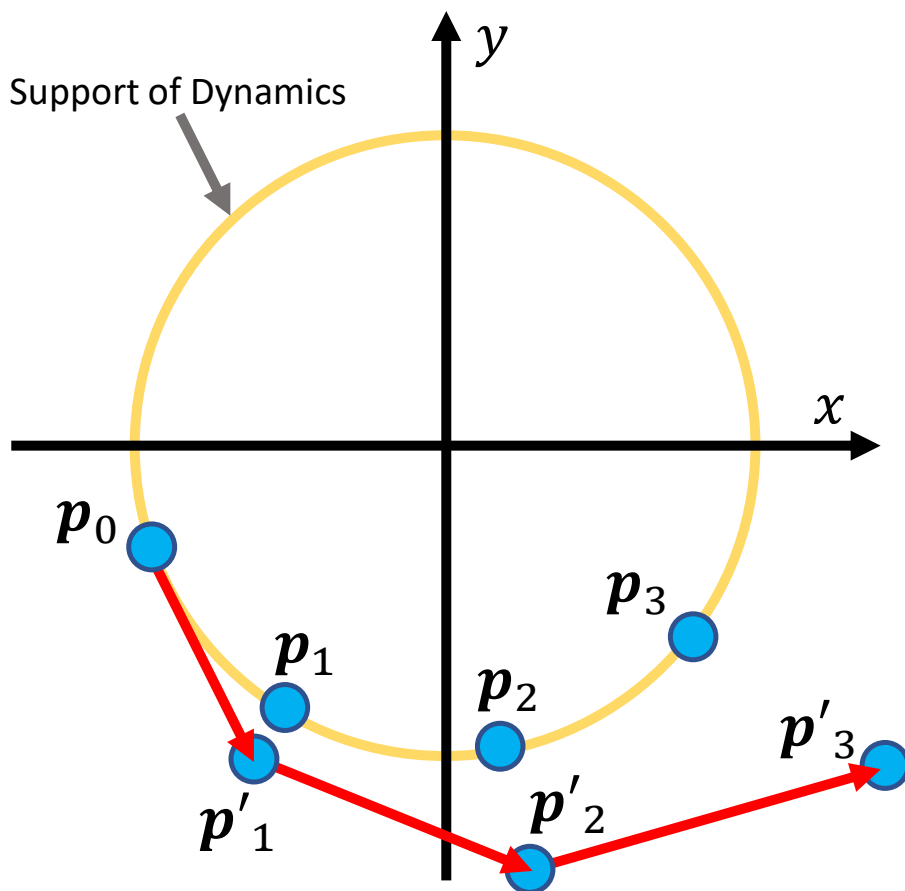


図 5.2 予測誤差の累積の模式図．初期値 p_0 から時間発展をする際に，本来のダイナミクスの分布の台（黄）から外れ，時間発展を繰り返すとともに大きな予測誤差を生み出している．

がない入力をなくすることができる．潜在変数を導入するのは，このような効果を期待するためである．

もともと考えていた問題に立ち返れば，潜在変数の次元を状態の複数時刻一括の状態の次元^{*4}と比べてなるべく小さくとれば，状態の確率分布の台が疎な空間になっていても，上記の例のように，次元数が少ないために潜在状態における確率分布の台はより密にできる可能性がある．具体的に GAN のよって獲得される潜在変数の分布は，学習結果に依存することになるが，時間発展が行われる次元数を下げることによって，未知入力存在を極力減らすことが重要である．

^{*4} 状態の次元を D としたとき， K 時刻が一括されているので， KD 次元になる．

潜在変数の導入によるエクスポージャー・バイアスの低減をより効果的にするために、さらに潜在変数の分布安定化機構を導入する。分布安定化機構とは、潜在状態のうち初めのいくつかの状態は分布が安定していないと考え、部分系状態の生成には使用しないようにすることである。このような考えのもと、サンプリングアルゴリズム 3 から次のように改善する。

サンプリングアルゴリズム 4(MD-GAN)

まず初めに、一様分布から隠れ変数の初期状態 \mathbf{h}_0 を $p(\mathbf{h}_0)$ からサンプリングする。時刻 0 の潜在状態 \mathbf{h}_0 をもとにして、確率分布 $p(\mathbf{h}_1|\mathbf{h}_0)$ から時刻 1 の潜在状態 \mathbf{h}_1 をサンプリングする。これを繰り返して、時刻 R の潜在状態 \mathbf{h}_R を得る。この時刻 R が、初めの部分系の状態 $\mathbf{Y}_{0:K-1}$ と対応すると考え、時刻 R の潜在状態 \mathbf{h}_R をもとにして、確率分布 $p(\mathbf{Y}_{0:K-1}|\mathbf{h}_R)$ から部分系の状態 $\mathbf{Y}_{0:K-1}$ をサンプリングする。また同時に、時刻 R の潜在状態 \mathbf{h}_R をもとにして、確率分布 $p(\mathbf{h}_{R+1}|\mathbf{h}_R)$ から時刻 $R+1$ の潜在状態 \mathbf{h}_{R+1} をサンプリングする。続けて、時刻 1 の潜在状態 \mathbf{h}_{R+1} をもとにして、確率分布 $p(\mathbf{Y}_{K:2K-1}|\mathbf{h}_{R+1})$ から部分系の状態 $\mathbf{Y}_{K:2K-1}$ をサンプリングする。また同時に、時刻 $R+1$ の潜在状態 \mathbf{h}_{R+1} をもとにして、確率分布 $p(\mathbf{h}_{R+2}|\mathbf{h}_{R+1})$ から時刻 $R+2$ の潜在状態 \mathbf{h}_{R+2} をサンプリングする。これを繰り返して、最終的に時刻 $L-1$ までのサンプリング値 $\mathbf{Y}_{0:L-1}$ を得る。ただし $L-1$ は K の倍数とする。

分布安定化機構は、潜在空間のトポロジーが一様分布のトポロジーと一致しないときに有効であると考えられる。式 5.12 の回転運動の例を用いて説明する。入力を式 5.12 の回転運動とし、潜在空間が 2 次元空間であるとする。「サンプリングアルゴリズム 3」では、円状に分布した回転運動する状態 \mathbf{p} が潜在変数 \mathbf{h} と関連づけられており、時刻 0 の潜在状態は一様分布からサンプリングすることになっている。従って、円運動の状態は一様分布に埋め込まれることになる。すると、状態の分布は偏角によって完全に分離されなければならないが、潜在状態は一様分布でなければならないので、どのような潜在空間の埋め込み方を選んでも、図 5.3 のように、偏角の大きく異なる状態が隣り合う点が存在することになる。すると、潜在空間上で時間発展した際に、この点にある状態は自移管発展の些細な予測誤差によって偏角が大きく変わってしまうことになり、誤差の著しい増大をまねく。

そこで、分布安定化機構を導入する。分布安定化機構は、初めのうちいくつかの時刻における潜在状態は分布が不安定であると考え、最初の S 個は実空間 \mathbf{y} への変換に用いないことにする。これは、潜在変数の時間発展モデルによって、時刻 S までに潜在空間上の

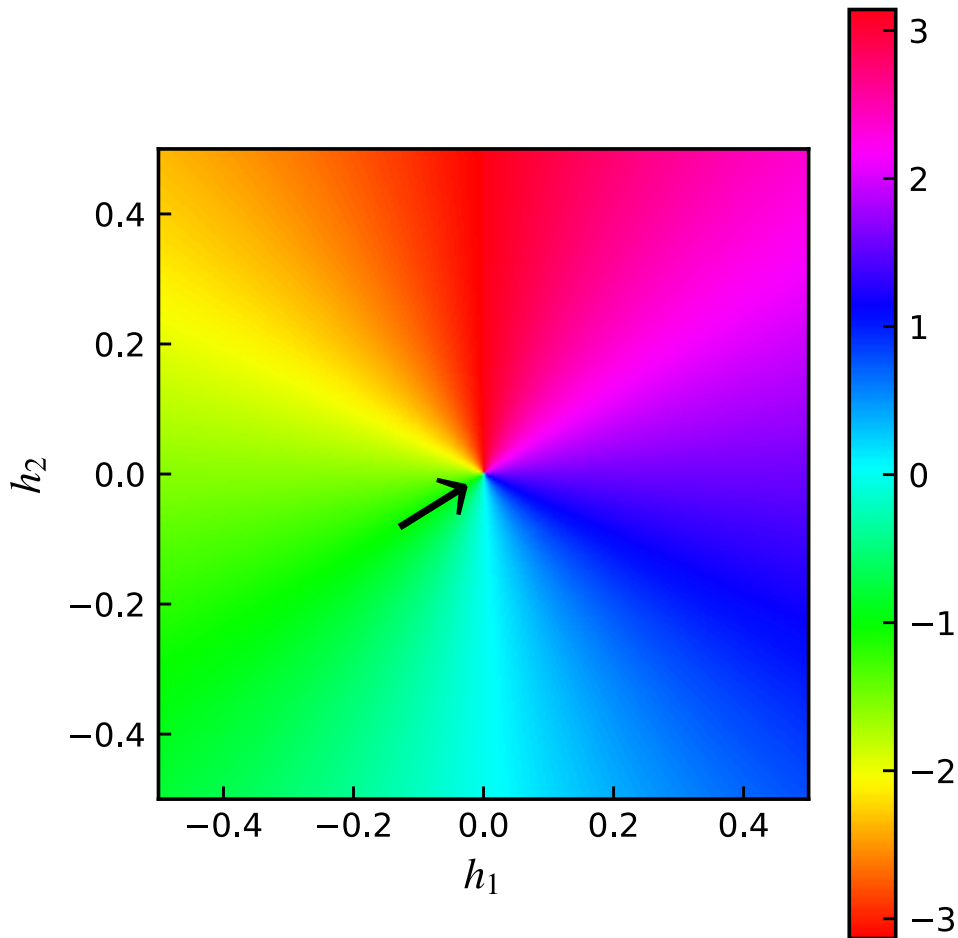


図 5.3 ダイナミクスの偏角を一様分布に埋め込んだ様子．偏角（色）の状態を 2次元の潜在空間 (h_1, h_2) に対応させようとした結果，図中央矢印に示す箇所において偏角が大きく異なる状態が隣接することになってしまった．

適切な分布が学習されることを期待するものである．最初の S 個を捨てることになるため，初めの部分系の状態 $\mathbf{Y}_{0:K-1}$ は一様分布と対応している必要が無く，一様分布から R 回の時間発展によって得られた分布と対応していればよい．むしろ，GAN が予測誤差を減らす過程で，自動的に適切な潜在分布が獲得されることを期待する．

予備実験の結果，分布安定化機構の安定化時刻 S は，実験的には 7 から 10 程度の範囲でランダムに選ぶ方が良いことが分かっている．また，分布安定化機構と合わせて，潜在状態の時間発展モデルの出力層にバッチ・ノーマライゼーションを適用し，ミニバッチ内で潜在状態ベクトルの各要素の平均・分散をそれぞれ 0 と 1^2 に設定することで，さらなる累積誤差の抑制を達成できていることが予備実験により確認されている．従って，これを含めて最終的に考案したモデルを MD-GAN とする．MD-GAN 内に使用されているモ

デルの具体的な構造は、図 5.1(d) となる。以上より、長期予測のエクスポージャー・バイアスをできるだけ低減する代理モデルを構成できた。まとめると、MD-GAN は以下のようになる。

1. MD シミュレーションを実行して、時刻 $2K$ までの部分系のダイナミクスのサンプルを十分に集める。
2. 図 5.1(d) に示す分布安定化機構付きの隠れマルコフモデルに対して、時刻 $K - 1$ までの状態を入力にとって、その次の K 個分の時刻の状態を確率分布 $p(\mathbf{y}_{K:2K-1}|\mathbf{Y}_{0:K-1})$ に従って出力する GAN を学習する。ただし、潜在空間の時刻 0 の初期状態は一様乱数から生成し、時刻 S を $\mathbf{Y}_{0:K-1}$ 、時刻 $S + 1$ を $\mathbf{Y}_{K:2K-1}$ と対応させる。また、時刻 S は 7 から 10 の一様ランダムな整数値を毎回ふり直す。
3. 潜在空間の時刻 0 の初期状態を一様乱数から生成し、逐次的に $(R + L/K)$ 回時間発展をさせ潜在状態を得る。その後、最初の S 個の潜在状態を捨て、後ろの L/K 回の潜在状態から部分系の状態を生成し、部分系状態の系列長さ L の長時間予測 $\mathbf{Y}_{0:L-1}$ を得る。
4. この長時間予測を何度も試行することで、長時間の部分系のダイナミクス $p(\mathbf{Y}_{0:L-1})$ からのサンプリングがいくらかでも実行できるので、得たサンプル集合から目的の統計量を算出する。

5.4 MD-GAN によるポリエチレン溶融体中心座標の平均二乗変位の高速計算

MD-GAN を用いて、実用的な系の 1 つであるポリエチレン溶融体中心座標を対象とした長期予測を行い、その効率性を評価する。

5.4.1 簡易データによる長期予測モデルの予測精度の比較

以上の議論より、エクスポージャー・バイアスを極力低減する長期予測モデル MD-GAN が構築された。本節では、実際にポリエチレン溶融体の分子中心の平均二乗変位を高速計算するために、MD-GAN を使用してその性能を評価するが、その前の予備実験として、簡易データを用いて 5.3 節に記載した 4 つのサンプリングアルゴリズムの予測精度を検証する。

検証用に用いる簡易データは、次の確率変数

$$y_k = \sin\left(\phi_0 + \frac{k}{2}\right) + 0.1\epsilon_k \quad (5.16)$$

に従うサンプリング集合であるとする。ここで ϕ_0 は 0 から 2π の値をとる一様分布を確率分布に持つ確率変数であり、 $\epsilon = (\epsilon_1, \epsilon_2, \dots)$ は無相関標準正規分布であるとする。すなわち、この簡易データはランダムな位相から始まって、かつ 1 ステップごとに位相が 0.5 だけ進む振幅 1 の正弦波に、標準偏差 0.1 の正規分布に従う独立なノイズが各ステップに加算されているような波形となっている。図 5.4(a) はこの簡易データのサンプル例を示している。

この簡易データをもとに、サンプリングアルゴリズム 1-4 を用いた学習と予測を行った*5。この実験では、部分系を直接与えていると考えることから、与えられたデータに対しての時間間引きや部分系の選択を考慮する必要がないため、ステップスキップ R は 1 とし、部分系としては簡易データ $\mathbf{Y} = (y_1, y_2, \dots)$ をそのまま用いる。また、マルコフ性の回数 K については、 $K = 8$ とした。今回用いているノイズ付き正弦波は、ノイズがない場合には直近 2 つの状態から初期位相 ϕ_0 が推定できるため、 $K = 2$ と設定できるが、ノイズが付加された場合の初期位相の推定は、厳密には無限の長さが必要である。ただし、 K の大きさが大きいほど初期位相の統計的な推定誤差を小さくできることから、適当なサイズで打ち切ってよいと考えて $K = 8$ を選んでいる。

サンプリングアルゴリズム 1-4 の学習についての詳細な設定は次の通りである。各アルゴリズムで用いているモデル構造は図 5.1 のとおりである。サンプリングアルゴリズム 1 および 2 で用いるモデルは、実空間の時間発展を行うモデル G と、それらを識別するモデル D からなっている。サンプリングアルゴリズム 3 および 4 で用いるモデルは、潜在変数上の時間発展を行なうモデル G_H 、潜在変数から実空間の状態に変換するモデル G_R 、生成された状態を識別するモデル D からなる。潜在空間の次元は 2 とした。学習のための損失関数は第 3 章の GAN の説明に示されているものを用いており、Gradient-Penalty の強さを定めるパラメータは $\lambda = 10.0$ と設定している。また、損失関数の最小化は勾配法的一种である Adam アルゴリズム [57] を使用した。

各アルゴリズムの学習完了後、それぞれのアルゴリズムについて系列長さ 64 の独立な予測を 512 回行った。図 5.4 に各アルゴリズムの予測結果から得られたサンプルを示している。図 5.4 から、次のようなことが観察される。まず、サンプリングアルゴリズム 1 および 2 については、時間発展に伴い振幅が減少あるいは増加していることが分かる。こ

*5 サンプリングアルゴリズム 4 は MD-GAN である。

これは時間発展に伴う誤差が累積するエクスポージャー・バイアスが現れているということであり、本実験で用いている非常に簡易的な時系列データでさえ、高々数回の予測の繰り返しによる誤差が無視できないほど大きくなっていることが分かる。次に、サンプリングアルゴリズム 3 においては、1 および 2 と比べると振幅については正しいデータに比較的近いようである。しかしながら、初めの時刻にランダムであった位相が、後ろの方の時刻では異なるサンプル間において位相が揃ってしまう同期現象が起きていることが観察できる。一方、サンプルアルゴリズム 4 は、これらのアルゴリズムの中で最もよくデータに一致しており、アルゴリズム 3 のような位相同期現象が観察されていない。この違いが発生した要因は、アルゴリズム 3 および 4 の間の唯一の違いである分布安定化機構であると考えられる。

この違いを詳しく考察するために、系列長さ 64 のデータのうち一番はじめの部分を生じた元となる隠れ変数を h_S 、一番うしろの部分を生じた元となる隠れ変数を h_E とし、それぞれサンプルアルゴリズム 3 および 4 の 512 個分の隠れ変数 h_S, h_E を散布図として図 5.5 に表示した。サンプルアルゴリズム 3 の初期隠れ変数 h_S は、図 5.5(a) のとおり一様分布となっているが、時間発展した最終的な隠れ変数 h_E は、図 5.5(b) のとおり一様分布からかけ離れた分布形状となっている。一方、サンプリングアルゴリズム 4 については、隠れ変数は図 5.5(c) の h_S および図 5.5(d) の h_E ともに円形上の分布を成しており、ほとんど同じ分布となっていることが分かる。サンプリングアルゴリズム 3 においては正弦波という周期的なデータに対して潜在変数が一様分布とトポロジーの一致しない構造を持っているため、位相の同期という本来データにない構造が現れたものであると考えられる。一方、サンプリングアルゴリズム 4 では分布安定化機構により潜在空間上の適切な分布が学習されたことを示している。

以上の実験より、サンプルアルゴリズム 4、すなわち MD-GAN は、適切なモデル構造によるエクスポージャー・バイアスの大幅な抑制と、分布安定化機構による適切な潜在変数構造の獲得を通して安定な時間発展を達成する、優れたモデルであることが示された。

5.4.2 ポリエチレン溶融体の運動について

次に、ポリエチレン溶融体の運動についての理論的背景について説明する。次節において、ポリエチレン分子の中心座標を部分系として MD-GAN により長期予測を行うことで、ポリエチレン分子の中心座標の平均二乗変位を計算する。ここで計算される平均二乗変位はポリエチレンの粘弾性と強い関係があり、高分子において重要な物性である粘弾性に関する統計量を高速に計算できることには大きな意義がある。

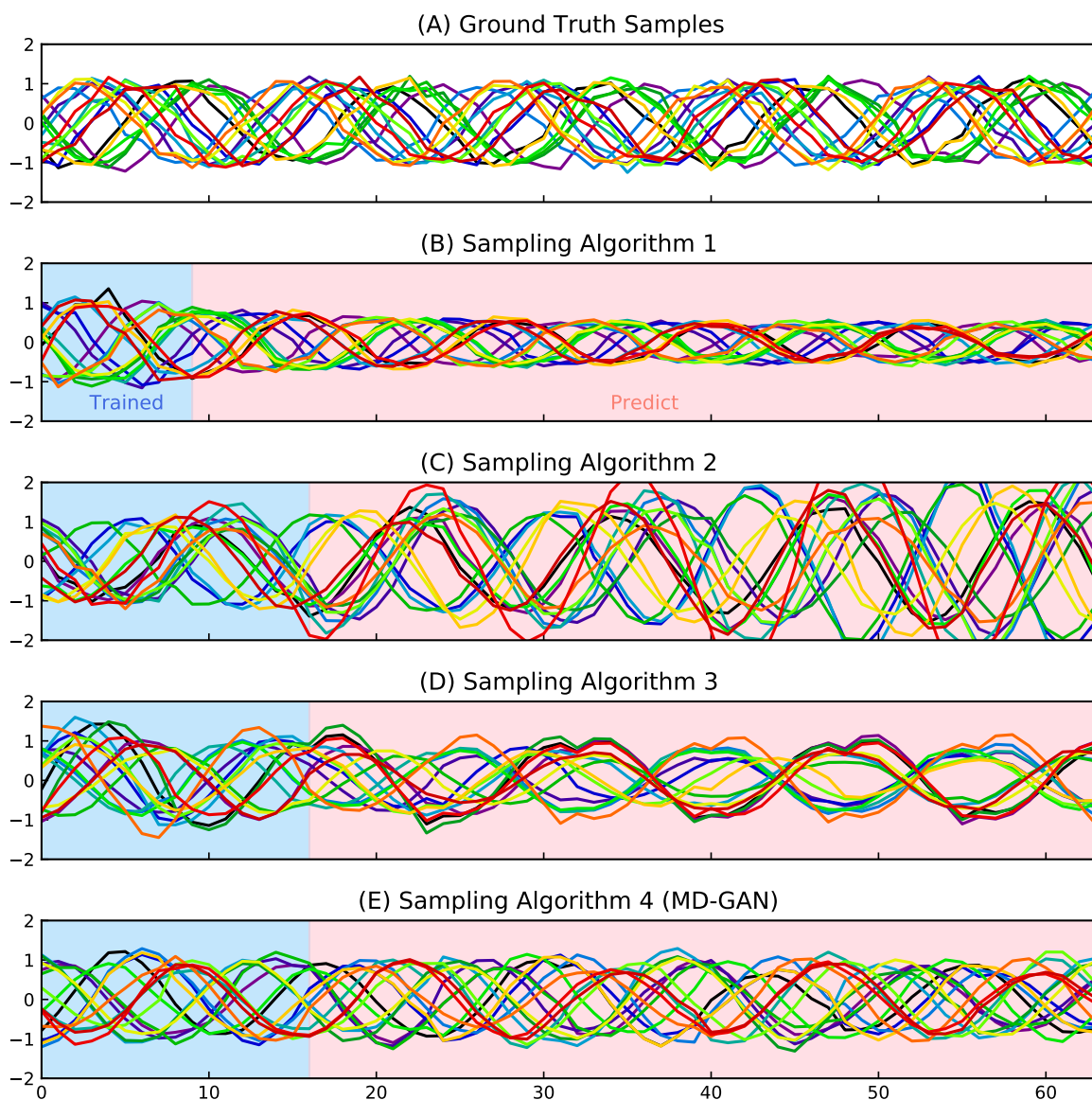


図 5.4 予備実験における真のデータと各サンプリングアルゴリズムによる予測結果の比較. (A) 真の分布から得られた長期ダイナミクスサンプル, (B) サンプリングアルゴリズム 1 によって得られた長期予測ダイナミクスサンプル, (C) サンプリングアルゴリズム 2 によって得られた長期予測ダイナミクスサンプル, (D) サンプリングアルゴリズム 3 によって得られた長期予測ダイナミクスサンプル, (E) サンプリングアルゴリズム 4 (MD-GAN) によって得られた長期予測ダイナミクスサンプル.

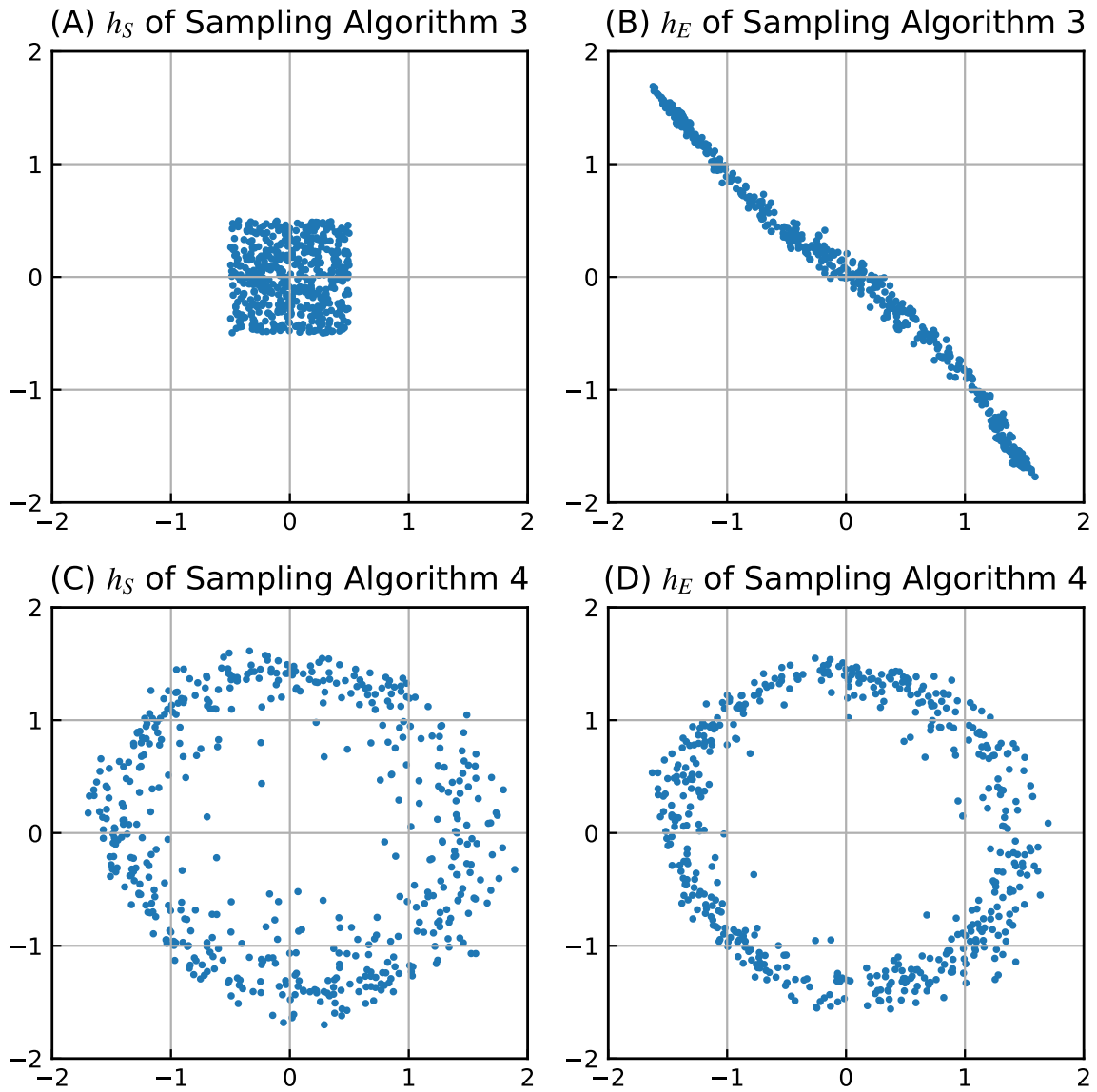


図 5.5 サンプルングアルゴリズム 3 および 4 における潜在変数 (初期状態・終状態) の分布の比較. (A) サンプルングアルゴリズム 3 の初期状態の分布, (B) サンプルングアルゴリズム 3 の終状態の分布, (C) サンプルングアルゴリズム 4 (MD-GAN) の (最初の実状態と対応する) 初期状態の分布, (D) サンプルングアルゴリズム 4 (MD-GAN) の終状態の分布.

高分子は、ゆっくり引っ張ることで元に戻らずにそのまま伸びたままになる低速領域での粘性の性質と、素早く力を掛けて変形させようとする元に戻る力が働く高速領域での弾性の性質の両方を併せ持つことが多く、これを粘弾性体という。粘弾性体の粘性・弾性は、ひずみを与えようとするときの変形の速さによって切り替わり、速いほうからガラス状態・ゴム状態・流動的状态と大きく変化する [58].

このような高分子の粘弾性は、分子スケールでは高分子鎖の絡み合いが関係している。分子量が小さい高分子では、図 5.6(a) に示すようなばね・ビーズに基づくラウスモデルによって説明できる。分子量が多くなってくると、分子が分子鎖を含む管上に固定され、図 5.6(b) に示すようなばね・ビーズが管に沿った方向にしか運動できないと考えるレプテーションモデルによる説明が有効である。この管内にのみ移動する性質は、分子同士が絡み合うことによって、周りの分子鎖により閉じ込められることにより発生すると考えられる。このレプテーションモデルにより、高分子の粘弾性を説明できる。

ポリエチレン溶融体の粘弾性は、ポリエチレン溶融体の分子中心の平均二乗変位によって部分的に評価することができる。ここで分子中心 \mathbf{r}_c とは、分子鎖の中央にある粒子の座標のことである（分子の重心とは異なる）。ポリエチレン溶融体の分子中心の平均二乗変位

$$g_1(t) = \langle \{\mathbf{r}_c(t) - \mathbf{r}_c(0)\}^2 \rangle \quad (5.17)$$

は、おおよそ

$$g_1(t) = \begin{cases} t^1 & (t < \tau_0) \\ t^{1/2} & (\tau_0 < t < \tau_e) \\ t^{1/4} & (\tau_e < t < \tau_N) \\ t^{1/2} & (\tau_N < t < \tau_R) \\ t^1 & (\tau_R < t < \tau_0) \end{cases} \quad (5.18)$$

と導かれる。ただし、 $\tau_0, \tau_e, \tau_N, \tau_R$ は、ラウスモデルとレプテーションモデルによって導かれる各スケール則を支配する時定数である。 τ_N 程度までの時間スケールではラウスモデルが支配的になっており、 τ_N から τ_R の時間スケールではレプテーションモデルによる絡み合いの効果が支配的となっていると考えられている。 τ_R 以降の時間スケールでは、分子は管から抜け出すことにより絡み合いの効果が薄れ、単なるブラウン運動をするようになる。ブラウン運動の拡散速度が $g_1(t) \sim t^1$ （通常拡散）なのに対し、ラウスモデルや絡み合いの効果により拡散速度が低下し、 $g_1(t) \sim t^{\alpha(<1)}$ （異常拡散）となっている。

このことから、ポリエチレン溶融体の分子中心の平均二乗変位はポリエチレン溶融体の粘弾性の粘性・弾性切り替わりの時間スケールと大きく関わっていることが分かる。従っ

(a)



(b)

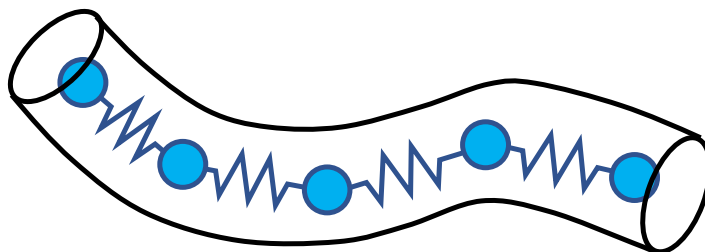


図 5.6 高分子の理論的モデルの概略図. (a) 隣り合う粒子がばねによって接続されるラウスモデルの概略図. (b) 高分子が管内の移動のみに制限されるレピュテーションモデルの概略図.

て、MD-GAN による長期予測では、平均二乗変位を予測する上で、これら絡み合いの効果による異常拡散およびそこからブラウン運動の通常拡散へ遷移していく過程を正しく予測できるかどうか重要である。

5.4.3 ポリエチレン溶融体の MD シミュレーション

ポリエチレン溶融体の MD シミュレーションは、表 5.1 に示す計算条件で行った。シミュレーションのためのソフトウェアとして GROMACS[59] を使用した。系には 300 個のポリエチレン分子が含まれており、ポリエチレン分子は、各炭素原子につながる水素原子を省略して炭素原子とまとめた TraPPE-UA 粗視化モデル [60] によって表現する。ポ

表 5.1 ポリエチレン溶融体シミュレーション条件

ソフトウェア	GROMACS
アンサンブル	NVT(Nose-Hoover)
数値積分	Leap-Frog
租視化モデル	TraPPE-UA
温度	500 K
密度	0.650 g/cm ³
時間刻み幅	2 fs

リエチレン分子はすべて租視化粒子が直鎖状にちょうど 100 粒子繋がっており、1 mol あたり 1405 g となっている。分子内の結合長は LINCS 法によって拘束されており、結合角、ねじれ角の相互作用に 4 章はポテンシャルモデルが使用されている。また、分子間の相互作用はカットオフ距離 1.2 nm の近距離 Lennard-Jones ポテンシャルが使用されている。さらにシミュレーション系には全軸に周期境界条件が適用されている。また、得る MD シミュレーションデータは平衡状態である必要があるから、適当な初期状態から NPT アンサンブルシミュレーションを十分長さ行い平衡状態を得た。本シミュレーションは NVT シミュレーションを用いた。

5.4.4 MD-GAN によるポリエチレン溶融体の中心座標の長期予測

MD-GAN を用いて、ポリエチレン溶融体の中心座標の長期予測を行う。まず第一段階として、前節の通りに実行する MD シミュレーションから時刻 2K までの部分系のダイナミクスのサンプルを十分に集める。ここで、MD シミュレーションの時間刻み幅は、ポリエチレン分子の動きより十分に遅く、MD シミュレーションの 1 ステップではほとんど分子が動かない。このような状態では、ほとんど同じようなデータが重複して MD-GAN に入力されることになり、またマルコフ性の階数 K が巨大になることから構築すべき MD-GAN の入出力次元も無意味に大きくなってしまう。このようなことを防ぐために、MD シミュレーションを時間方向に間引くことにする。本研究で使用するポリエチレン溶融体シミュレーションの時間刻み幅は 2 fs であるから、例えば 2 つごとに 1 つを用いてその他を間引くことで 4 fs の時間刻み幅にすることができる。

本研究では、シミュレーションの間引き時間幅とマルコフ性の階数 K の両方を同時に調整することを避けるために、マルコフ性の階数 K を 64 と先に決定して、シミュレー

シヨンの間引き時間幅を調整する．ここで，調整パラメータとしてステップスキップ R を導入して，MD シミュレーションのから R 個ごとに 1 つを用いてその他を間引くことにする．すなわち，MD-GAN に入力される部分系の時間刻み幅は $2R$ fs となり， K が 64 であるから，系列長さ $128R(= 2RK)$ fs がマルコフ性として必要な直近状態であると考えている．また，これにより，必要となる短時間シミュレーションの系列長さは $256R(= 2K \times 2R)$ fs となる．なお，予備実験において，ステップスキップ $R = 11000$ 以上のときに，MD-GAN による予測結果と元の MD シミュレーション計算結果の比較として，後述する統計量の平均二乗変位が一致した．よって，本研究では $R = 11000$ として計算を行った．

また，部分系については，ポリマー分子の分子中心粒子の座標を部分系とする．前述の通り，ポリエチレン溶融体シミュレーションにおいては分子は互いに同じ統計的性質を持っていると考えて差し支えない．よって，1 シミュレーション中に含まれる 300 分子について，300 個のサンプルを集めることができる．

第二段階として，集めた部分系のダイナミクスのサンプル集合に基づいて，MD-GAN を学習する．MD-GAN の学習に際して，モデルが過学習を起こさないように途中で学習を打ち切るアーリーストッピングを採用した．MD-GAN の潜在変数の次元は 4 とした．その他の設定は GAN の節の通りである．第三段階として，学習した GAN から長時間予測を得る．長時間予測は，絡み合いの効果が薄れていく遷移領域と，その後のブラウン運動をする領域までを予測したいため，約 10^5 ps までの長期予測を行った．これを繰り返すことで，十分な量の部分系のダイナミクスを得た．第四段階として，長期予測した部分系ダイナミクスの集合から，ポリエチレン溶融体の分子中心の平均二乗変位を計算する．MD-GAN によって分子中心のダイナミクスが正しく予測できていれば，求めた平均二乗変位は MD シミュレーションにおいて長時間シミュレーションを行った場合と一致するはずである．

MD シミュレーションによる平均二乗変位と MD-GAN によって求めた予測結果による平均二乗変位を図 5.7 に示す．この図では，前者は黒破線，後者は緑実線で平均二乗変位をプロットしており，短時間 MD シミュレーションから計算できる時間スケールを青色，それ以降の時間スケールを赤色で塗りつぶしている．従って，赤い領域はもとの短時間データからは計算できない，MD-GAN の予測によってのみ求められるものであると言える．この図から，MD シミュレーションによる平均二乗変位（黒破線）は， 10^3 ps までは $t^{0.5}$ ， 10^4 ps 以降は t^1 の拡散をしており，その中間は遷移領域となっている．これはラウスモデルとレプテーションモデルから得られる理論的予測（式 5.18）に合致していることに注意されたい．また，MD-GAN の学習に用いている訓練集合は，遷移領域の中ほ

どまでとなっており、その先の遷移領域とブラウン運動領域が正しく一致するかどうか
が重要である。緑実線の予測結果を見ると、遷移領域と通常拡散領域の双方において予測結
果がよく一致していることが分かる。よって MD-GAN による予測が成功した。

最後に、MD-GAN によってどの程度高速化が達成されたかについて述べる。図 5.7
をプロットするための計算コストとして、MD シミュレーションでは、 4.0×10^4 ps
(= 40 ns) のシミュレーションが必要である。ポリエチレン溶融体のシミュレーション
は、Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz x 16 スレッド並列を使用した場合、
1 ns あたり 3 時間の計算時間が必要であった。よって MD シミュレーションのみを用い
ると 120 時間の計算が必要であることになる。一方、MD-GAN を用いた場合、短時間
シミュレーションの必要量の系列長さは、 $R = 7000$ のとき 1792 ps (= 1.792 ns) であ
るから、およそ 5 時間の計算が必要である。それに加えて、MD-GAN は学習と予測が
必要であるから、これらも所用時間として考慮する必要がある。MD-GAN の学習には、
NVIDIA GeForce GTX 1080 を用いた場合、およそ 3 時間かかり、予測は 1 分以下で終
了した。従って、MD-GAN による予測では合計 8 時間要したことになる。これと MD
シミュレーションを比べれば、15 倍の高速化に成功したと言える。

もちろん、MD 計算や機械学習モデルの学習は、使用する計算環境によって大きく依存
することに注意する必要がある。特に機械学習モデルについては、近年、並列化による数
桁オーダーの高速化が達成できることが報告されているから、MD-GAN の学習にかかる
時間については大きく短縮される可能性がある。一方、MD シミュレーションについて
は、逐次計算であることから、高速な計算機を用いたとしても、時間長さに比例した計算
時間がかかるだろう。従って、高速化の度合いとしては、MD-GAN によってさらなる倍
率での高速化が達成できると考えられる。

また、本研究で議論しなかった点として、今回の MD シミュレーションは初期状態とし
て既に平衡状態が得られている状態を仮定しており、MD シミュレーションの平衡化計算
に必要な時間については無視している点がある。ポリマー分子の平衡状態の取得には、
全く平衡化していない初期状態から始めて、MD シミュレーションでは通常かなり長い計
算時間 (1 日~1 週間) が必要である。平衡状態の計算の高速化については、原理的に平衡
状態を高速にサンプリング可能であるはずの MC シミュレーション手法の発展を期待し
たい。その方針の一環として、後述の章で MC シミュレーションの高速化手法の検討を
している。

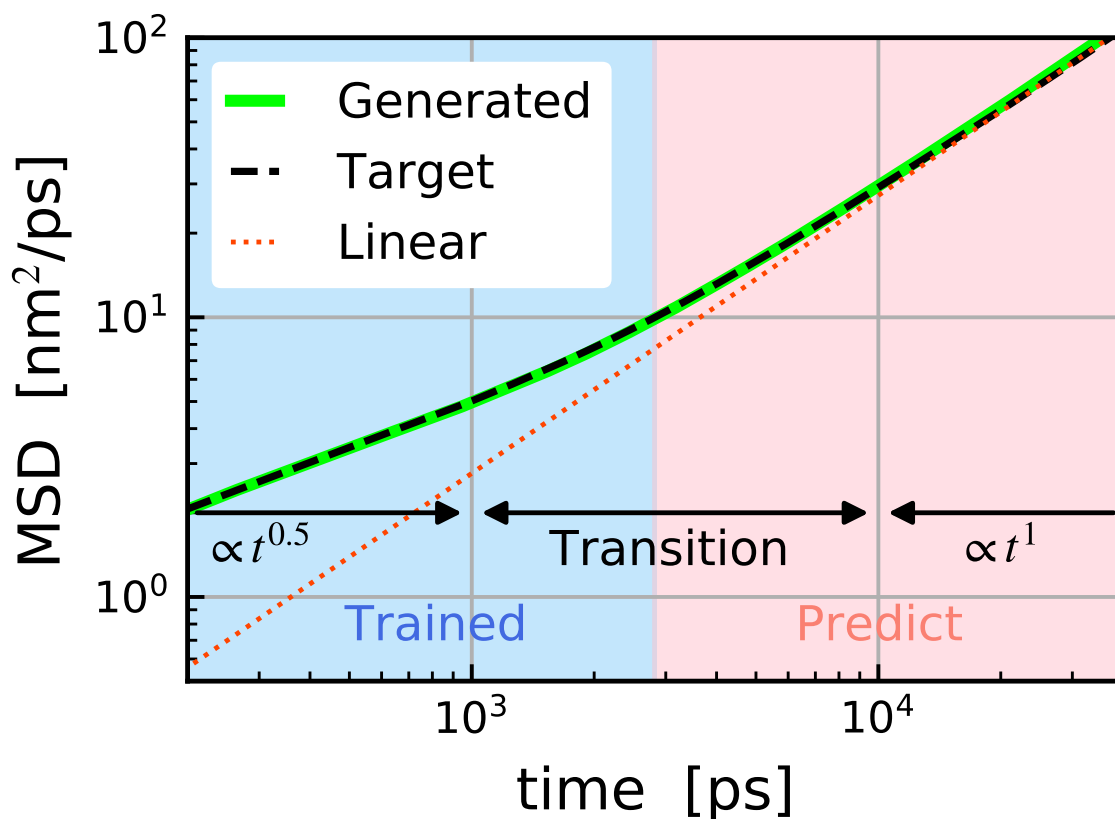


図 5.7 ポリエチレン溶融体シミュレーションの MSD 長期予測結果. MD シミュレーションによる真の MSD (黒破線) は, 10^3 ps までは $t^{0.5}$ スケール, 10^4 ps 以降はブラウン運動による t^1 (線形) スケール, その間は遷移領域となっていることが分かる. 赤点線は線形スケールをフィッティングして漸近線を引いたものである. MD-GAN による予測結果 (緑実線) は, 遷移領域の半ばまでしか学習データが与えられていない (青領域) のにもかかわらず, それ以降 (赤領域) も含めて真の MSD と良く一致していることが分かる.

5.5 結論と今後の展望

本章で行った研究について, 結論と今後の展望を述べる.

5.5.1 結論

本章では、MD シミュレーションにおいて部分系の確率的時間発展による計算高速化について調べ、以下の通り報告した。

1. 部分系のみを選択して確率的な時間発展を行うための理論的整理を行った。
2. 確率的な時間発展を行う機械学習モデルを構築し、さらに累積誤差 (エクスポージャー・バイアス) を大幅に抑える機械学習モデルとして MD-GAN を提案した。
3. その結果、ポリエチレン溶融体シミュレーションの平均二乗変位の計算を 15 倍高速化することに成功した。

5.5.2 さらになる発展：拡大した部分系によるポリエチレン溶融体の長期予測における MD シミュレーション必要長さの低減について

確率的ダイナミクスの長期予測可能性の節で議論した通り、部分系の選び方によってはマルコフ性の時間依存スケールを小さくすることができると考えられる。本章にて述べた主となる研究のさらなる発展として、ポリエチレン溶融体シミュレーションにおいて、部分系の拡大によって短時間 MD シミュレーションの必要量が低下することを発見したことを報告する [61]。

前述の設定では、部分系として分子中心の座標だけを選び、確率的時間発展を予測した。分子中心だけではなく、対象とする分子内の他の粒子を部分系として含めると、マルコフ性の階数が減少するのではないかという仮説を立てた。これを検証するために、分子中心に加えて同分子内の他の 4 か所の粒子を部分系に含めて、他は全く同じ条件で MD-GAN による予測を行った。すると、 $R = 4000$ の場合でも正しくポリマー分子の平均二乗変位を求めることができることが明らかになった。このことから、部分系の選び方によってはマルコフ性の時間依存スケールを小さくすることが示された。

今後の展望として、絡み合いの効果の大きな要因である分子同士の相互作用について考慮することができれば、より MD シミュレーションの必要長さを減らすことができないか、という点を検討したい。分子同士の相互をとらえるためには、対象とするポリエチレン分子だけではなく近接する他の分子の情報についても部分系に含める必要があるが、単に他の分子の座標としてしまうと、時間経過とともに対象分子から遠く離れていってしま

う問題があり，MD-GAN への入出力を工夫する必要があるだろう．

第 6 章

深層学習を用いた MD シミュレーション結果を自動解析する手法の開発

本章では、MD シミュレーションによって得られたデータを半自動的に解析するための手法について調べた研究を報告する。

6.1 本章の目的と概要

序論で述べた通り、MD シミュレーションを実行したあとに得られる出力は、単なる大量の分子配置の時系列データである。これらのデータから有意義な知見を取り出すためには、計算対象に対して深い専門知識を持つ専門家が、解析したい対象に合わせて適切な統計量を設計する必要がある。しかしながら、シミュレーションデータには本来、統計量を含む様々な情報が始めから含まれているはずであり、むしろシミュレーションデータを統計量に変換することによって、データに含まれる有意義な情報をほとんど落としてしまっている可能性がある。むしろ、シミュレーションデータから自動的に有意義な情報を取り出すことができれば、今まで専門家の目に入らずに見逃されていた知見が得られる可能性がある。

そこで本研究では、「ある系を特徴づけるものとは、それ以外の系と比べて違っている点である」という考えのもと、少しずつ異なる複数の系の MD シミュレーションを行って、その結果から、事前知識なしに系を特徴づける違いを識別し、かつその違いが様々な系の中でどういった立ち位置にいるのかを可視化する。この考え方は、MD シミュレー

シオン対象とする様々な系に適用可能であり、高い汎用性を持つと考えられる。

これを実現するために、本研究では、複数の系の MD シミュレーションデータから、それらがどの程度違っているか、また、何が差異をもたらしているかについてを調べる機械学習モデルを構築している。MD シミュレーションはシミュレーション結果全体が非常に膨大なデータ量 (1 GB~) となり、直接シミュレーションデータ全体を直接取り扱うのは事実上不可能である。そこで、MD シミュレーションの高速化の章 (5 章) で扱ったように、部分系の一定の長さのダイナミクスが従う確率を考えている。本章ではこの確率分布に特に着目しているため、これをローカルダイナミクスアンサンブル (Local Dynamics Ensemble, LDE) と呼び、本章のはじめに LDE の定義について述べている。また、部分系ダイナミクスの個々のサンプルではなく、その確率分布としての LDE に着目することで、系の些細な違いを検出できるようにしている。

LDE の定義の後に、LDE の違いを測定する機械学習モデルについて述べている。提案する機械学習モデルでは、LDE 間の Wasserstein 距離を深層学習モデルによって求め、各系間の距離を低次元空間に埋め込むことで、各系がどのように異なっているかの可視化が可能である。これにより、各系の違いや近さを持っているかなどの全体像を知ることができる。

さらに、Wasserstein 距離を求めるために学習した機械学習モデルから、Wasserstein 距離の大小に貢献した個別のダイナミクスサンプルを抽出する方法についても示している。これは LDE としてまとめられたサンプル集合に対して、ひとつひとつのサンプルに特徴ラベルを付けることができるということである。前述の低次元空間の可視化とも併用することで、埋め込み形状の原因となっている特徴的な部分系のふるまいを識別することができる。

上記のプロセスを実行すると、部分系さえ決定してしまえば、専門的知識を持たない者であっても半自動的に解析が行える。その次の節では、これらの自動解析する手法の検証として、アミノ酸水溶液を対象として行った検証実験について述べている。本実験では、20 種類のアミノ酸がそれぞれ溶けた水溶液および (対照としての) 水分子のみの系、の合わせて 21 の系について MD シミュレーションを行い、本手法を用いて、低次元空間への埋め込みおよび埋め込みに基づく距離要因のサンプルの特定を行った。その結果、アミノ酸分子に近い分子ほど水分子のみの系と異なる性質をもったダイナミクスとなることが確認された。これは先行して行われた研究結果 [62, 63, 64] と一致しており、よって、これらが本手法によって事前知識なしに可視化されたことは有意義であると考えられる。最後に、結果の考察と今後の展望について述べている。

6.2 ローカルダイナミクスアンサンブル (LDE) について

MD シミュレーションの結果全体を 1 サンプルとして機械学習に入力するようなモデルは、1 サンプルのサイズが非常に膨大であり、かつ学習のために十分なサンプル数を集めることが困難である。そこで、本研究では系全体のうち的一部分、すなわち部分系について、一定の長さのダイナミクスだけを切り取って 1 サンプルとすることで、機械学習に入力することを可能とする。

部分系の例としては、前章で述べたポリエチレン溶融体系におけるポリマー分子を 1 分子だけ取ってきたり、あるいはタンパク質とリガンドの複合体シミュレーションにおけるタンパク質において各アミノ酸の重心座標を部分系として選んだりすることができる。この部分系の選び方は、以降の半自動解析の質に大きく影響するため、実用的には様々なパターンを試すと良いと考えられる。

さて、部分系の一定の長さのダイナミクスを切り取ることを考えると、始点とする時刻によって異なったサンプルが得られることになる。これらのサンプルは、ある固有の確率分布に従うと考えられる。これを、局所的なダイナミクスの従う統計集団という意味で、ローカルダイナミクスアンサンブル (LDE) と呼ぶことにする。すなわち、LDE とは、サンプルの従う確率分布のことである。

ここで、部分系ダイナミクスの個々のサンプルではなく、その確率分布としての LDE に着目している理由は、個々のサンプルだけ見ても、頻度の違いが分からないためである。統計力学的には、あらゆる状態は（実現確率の大小はあれど）実現する可能性がある。特に系のマクロ量に些細な違いしかないときは、十分な量のサンプル集合が与えられれば、一方に含むものは、多少の頻度の違いはあれどほとんど同じものがもう一方のサンプル集合に含まれているだろう。従って、些細な差しかない系においては、部分系を比較するための対象として、ダイナミクスの従う確率分布自体に着目することが重要である。

LDE に従うサンプルとは、厳密には次のようになる。いま、解析の対象となる系が N 個あるとして、これらを順に $S_0, S_1, S_2, \dots, S_{N-1}$ とする。ある系 S_i について、平衡状態を初期状態とした MD シミュレーションを実行し、全体系の長時間ダイナミクス $\mathbf{X}_{0:L-1} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{L-1})$ を得たとする。ただし \mathbf{x} は全体系の状態である。次に、全体系の状態から部分系の状態を抽出する関数 $\mathbf{y} = h(\mathbf{x})$ を与え、全体系のダイナミクス $\mathbf{X}_{0:L-1}$ の各時刻に $h(\mathbf{x})$ を適用して部分系の長時間ダイナミクスを得る $\mathbf{Y}_{0:L-1} = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{L-1})$ 。さらに、ダイナミクスを固定長さで切り取って、系 S_i の LDE を 1 サンプル得る。切り取り方については、ある時刻 s から p ステップごとに l 個

分, すなわち $\mathbf{Y}_{s:s+p(l-1):p} = (\mathbf{y}_s, \mathbf{y}_{s+p}, \dots, \mathbf{y}_{s+p(l-1)})$ と切り取ることにする. すなわち, 切り取った部分系のダイナミクス $\mathbf{Y}_{s:s+p(l-1):p}$ の従う確率分布

$$p(\mathbf{Y}_{s:s+p(l-1):p}) \quad (6.1)$$

が LDE であり, 系 S_i に加えて抽出関数とパラメータ s, p, l が与えられてはじめて LDE が決定されることになる. 結果的に, 系 S_i の MD シミュレーション一度につき系 S_i の LDE の 1 サンプルが得られるということになる.

もちろん, 通常 MD シミュレーションはその時間発展が時不変であるから, s として異なる値 s' を持ってきても

$$p(\mathbf{Y}_{s:s+p(l-1):p}) = p(\mathbf{Y}_{s':s'+p(l-1):p}) \quad (6.2)$$

が成り立つだろう. 従って, 起点時刻 s を気にせず, どの起点時刻から切り取ってきても LDE としては同一視できる. 従って, 実質的にはパラメータは LDE サンプル長 l および間引き幅 p だけ与えればよい. これは, 起点時刻 s をどれをとっても LDE のサンプルが得られるということであるから, MD シミュレーション一度につき LDE のサンプルが $(L - pl)$ 個得られるということになり, データ数の増加も見込める (MD シミュレーションを何度も実行せず済む).

また, MD-GAN の際と同様に, 部分系の選び方によってはひとつの起点時刻 s から複数のサンプルが得られることもあるだろう. 例えば, 水分子のみからなるバルク系であって, ある特定の水分子を部分系に選んで LDE を構成したとする. この場合, 他の水分子を部分系として選んだ場合の LDE と元の LDE は統計的に区別がつかない. したがって, 結局どの分子を選んでも LDE に従うサンプルになる.

気を付けるべき事項として, 部分系の抽出関数 $h(\mathbf{x})$ は, 系によらず調べたい対象に対して一貫している必要がある. これは, 各系が LDE のサンプリングのため異なる抽出関数を使用していると, 元から違うものなので比較する意味がないことになってしまう可能性があるということである. また, 系によって扱う分子種が異なるときには特に注意が必要である. たとえば, 系 S_1 がメタノール水溶液, 系 S_2 がエタノール水溶液であるとき, 系 S_1 の部分系としてメタノール分子の重心座標, 系 S_2 の部分系としてエタノール分子の重心座標を選ぶとすると, これは違う分子種のダイナミクスを比較することになり, 分子量や分子形状が違うのであるから, ダイナミクスが違って当然のものを比較することになってしまう. もちろん, 複数の分子種があって, その分子の重心のダイナミクスが傾向として近いのか遠いのかどうかを識別するという場合であればよいかもしれないが, メタノール (エタノール) が水に溶けたことによる系への影響を調べたいのであれば, 系に共通する分子として水分子を部分系に選ぶのが妥当であると考えられる.

6.3 LDE 間の距離の測定

LDE の違いを調べる方法と、それを実現する機械学習モデルについて述べる。

6.3.1 LDE 間の距離とは

前節にて各系 $S_0, S_1, S_2, \dots, S_{N-1}$ それぞれに対して LDE を構成し、LDE ごとにダイナミクスのサンプル集合を得た。次に、LDE 同士の比較を行う。LDE の比較では、まず LDE がお互いにどれだけ違っているかについてを知る方法が必要である。LDE はダイナミクスの確率分布であるから、確率分布を比較するものでなければならない。そこで本研究ではまず、2 つの LDE 間の違いとは、2 つ LDE の従う確率分布の距離であると考えた。

また、LDE の比較のための確率分布として、特に Wasserstein 距離を選んで用いることにする。前章にて述べた通り、Wasserstein 距離は確率分布同士の距離のひとつである。確率分布同士の距離としては、他にも KL ダイバージェンス^{*1}や L^2 距離などがある。本研究では、後述する通りではあるが、Wasserstein 距離の計算のしやすさと、距離の計算に用いた深層学習モデルから距離の大小の要因となるサンプルを特定できる手法を開発したため、Wasserstein 距離を選んでいる。よって、各 LDE 間の Wasserstein 距離を計算する。比較したい系が N 個あれば、 $N(N-1)/2$ 通りの距離を計算する。これは系の数に対して二次的に増えていくが、通常、MD シミュレーションの計算コストが大きいので N はそれほど大きくなく、Wasserstein 距離の計算時間が支配的になることはないだろうと考えられる。

ここで、確率分布の距離を計算するというときに、確率分布の密度が与えられているのか、あるいは確率分布に従うサンプル集合が与えられているかによって距離の計算方法は全く異なるため、注意する必要がある。今回は、LDE はサンプル集合として与えられており、2 つのサンプル集合からそのサンプル集合に従う確率分布の距離を計算するものである。KL ダイバージェンスや L^2 距離をサンプリング集合から求める方法としてまず考えられるのは、各サンプル集合から LDE の確率密度を推定し、それらの推定量から距離の定義に従って近似計算する方法がある。またこれらと比べて近似精度が良い手法として、確率密度の推定を経由せず、2 つのサンプル集合から直接的に近似を得る方法も提案されている。たとえば、後者の手法のひとつである KLEIP 法 [65] (KL ダイバージェン

^{*1} KL ダイバージェンスは距離の公理を満たしていないので厳密には距離ではない

ス) や LSDD 法 [66] (L^2 距離) では, ガウスカーネル密度差モデルによって密度差推定量を求めてから, 距離を計算している.

今回使用する Wasserstein 距離は, 3 章に示した通りニューラルネットワークによる関数近似によってサンプル集合から効率的に求められることが知られているため, これを用いる. ここで, Wasserstein 距離の計算方法として一般的に用いられている手法として, Sinkhorn アルゴリズム [67] があるが, この手法はサンプル数 n と誤差 ϵ に対して計算量が $\mathcal{O}(n^2/\epsilon^3)$ であり, サンプル数に対して二次的に計算量が増えてしまう. 本研究では, MD シミュレーション一度につき LDE としてサンプル集合が数百万以上となるため, サンプル数に対して二次的に計算量が増えるアルゴリズムは不適であるから使用できない*2. 一方, ニューラルネットワークによる計算法は, 近似精度は不明確であるがサンプル集合のサイズに依存せず Wasserstein 距離を計算することができる*3. よって, 厳密性よりも解析に対して有用性があればよいこと, および, 似たようなサンプルが大量にあること, という本研究の趣旨を踏まえて, ニューラルネットワークによる計算法を採用する.

6.3.2 Wasserstein 距離の計算方法について

次に, Wasserstein 距離のニューラルネットワークによる計算方法を述べる. 計算の理論的根拠は 2 章および 3 章で述べた通りではあるが, 簡単にまとめると, 次のようである.

ある空間 \mathcal{X} を共通の状態空間とする 2 つの確率変数 A, B があり, その確率分布をそれぞれを $p(A)$ および $p(B)$ と書く. このとき, 確率分布 $p_A(A)$ と $p_B(B)$ の Wasserstein 距離 $W[p_A, p_B]$ とは

$$W[p_A, p_B] = \min_{\gamma \in \Pi} \sum_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) \quad (6.3)$$

のことである. ただし Π は $\sum_{\mathbf{y} \in \mathcal{X}} \gamma(\mathbf{x}, \mathbf{y}) = p(A = \mathbf{x})$ かつ $\sum_{\mathbf{x} \in \mathcal{X}} \gamma(\mathbf{x}, \mathbf{y}) = p(B = \mathbf{y})$ を満たすすべての非負関数 γ の集合である. 関数 γ は確率変数 A から B への輸送方法と呼ばれ, 輸送コスト $\sum_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y})$ が最も小さくなるように輸送方法を選んだ時の値を Wasserstein 距離と呼んでいる.

上記の Wasserstein 距離の計算は, 双対変換によって次のように書き換えられる.

$$W[p_A, p_B] = \max_{\|f\|_L \leq 1} \mathbb{E}_A[f(A)] - \mathbb{E}_B[f(B)] \quad (6.4)$$

*2 これが KLEIP 法や LSDD 法などのカーネル密度比推定による距離計算を利用しない理由でもある.

*3 これは経験的なものである.

ただし、 $\|f\|_L \leq 1$ は関数 f が 1-リプシッツ連続であることを表しており、1-リプシッツ連続関数 f の中で $(\mathbb{E}_A[f(A)] - \mathbb{E}_B[f(B)])$ が最大となるような関数を選んだ時の値が Wasserstein 距離である。

ニューラルネットワークによる Wasserstein 距離計算法では、関数 f をニューラルネットワーク $f(\boldsymbol{\theta})$ で構成し、学習によって $(\mathbb{E}_A[f(A)] - \mathbb{E}_B[f(B)])$ が最大となるような関数 $f(\boldsymbol{\theta})$ を近似的に求めることで計算する。すなわち、損失関数を

$$L(\boldsymbol{\theta}) = -\mathbb{E}_A[f(A; \boldsymbol{\theta})] + \mathbb{E}_B[f(B; \boldsymbol{\theta})] \quad (6.5)$$

とにおいて、損失関数が最小化するようなパラメータ $\boldsymbol{\theta}^*$ を学習によって求め、Wasserstein 距離の近似値を $W[p_A, p_B] = -L(\boldsymbol{\theta}^*)$ であると計算する。ただし、関数 f は 1-リプシッツ連続である必要があるから、One-centered Gradient Penalty を損失関数に加えることで、これを満たすように制約する。

上記の計算方法に基づき、各 LDE 間の Wasserstein 距離を計算する。比較したい系が N 個あれば、 $N(N-1)/2$ 通りの距離を計算することになるから、関数 f としてのニューラルネットワークモデル $f(\boldsymbol{\theta})$ も $N(N-1)/2$ 個違うものを用意して、それぞれ学習する。以降、系 S_i の LDE と系 S_j の LDE 間の求めた Wasserstein 距離を W_{ij} と表記し、かつその計算で求めたモデルを f_{ij} と表記することにする。ただし、 $f_{ij} = f_{ji}$ である。

6.4 LDE 間の距離の埋め込み

前節では、各 LDE 間の Wasserstein 距離をニューラルネットワークにより求める方法を述べた。次に、求めた Wasserstein 距離をもとに、距離を低次元空間に埋め込むことで、各系のお互いの近さや遠さを視覚的に表したい。そのためには、Wasserstein 距離 W_{ij} の組を低次元上の点 \mathbf{p}_i の組へ変換する手法が必要である。

本研究では、低次元上の点 \mathbf{p}_i 同士の距離が、なるべく Wasserstein 距離を再現するように埋め込むこととする。これは Multi-Dimensional Scaling (MDS)[68] の 1 種であり、次のように定式化する

$$\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1} = \underset{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}}{\operatorname{argmin}} \sum_{i,j} (W_{ij} - \|\mathbf{p}_i - \mathbf{p}_j\|)^2. \quad (6.6)$$

上式を満たす低次元上の点 \mathbf{p}_i の組を求めるために、損失関数を

$$L(\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}) = \sum_{i,j} (W_{ij} - \|\mathbf{p}_i - \mathbf{p}_j\|)^2 \quad (6.7)$$

とにおいて、最急降下法によって損失関数 $L(\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1})$ を最小化する点を求める。ただし、損失関数が凸関数であるとは限らないため、低次元上の点 \mathbf{p}_i の初期値をランダムに設定し、最急降下法による求解を何度か繰り返す。経験的には1度の求解で妥当な解が求まることが分かっている*4。

以上より、Wasserstein 距離を低次元空間に埋め込むことで、各 LDE がどのような傾向にあるのか全体像を知ることができる。

6.5 LDE 間の距離の増減に関わるダイナミクスサンプルの特定

さらに、これらの各系の違いを生み出した個別の要因について調べる。Wasserstein 距離の計算に使ったニューラルネットワーク f_{ij} から、さらに情報を抽出することを考える。そこで、ニューラルネットワークの学習に使った式を見直す。

ニューラルネットワーク f_{ij} と求めた Wasserstein 距離 W_{ij} には

$$W_{ij} = \mathbb{E}_A[f_{ij}(A)] - \mathbb{E}_B[f_{ij}(B)] \quad (6.8)$$

という関係があった。式 6.4 と比べて argmax が外れているのは、既に最適化された後の関数 f_{ij} を考えているからである。ここで、距離因子関数 $g(\mathbf{x})$ を

$$g(\mathbf{x}) = \mathbb{E}_B[f_{ij}(\mathbf{x}) - f_{ij}(B)] \quad (6.9)$$

と定義する。すると、式 6.8 は次のように書き換えられる

$$W_{ij} = \mathbb{E}_A[g(A)]. \quad (6.10)$$

上式は、Wasserstein 距離は関数 g の期待値になっていることを示している。これを解釈すれば、関数 $g(\mathbf{x})$ が大きくなるようなダイナミクスサンプル \mathbf{x} は Wasserstein 距離を大きくする方向へ貢献し、一方逆に、関数 $g(\mathbf{x})$ が小さくなるようなダイナミクスサンプル \mathbf{x} は Wasserstein 距離を小さくする方向へ貢献しているということを意味している。従って、各サンプルについて関数 $g(\mathbf{x})$ の値を調べることで、ダイナミクスのサンプル集合のうちどのサンプルが Wasserstein 距離を大きくする（あるいは小さくする）のに貢献したかを調べることができる。これにより、シミュレーション可視化ソフトウェアに

*4 なお、この問題の解は SMACOF algorithm[69] によって高速に求めることができることが知られているが、本研究においては勾配法による計算で十分な精度の解を素早く求めることができるため、用いていない。

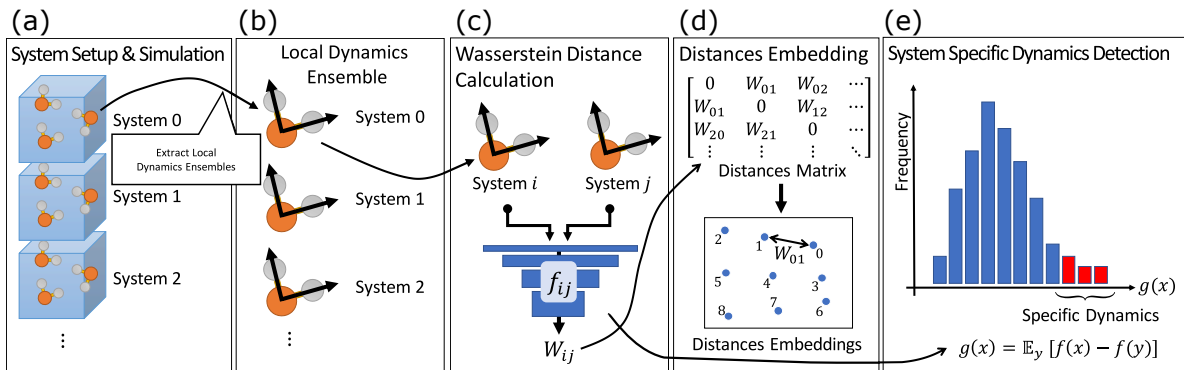


図 6.1 MD シミュレーション結果の自動解析手法の模式図 (a) 様々な系に対してシミュレーションを行って, (c) 部分系を抽出して LDE を構成し, (d) ニューラルネットワークによって LDE 間の距離を測定, (e) さらにその要因となるダイナミクスサンプルを特定する.

て関数 $g(x)$ の値の大小を表示すれば, どの分子が比較対象となる系との違いを生み出しているのかを効率的に調べることができる.

前述の低次元空間に埋め込みをもとに, 距離の大小の原因となっているサンプルを抽出し, 可視化することで, 系の違いを特徴づけるダイナミクスが何かを特定することができる. これまでの説明をまとめると, 本研究では次のように解析を行う.

1. (第 1 段階) 複数の異なる系において MD シミュレーションを実行し (図 6.1(a)), 調べたい現象に関わるとされる部分系を抽出することにより各 LDE のサンプル集合を得る (図 6.1(b)).
2. (第 2 段階) 各系の LDE について, それぞれの確率分布の Wasserstein 距離をニューラルネットワークモデルの学習によって求める (図 6.1(c)).
3. (第 3 段階) 求めた各系間の Wasserstein 距離を低次元空間に埋め込むことで, 系間の大局的な関係を可視化する. (図 6.1(d))
4. (第 4 段階) 第 3 段階の低次元空間に埋め込みをもとに, 系間の違いをもたらしている要因となる個別のダイナミクスサンプルを抽出し, 可視化する. (図 6.1(e))

6.6 アミノ酸水溶液に対する解析例

提案する自動解析手法を用いて, 実用的な MD シミュレーションの 1 つであるアミノ酸水溶液に対する解析例を述べる.

6.6.1 簡単な例を用いた提案手法の検証

以上の議論より、シミュレーション結果を自動解析する手法が構築された。本節では、アミノ酸水溶液に対して提案手法を適用して、どのような結果が得られるかを検証していくが、その前の予備実験として、違いが理解しやすい簡易データを用いて、どのように解析が進むかを確認していく。

まず、第1段階としてLDEの設定を行う。この予備実験では、比較対象となる系として9つの系 (S_1, S_2, \dots, S_9) があるとして、かつ各系に対応するLDEの共通のダイナミクスは、長さ64の確率変数

$$y_k = a\epsilon_k^{(0)} + b \sum_{i=1}^k \epsilon_i^{(1)} \quad (6.11)$$

に従うデータを用いるとする。ここでは、LDEのダイナミクスを直接与えているため、部分系の抽出は省略されている。ここで、 $\epsilon^{(0)} = (\epsilon_1^{(0)}, \epsilon_2^{(0)}, \dots, \epsilon_{64}^{(0)})$ および $\epsilon^{(1)} = (\epsilon_1^{(1)}, \epsilon_2^{(1)}, \dots, \epsilon_{64}^{(1)})$ はすべての成分が互いに無相関な標準正規分布であるとする。また a, b はそれぞれ各系に固有のパラメータであり、表6.1に示すように設定した。すなわち、このLDEは正規乱数の累積和によるランダムウォークと各時刻個別の正規乱数による無相関ノイズ成分をそれぞれ a, b の重みで合算したものとなっている。これらの9つの系のLDEから生成されるサンプル例を図6.2にそれぞれ示している。

ここで問題設定の注意すべき点として、各LDEについてパラメータ a, b およびそのパラメータからのダイナミクスの生成方法を与えているが、実際にデータが与えられて解析を始める時点ではこれらは全く分からない状況であることを理解する必要がある。つまり、系がどのように生成されて、どのようなパラメータが与えられていたかという状況がわからない状態から、本提案手法によって a, b という二つの自由度があつて、かつ a, b は何を変更しているパラメータであるのかを推定する必要があるということになる。

次に、第2段階として各LDE間のWasserstein距離を求める。9つの系があるため、合計 $36(=9 \times 8 / 2)$ 個の異なるニューラルネットワークを用意して、それぞれLDE間の距離を学習によって計算した。その結果を図6.3に示している。図6.3からは、特に S_1, S_2, S_3 と S_7, S_8, S_9 の間の距離が大きくなっていることが分かるが、具体的に系間がどのような自由度をもって分布しているかはわからない。そこで、第3段階として、求めたWasserstein距離を低次元空間に埋め込みを行う。図6.4は埋め込みの結果を示している。図6.4によると、埋め込みは 3×3 のグリッド状に整列して配置されていることが分かる。このことから、与えられた9つの系は、2つの自由度を持っていることが判明した。

実際、(本来はわからないものであるが)もともとの LDE を生成したパラメータ (a, b) を確認すると、埋め込みによって得られたグリッドはそれぞれ各自由度の方向へパラメータが順に並んでいることが分かる。これはすなわち、提案手法によって自動的に、埋め込みからパラメータ (a, b) の存在が (再) 発見されたということである。

最後に、第 4 段階として系間の違いをもたらす要因となるサンプルを抽出する。第 3 段階では、パラメータ (a, b) に対応する 2 つの自由度が存在することが分かった。第 4 段階では、これらの自由度がダイナミクスのどのような違いを生み出しているかについても自動的に解析する。まず、1 つ目の自由度として S_4 から S_6 方向に沿う自由度の違いを確認したい。そこで、系 S_4 と系 S_6 間の距離を測定するために学習したニューラルネットワークに基づき、距離因子関数 $g(\mathbf{x})$ を計算する。 S_6 の LDE からいくつかのサンプルを生成し、サンプルの距離因子関数 $g(\mathbf{x})$ を計算し、距離因子関数の値が上位 1% となったサンプルと下位 1% となったサンプルをそれぞれ図 6.5(a) および図 6.5(b) に示している。上位と下位のサンプルを比較してみると、上位サンプルは後ろの時刻で 2 又にわかれて絶対値が大きくなっており、下位サンプルは逆に後ろの時刻の絶対値が小さいものが得られていることが分かる。これは、系 S_4 と系 S_6 間の比較において、系 S_6 に近いサンプルの方が大きく拡散しているということを示している。すなわち、 S_4 から S_6 方向に沿う自由度は、拡散の大きさの違いを示しているということである。ここで、LDE のもともとの生成方法を確認してみると、系 S_4 と系 S_6 間はパラメータ b に違いがあり、パラメータ b はダイナミクスのランダムウォーク成分の強さを決定しているのであった。これは距離因子関数による上述の考察と一致しており、提案手法によって自動的にパラメータが定める影響を推定できたということである。

もう一方の自由度についても、パラメータが定める影響を推定してみる、系 S_4 と系 S_6 の方向とは垂直な方向を選ぶとして、ここでは系 S_1 から S_7 を比較した時の同様の方法で距離因子関数を調べた。距離因子関数の値が上位 1% となったサンプルと下位 1% となったサンプルをそれぞれ図 6.5(c) および図 6.5(d) に示している。すると今度は、パラメータ a に対応する、各時刻個別のノイズ成分の強さをの違いが上位と下位に現れていることが分かる。

よって、以上の予備実験により、複数の系の LDE から、各系が成す自由度と、各自由度の振る舞いの違いをほぼ自動的に推定することができるということが示された。

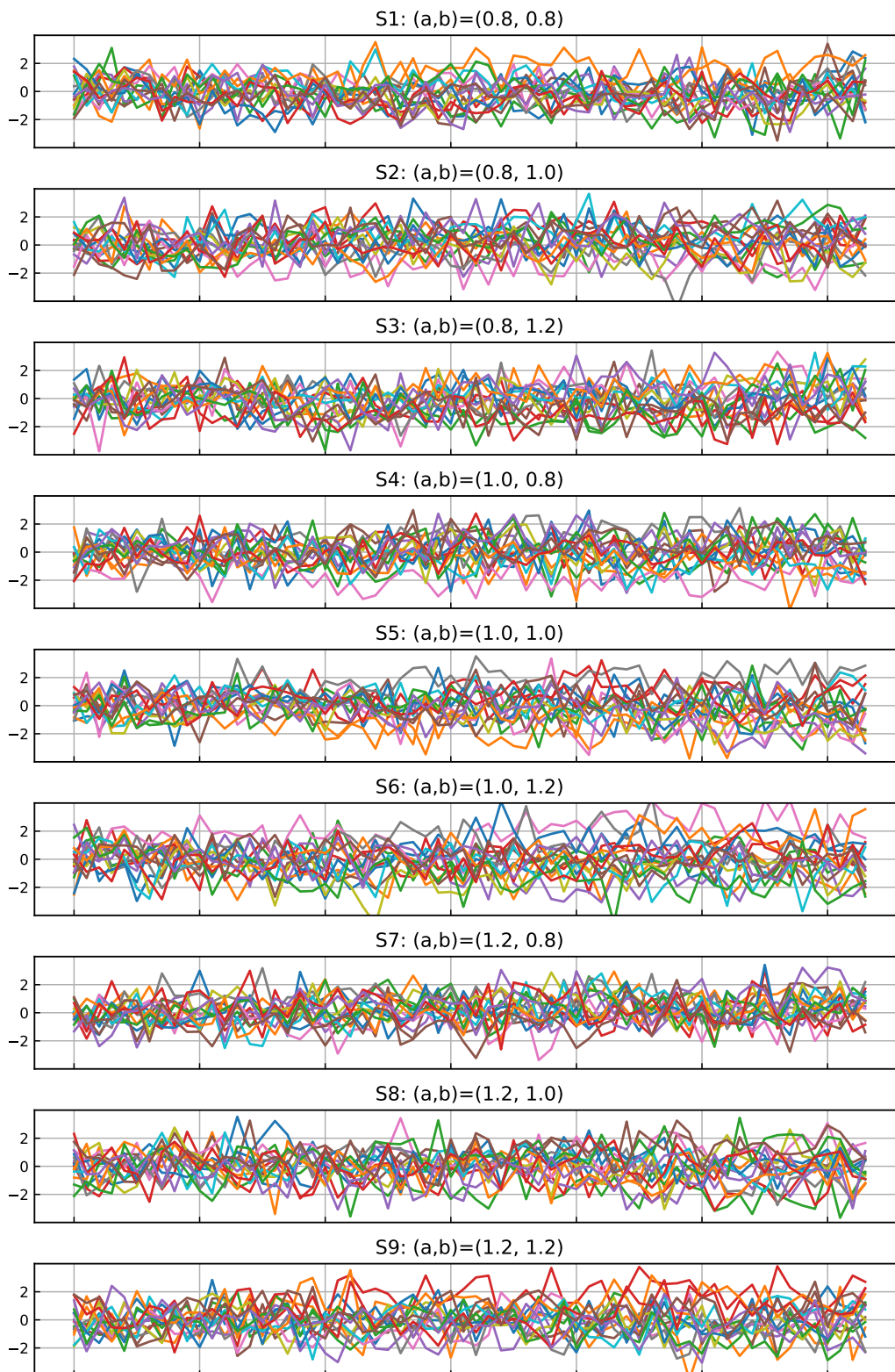


図 6.2 予備実験にて用意する 9 つの系のダイナミクスサンプル。

表 6.1 各系の LDE を定めるパラメータの一覧

系	a	b
S_1	0.8	0.08
S_2	0.8	0.10
S_3	0.8	0.12
S_4	1.0	0.08
S_5	1.0	0.10
S_6	1.0	0.12
S_7	1.2	0.08
S_8	1.2	0.10
S_9	1.2	0.12

6.6.2 アミノ酸水溶液の MD シミュレーション

提案手法を用いて、アミノ酸水溶液の解析を行う。まずは、アミノ酸水溶液の MD シミュレーションを実行する必要があるから、その詳細を述べる。

アミノ酸水溶液の MD シミュレーションは、表 6.2 に示す計算条件で行った。シミュレーションのためのソフトウェアとして GROMACS[59] を使用した。シミュレーションは、20 種の異なるアミノ酸水溶液と対照となる水分子のみのバルク系の合計 21 つの系に対して行っており、全て温度 300 K 一定、圧力 0.1 MPa 一定の NPT シミュレーションを行った。水バルク系を除く各系はそれぞれ系に固有のアミノ酸分子 40 個と水分子 800 個で構成されている。水バルク系については水分子 800 個のみで構成されている。アミノ酸分子は modified AMBER FF99SB-ILDN モデル [70, 71]、水分子は TIP4P-Ew モデル [72] によって表現されている。ただし、別種分子同士では、Lorentz-Berthelot 混合則が適用されている Lennard-Jones ポテンシャルが働いている。また、アミノ酸の部分電荷の値は Horn[73] によって報告された値を用いている。分子内の結合長と結合角は SHAKE 法によって拘束した。また、分子間の近距離相互作用は距離 1.0 nm でカットオフしており、長距離クーロン相互作用についてはパーティクルメッシュエワルド法によって計算した。シミュレーションの出力は、1000 step ごと (2 ps) に出力した。

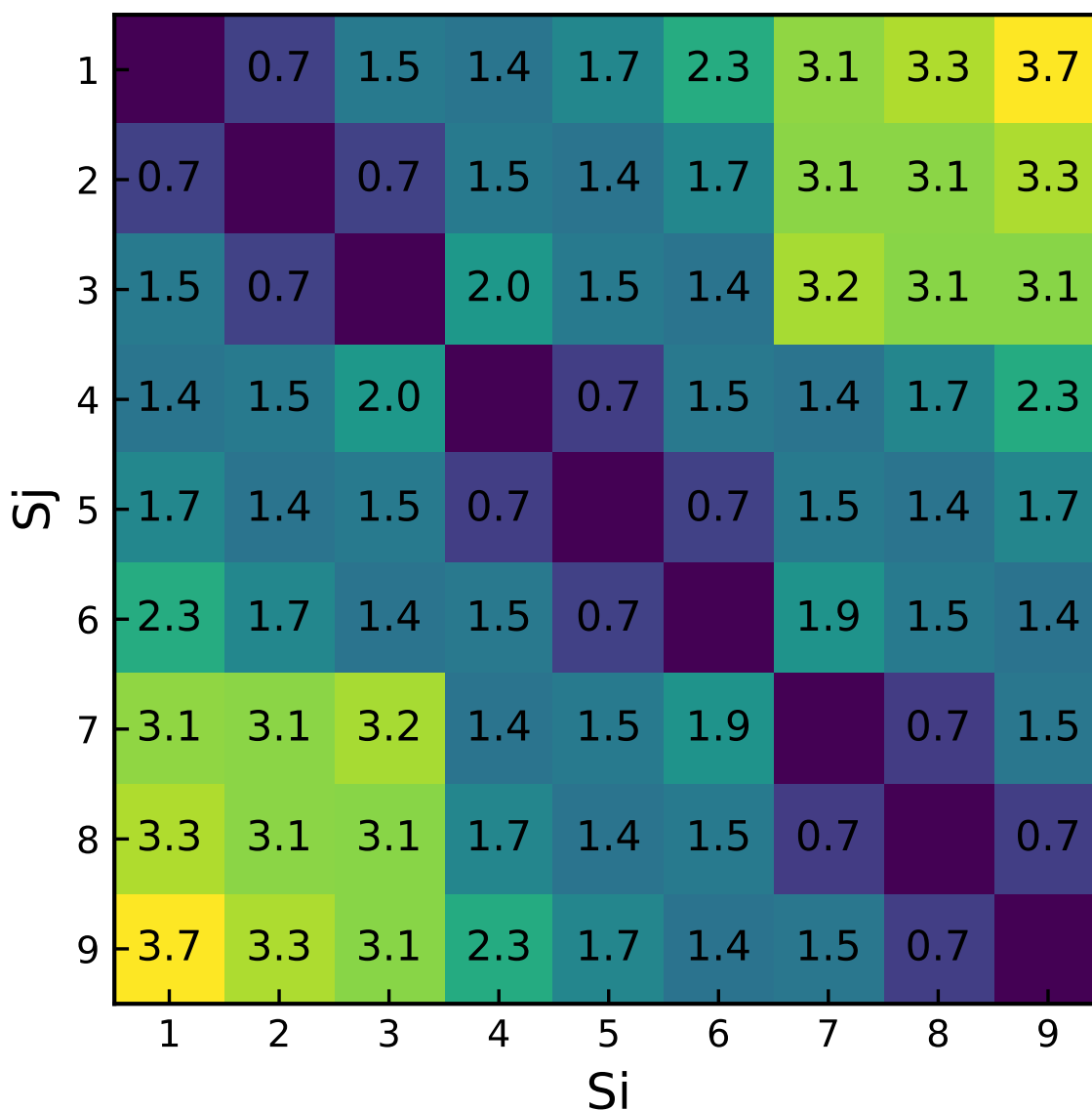


図 6.3 予備実験の 9 つの系の各ペアの Wasserstein 距離の予測結果.

6.6.3 提案手法による解析

上述の MD シミュレーションをもとに、アミノ酸水溶液の半自動解析を行う。まず、第 1 段階として、解析したい現象に合わせて部分系を抽出した。今回のシミュレーションでは、各系によって溶けているアミノ酸分子が異なるため、一貫した比較を行うためには水分子を選ぶ必要がある。よって、部分系の抽出関数の設定として、系に含まれる水分子を

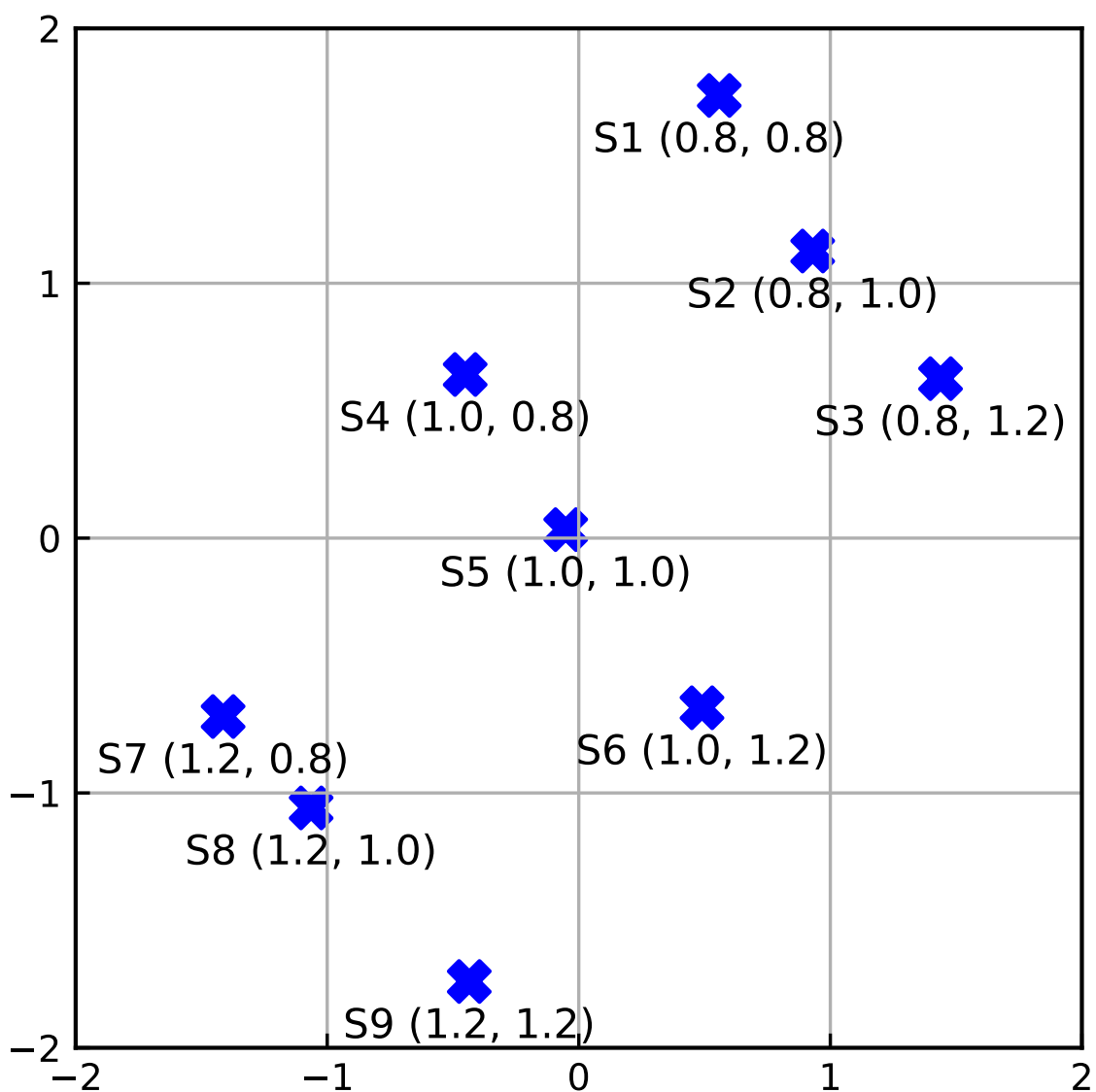


図 6.4 予備実験の 9 つの系の各ペアの Wasserstein 距離の 2 次元空間への埋め込み。

1 分子だけ選ぶことにする。ただし、シミュレーションの座標系の値を LDE のサンプルとしてしまうと、本来物理的に差がないはずの初期座標によって値が変動することになってしまう。これを防ぐため、部分系として、酸素原子を基準とした相対座標を採用する。本実験の水分子は、酸素原子 1 つ、水素原子 2 つとダミー粒子 1 つからなる TIP4P-Ew モデルを採用しているため、水素原子 2 つとダミー粒子 1 つの相対座標系によって部分系を

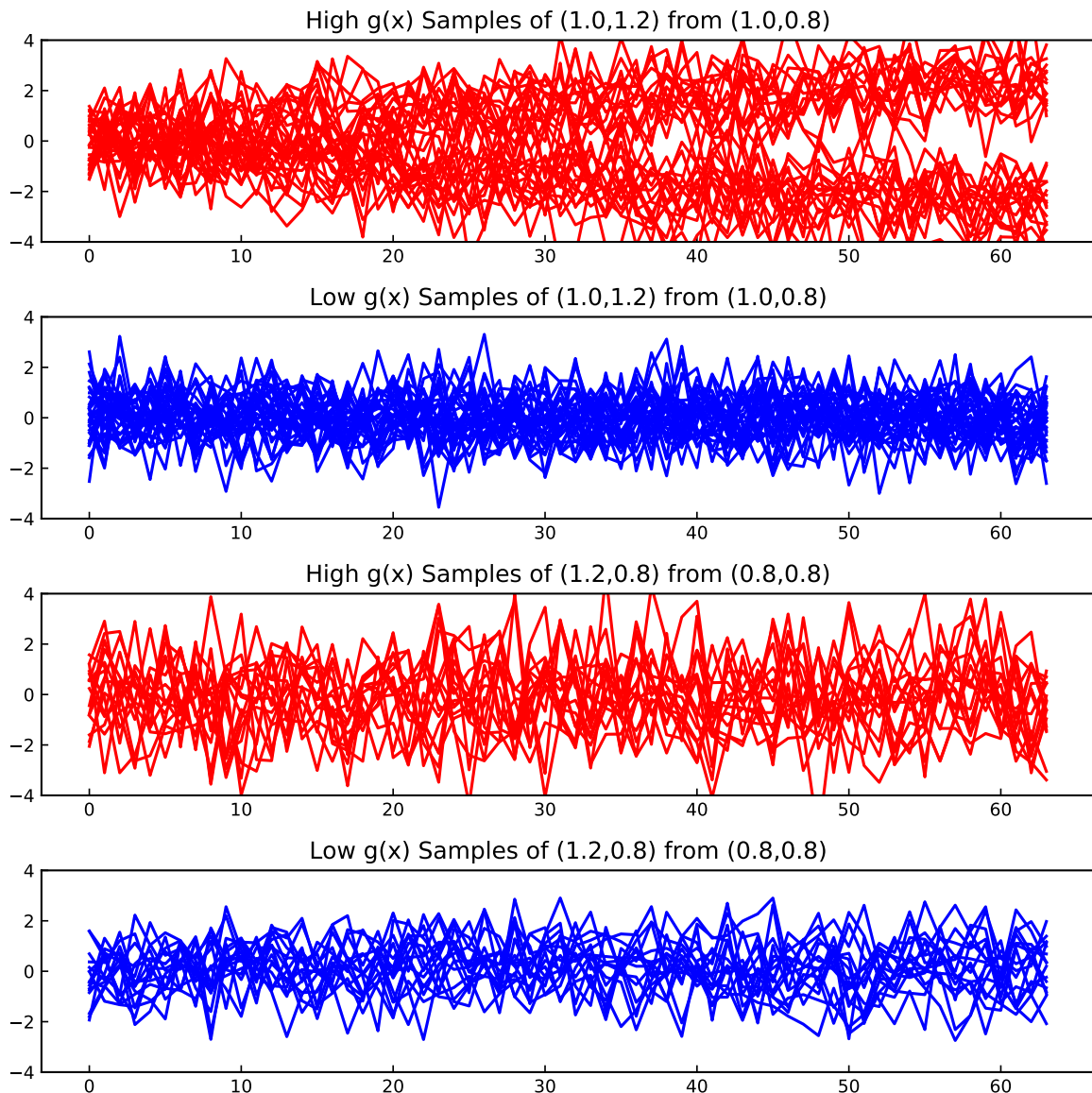


図 6.5 予備実験において、埋め込みから観察された 2 つの自由度の違いをもたらすダイナミクスサンプルの距離因子関数による抽出結果。

表現する^{*5}。よって、部分系は（時刻 1 つあたり）9 次元となる。合わせて、LDE のサンプル長 l と間引き幅 p も決める必要がある。本研究ではシミュレーションの出力間隔に合わせて間引き幅は 2 ps に設定し、サンプル長は 64 とした。従って、LDE が含むダイナミクスの時間長さは 128ps, サンプル 1 つの次元は 192 次元となる。

*5 自明な酸素原子の位置は除く。

表 6.2 アミノ酸水溶液シミュレーション条件

ソフトウェア	GROMACS
アンサンブル	NPT(Nose-Hoover, Parrinello-Rahman)
数値積分	Leap-Frog
分子モデル	modified AMBER FF99SB-ILDN(アミノ酸) & TIP4P-Ew(水分子)
温度	300 K
粒子数	水分子 x800 + (単一) アミノ酸分子 x40
時間刻み幅	2 fs

第 2 段階として各系同士ごとに Wasserstein 距離を推定するニューラルネットワークモデルを学習し、続けて第 3 段階として、求めた Wasserstein 距離を 2 次元空間に埋め込んだ。その結果、図 6.6(a) の埋め込みが得られた。埋め込みは 1 次元状に並んでいることが分かる。この 1 次元状の埋め込みの左端 (S_0) は水バルク系、右端 (S_{12}) はグルタミン酸水溶液となった。埋め込みが 1 次元状になったこと、および、左端 (S_0) は水バルク系となったことから、20 種類のアミノ酸水溶液系には自由度として 1 次元の違いしかなく、その自由度は水バルク系を基準としていて、アミノ酸によってその違いの強さに違いがあるものだ、というように考察できる。この考察は、LDE を決定してからほぼ自動的に得られたものであり、本手法が MD シミュレーションの解析に対して有効である理由のひとつであることを示している。

第 4 段階として、低次元空間に埋め込みをもとに、系間の違いをもたらしている要因となる個別のダイナミクスサンプルを抽出し、可視化する。本実験では、アミノ酸水溶液系には 1 次元の自由度しかないことから、この自由度の原因となるダイナミクスサンプルを調べたいと考えるのが妥当だろう。この自由度の原因となっているダイナミクスサンプルが最も顕著となる LDE の組は、低次元空間に埋め込みの両端にある S_0 と S_{12} であると考えられる。よって、 S_0 と S_{12} 間の Wasserstein 距離の学習に用いたニューラルネットワークから前述の距離因子関数 $g(\mathbf{x})$ を計算し可視化する。系 S_{12} の MD シミュレーションのあるスナップショットについて、各分子の $g(\mathbf{x})$ の値を計算すると、そのヒストグラムは図 6.7(b) となった。このヒストグラムの左端 (青)・右端 (黄) に位置するダイナミクスが、系 S_{12} の中でもそれぞれ系 S_0 の性質に遠いもの、または近いものとなっている。そこで、この両端のダイナミクスがどのような箇所に現れているかを可視化したものが図 6.7(e) である。赤がアミノ酸を表しており、青から黄のグラデーションで $g(\mathbf{x})$ の大小を表している。

このスナップショットの可視化の配置を注意深く観察すると、赤いアミノ酸分子に近い水分子が高い $g(\mathbf{x})$ 値をもち、離れたところにある水分子は低い $g(\mathbf{x})$ を持っていることに気付く (図 6.7(a) および (c)). よって、アミノ酸分子の周囲にいることによって、アミノ酸系における水分子の振る舞いが、水分子バルク系における水分子の振る舞いから大きく違っているということを示している. また、アミノ酸分子から離れている水分子は、水分子バルク系における振る舞いと似たものであることを示している.

さらに、特に高い $g(\mathbf{x})$ 値を持つ水分子と、特に低い $g(\mathbf{x})$ 値を持つ水分子に着目してダイナミクスを注意深く観察すると、特に低い $g(\mathbf{x})$ 値を持つ水分子では全天を自由に回転している様子が観察でき、逆に特に高い $g(\mathbf{x})$ 値を持つ水分子は、OH ベクトルが狭い領域に固定されて振動している様子が観察できた. これを視覚的にわかりやすく図に表したものがそれぞれ図 6.7(d) および (f) である. この図は、OH ベクトルの動きのトラジェクトリーを表示している.

ところで、アミノ酸分子によって、周辺の水分子の振る舞いが変わることは、既に先行研究で報告 [62, 63, 64] されているものである. その研究では、アミノ酸との相互作用で水分子の回転緩和がバルク水分子と比べゆっくりになることが報告されている. そこで本研究に用いた MD シミュレーションにおいて回転時間相関関数 [64]

$$C(t) = \left\langle \frac{3}{2} (\mathbf{u}(\tau+t) \cdot \mathbf{u}(\tau))^2 - \frac{1}{2} \right\rangle_{\tau} \quad (6.12)$$

を 21 種それぞれについて書いてみると、図 6.6(b) のようになった. ただし、 $\mathbf{u}(t)$ は時刻 t における水分子のダイポールモーメントである. ここで注目すべき点は、図 6.6(b) 回転時間相関関数の緩和曲線の順序と、図 6.6(a) の低次元埋め込みの順序が一致していることである. 図 6.6(b) の緩和が最も遅い系は S_{12} となっており、緩和が最も早い系は S_0 となっている. また、低次元埋め込みの第一主成分と、回転時間相関の時定数は、相関係数 0.99 で一致していることが判明した (図 6.8). よって、本研究による解析手法は、回転緩和について全く知らない状態から同じ結論を得られたのであるから、本手法が MD シミュレーションの解析に対して有効である理由のひとつであると考えられる.

6.7 結論と今後の展望

本章で行った研究について、結論と今後の展望を述べる.

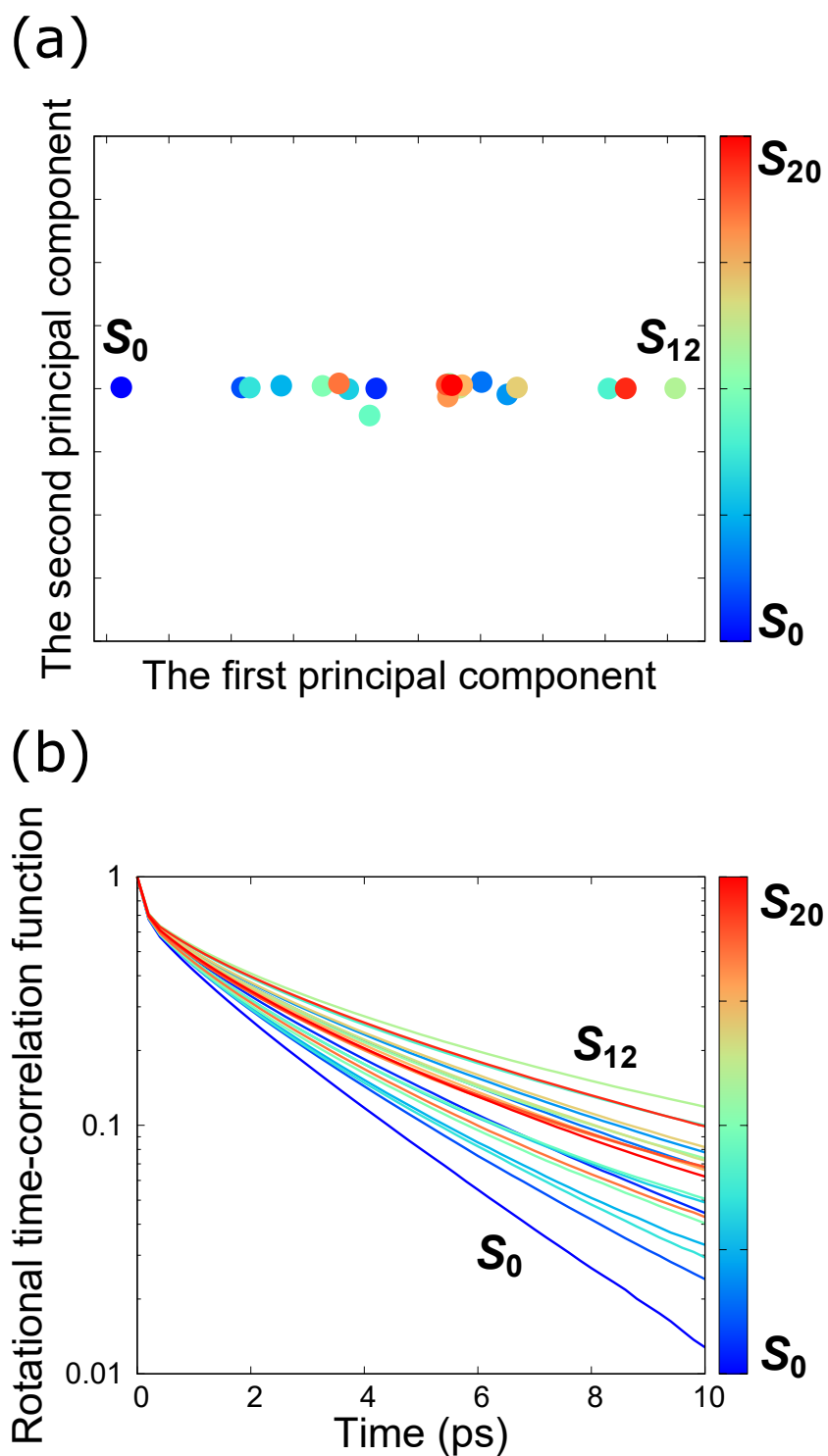


図 6.6 (a) アミノ酸水溶液系から構成した LDE の提案手法による埋め込み結果, および (b) 各アミノ酸水溶液の回転自己相関関数.

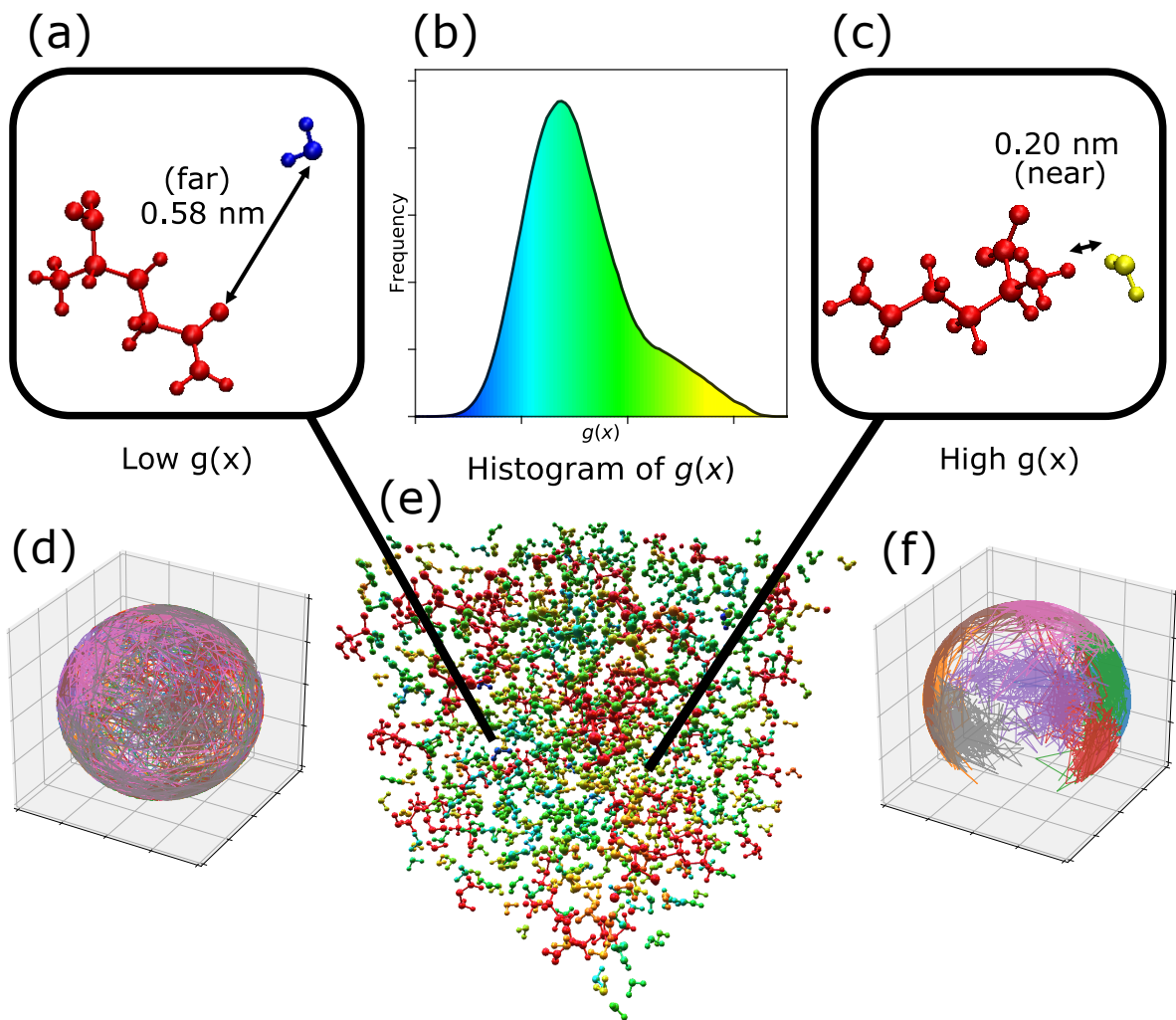


図 6.7 (b) 系 S_0 (バルク水) と比較した際の系 S_{12} のダイナミクスサンプルの距離因子関数値のヒストグラム, (e) 系 S_{12} のダイナミクスサンプルの距離因子関数を用いたスナップショットの可視化, (a,c) 高いまたは低い距離因子関数値をもつ水分子の静的な様子, (d,f) 高いまたは低い距離因子関数値をもつ水分子の動的な様子.

6.7.1 結論

本章では MD シミュレーション結果を自動解析する手法について調べ、以下の通り報告した.

1. 部分系間の比較をするために、局所ダイナミクスの確率分布 LDE を定義した.
2. LDE 間の比較を行うために、確率的距離を調べるニューラルネットワークを導入

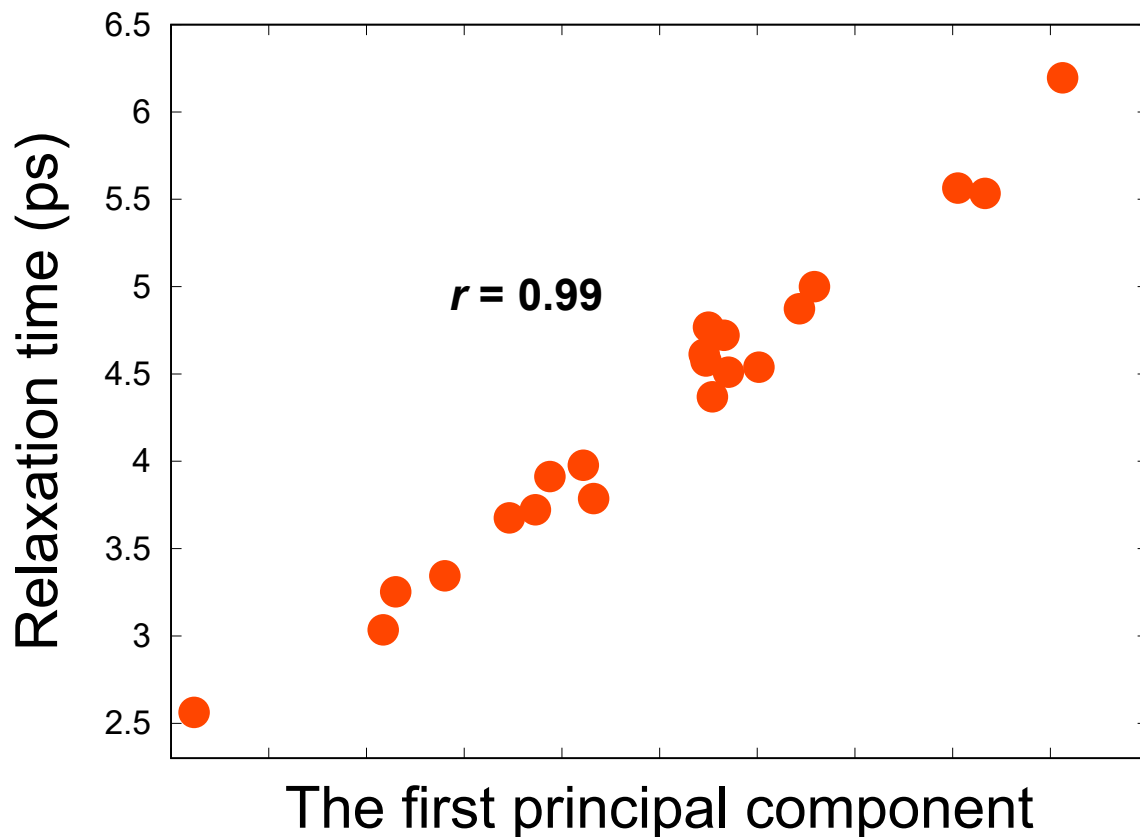


図 6.8 低次元埋め込みの第一主成分と回転時間相関の時定数の比較.

した.

3. 系同士の関係を視覚的にわかりやすくするために、確率的距離を埋め込みを導入した.
4. 系同士の関係において、どのダイナミクスが系に特徴的な振る舞いをしているかを調べる方法を導入した.
5. その結果、アミノ酸水溶液系において有用な解析を半自動で行えることを示した.

6.7.2 さらに発展：リガンド-タンパク質シミュレーション系に対する本手法の適用による自由エネルギーとの関連付けについて

本手法を他の MD シミュレーションデータセットへ適用することで、いまだ発見されていない知見が得られる可能性がある。本章にて述べた主となる研究のさらなる発展とし

て、蛋白質とリガンドの系に本手法を適用することでたんぱく質とリガンドの結合親和性を予測できる可能性を発見した [74].

創薬の分野では、タンパク質と結合しやすい化合物（リガンド）を低コストで発見したいという需要がある。この研究では、タンパク質とリガンドの結合しやすさは、結合時のリガンドの短時間ダイナミクスからもたらされているという仮説を立てた。この仮説を検証する道具として、本手法を適用した。複数種類のリガンドを用いた固定のタンパク質-リガンドのシミュレーションに対して、タンパク質のダイナミクスを LDE とした埋め込みを行ったところ、結合のしやすさの指標のひとつである自由エネルギー差と埋め込み表現が高い相関を持つことが明らかになった。また、特徴抽出因子を調べることによって、結合のしやすさをもたらしているダイナミクスを抽出することができた。このことから、本手法によってタンパク質とリガンドの結合親和性を低コストで判定できる可能性を示した。

第7章

機械学習による MC シミュレーション高速化法の開発

本章では、機械学習により MC シミュレーションを高速化する手法について調べた研究を報告する。

7.1 本章の目的と概要

序論で述べた通り、MC シミュレーションは統計力学から要請されるボルツマン分布をサンプリングすることで、理論上ダイナミクスとは無関係に平衡状態を得ることができると。一方、MD シミュレーションは平衡状態の取得のために、本計算とは別に、(平衡状態ではない) 初期状態から一定の間計算を流す平衡化計算が必要であり、この平衡化計算は本計算よりもずっと長い計算時間が必要である場合もある。したがって、MC シミュレーションによって直ちに平衡状態を取得できれば、MD シミュレーションと比べ計算コストに関して有利であると考えられる。また、ダイナミクスを知りたい場合でも、MC シミュレーションによって平衡状態を得てから、それを初期状態として MD シミュレーションを行うことで、MD シミュレーションの平衡化計算を不要とできる可能性もある。

しかしながら、MC シミュレーションを実用してみると、大きな問題にぶつかることになる。例えば、基本的で汎用的なサンプリングアルゴリズムの一つである Metropolis-Hastings (MH)[38] 法では、ボルツマン分布のサンプリングのために、直近の状態と似たような状態をサンプリングせざるを得ない状況に陥るため、十分に多様な平衡状態を得るためには (MD シミュレーションと同等かそれ以上に) 長時間の計算時間が必要になってしまう。この原因は、MC シミュレーションでサンプリングすべきボルツマン分布が、

エネルギー変化が激しい構造を持つ確率分布だからである。原子 1 つ 1 つのスケールで見ても、原子が衝突するような配置は非常に高いエネルギーを持ち、その結果このような配置は一切実現されない。そのような高エネルギーの状態を回避するためには、MC シミュレーションの 1 ステップでは粒子 1 つだけをランダムに選んで少しだけ移動する方法を取らざるを得ず、従って、直近の状態と似たような状態をサンプリングせざるを得なくなる。次節にこの現象を詳しく説明し、またこの現象を回避するための先行研究についても述べている。もちろんのこと、原子 1 つ 1 つのスケールではなく、もっと大きなスケールにおいても、例えばポリマーやタンパク質などの複雑な構造を持つ分子を含むような系では、より大局的に複雑なエネルギー分布となり、これがさらにサンプリングを難しくする。ただし、本研究では、そのような大きいスケールの複雑性を持つ系については研究の範囲外とする。よって、本研究では、上記のような原子の接近による高エネルギーの状態が生成されないようにしながらも、系の状態を大きく変化させることができるような MC シミュレーション手法を構築したい。

ところで近年、機械学習の分野で Continuous Normalizing Flow (CNF)[36] と呼ばれるモデルが提案された。このモデルは、ある既知の確率分布のサンプルを、1 階微分方程式によって変換し、その変換後のサンプリングがどのような分布に従うかを知ることができる手法である。CNF は深層生成モデルとしてすでに様々な適用例が研究されている [75, 76, 77]。本研究では、提案分布に対して原子の接近する配置される状態を取り除く変換を CNF を用いて施すことで、MH 法の 1 ステップに粒子の移動量を大きくすることを可能にできるのではないかと考えた。移動量を大きくしても粒子に衝突する割合が増えなければ、その結果として、MC シミュレーションの効率を改善することができるのではないかと考えられる。よって、本研究では、CNF を用いた分布変換によって近接配置のない提案分布を実現することで、MC シミュレーションの効率を改善することを目的とする。

そこで、まず CNF を MC シミュレーションへ適用できるようにするために、CNF を多体相互作用へ拡張した Continuous Normalizing Molecular Flow (CNMF) モデルを提案する。オリジナルの CNF は分子の座標を状態として入力した際に、平行移動不変性および回転移動不変性、置換不変性を持たず、分子系へ適用するのが難しい。CNMF では、これらの不変性をすべて満たしたモデルとなっており、かつ確率分布の変化の値が効率的に求められる形式となっている。加えて、CNMF を用いて提案分布を変換した場合のサンプリングアルゴリズム Molecular Flow MH (MFMH) アルゴリズムを述べている。

さらに、CNMF モデルの自由度である変形関数を決定する手法として、Inverse square flow (ISF) 法と Self-learning (SL)-MFMH 法の 2 つを提案する。両者とも、CNMF モ

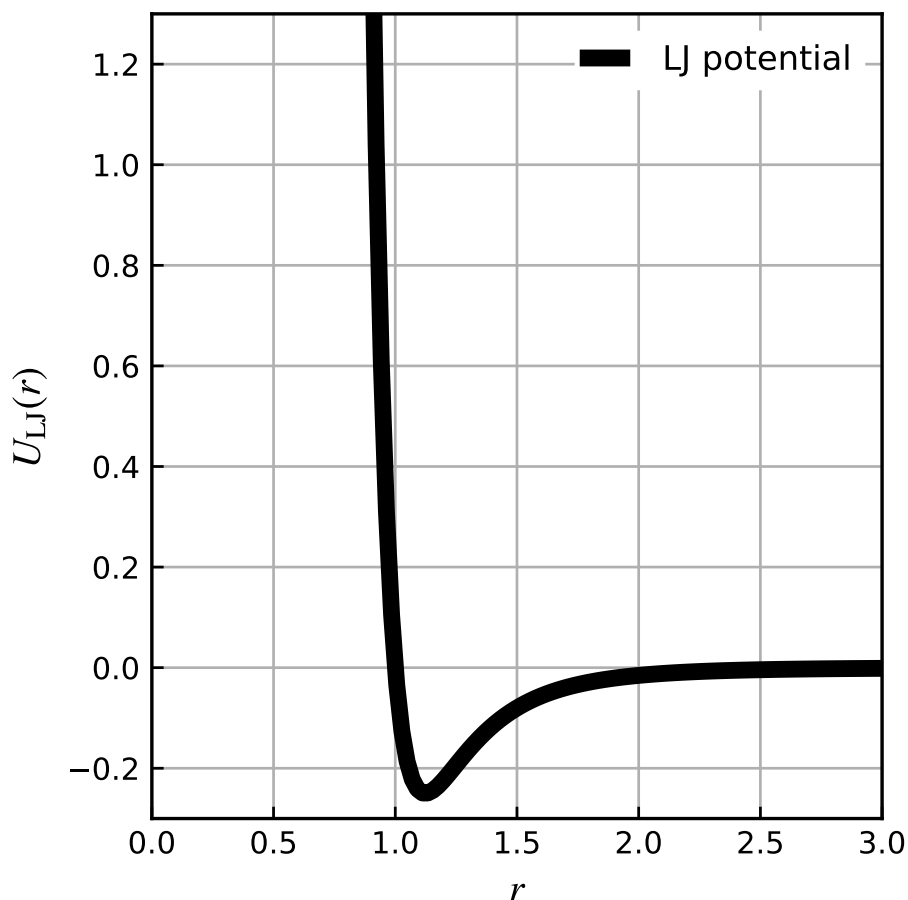


図 7.1 LJ ポテンシャルの様子。およそ $r = 0.9$ より内側で急激な増大を示している。

デルを基本としており、どのような変形を行うかについて異なっている。前者の ISF 法は、近接原子ペアを取り除きながらも、それ以外の状態は確率分布が一様に圧縮した分布となるようなものを解析的に求めたものである。また、後者の SL 法は、機械学習によって系に合わせた最適な変形関数を獲得する。これらの提案手法について、LJ 粒子系を実験対象として、ベースラインの MH 法によるサンプリングと合わせてサンプリングの効率を評価している。最後に、結果の考察と今後の展望について述べる。

7.2 サンプリングアルゴリズムの問題点と先行研究

統計力学より、カノニカルアンサンブルに従う場合のボルツマン分布は

$$p(\mathbf{r}) \propto \exp\{-\beta U(\mathbf{r})\} \quad (7.1)$$

と導出される。ただし $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ は系に含まれる全 N 粒子の座標、 β は系の無次元化逆温度であり、 $U(\mathbf{r})$ は系のポテンシャルエネルギーである。これはすなわち、エネルギーが高い状態ほど指数的に比例してその実現確率が小さくなるということである。分子シミュレーションの場合、ポテンシャルエネルギー $U(\mathbf{r})$ は基本的には 2 体間ポテンシャルエネルギーの和

$$U(\mathbf{r}) = \sum_{i < j} U_{ij}(\|\mathbf{r}_i - \mathbf{r}_j\|) \quad (7.2)$$

で表される^{*1}のであった。ここで、上式の 2 体間ポテンシャルエネルギー $U_{ij}(r)$ としては通常 LJ ポテンシャル

$$U_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (7.3)$$

が用いられることが多い (図 7.1)。この LJ ポテンシャルのうち粒子間距離の -12 乗に比例するエネルギーを持つ第 1 項は、パウリの排他律による斥力を表現しており、原子同士が接近するほど高エネルギーになることを意味する。距離の -12 乗というのは通常考える温度においては $r = 0.9$ 付近にかなり急峻なエネルギーの壁を持っているようなものである。ボルツマン分布はエネルギーが高い状態ほど指数的に比例してその実現確率が小さくなることを主張するわけであるから、原子がその壁を超えて内側に接近するのはあり得ないと言っても過言ではない。従って、MC シミュレーションにおいて、原子ひとつひとつのスケールで見ても、原子が (エネルギーの壁を越えて) 重なるような配置は実質的に許されないということである。

上記の検討を踏まえて、MH 法によるサンプリングについて考える。MH 法では、次の 2 段階のプロセスを繰り返すことで目的の分布 $p(\mathbf{r})$ からのサンプリングを達成する。

1. 目的の分布とは別にある、現在の状態 $\mathbf{r}^{(i)}$ をもとにして次の状態の候補を生成する提案分布 $q(\mathbf{r}^*|\mathbf{r}^{(i)})$ から、次の状態の候補 \mathbf{r}^* を生成する。
2. 得た次の状態の候補 \mathbf{r}^* を採択率 $A(\mathbf{r}^*|\mathbf{r}^{(i)})$ の確率で採択し、次の状態 $\mathbf{r}^{(i+1)}$ とする。ただし採択率 $A(\mathbf{r}^*|\mathbf{r}^{(i)})$ は詳細つり合い条件を満たすように、

$$A(\mathbf{r}^*|\mathbf{r}^{(i)}) = \max \left(\frac{p(\mathbf{r}^*)}{p(\mathbf{r}^{(i)})} \frac{q(\mathbf{r}^{(i)}|\mathbf{r}^*)}{q(\mathbf{r}^*|\mathbf{r}^{(i)})}, 1 \right) \quad (7.4)$$

とする。

^{*1} 分子内結合においては 3 体間以上の相互作用エネルギー項も含む場合も多々あるが本研究では省略する。

MH 法を用いた MC シミュレーションにおいては、提案分布 $q(\mathbf{r}^*|\mathbf{r}^{(i)})$ は $\mathbf{r}^{(i)}$ を平均とした正規分布や一様分布が選ばれることが多い。その場合、提案分布は \mathbf{r}_i を中心とした反転に対称な分布となるため、

$$q(\mathbf{r}^{(i)}|\mathbf{r}^*) = q(\mathbf{r}^*|\mathbf{r}^{(i)}) \quad (7.5)$$

が常に成り立つから、採択率は

$$A(\mathbf{r}^*|\mathbf{r}^{(i)}) = \max\left(\frac{p(\mathbf{r}^*)}{p(\mathbf{r}^{(i)})}, 1\right) \quad (7.6)$$

と簡単化できる。さらに、採択率にボルツマン分布を代入すれば

$$A(\mathbf{r}^*|\mathbf{r}^{(i)}) = \max\left(\exp\left\{-\beta[U(\mathbf{r}^*) - U(\mathbf{r}^{(i)})]\right\}, 1\right) \quad (7.7)$$

となる。

よって以上の計算から、MH 法の採択率は、現在の状態と比べて候補状態のエネルギーが大きければ大きいほど、そのエネルギー差に指数的に比例して採択率が小さくなるということが分かる。このことから、MH 法において提案分布 q が候補状態 \mathbf{r}^* を生成する際、エネルギー差が大きくなりすぎるようなものを生成しても採択されることはなく、無駄な計算コストを消費したことになる。前述の通り、ボルツマン分布は原子スケールでも原子同士が近くなる配置が生まれると系のエネルギーは非常に高い値になってしまう。これを回避するために、MH 法を用いた MC シミュレーションでは通常、(1) エネルギー変化が極力小さくなるように、ランダムに選んだ原子を一度に 1 つだけ動かすようにする、かつ (2) 原子同士の衝突を避けるために一度の移動量を低くする、という工夫を行う。よって、提案分布は、ランダムに選んだ粒子を微小に動かすような分布が選ばれる。結局、MC シミュレーションの 1 ステップは微小にしか変化せず、直近の状態と似たような状態をサンプリングせざるを得ない状況に陥るため、十分に多様な平衡状態を得るためには長時間の計算時間を必要となってしまう。これはダイナミクスとは関係なく平衡状態を直接サンプリングをが可能であるという MC シミュレーションの利点を大きく損なうものになる。

この問題を解決しようとする試みは、既にいくつかの先行研究が知られている。例を挙げると、フォースバイアスモンテカルロ法 (Force bias Monte Carlo, FBMC)[78] やスマートモンテカルロ法 (Brownian dynamic smart Monte Carlo, SMC) 法 [79] では、現状態のエネルギーの勾配方向に大きく確率をもつような提案分布を用いることによって、衝突確率を低減している。また、より大局的なエネルギー変化を取り扱う手法として、凝集体積バイアスモンテカルロ法 [80]、レプリカ交換法 [81]、ハイブリッドモンテカルロ

法 [82], アンブレラサンプリング [83] などがあるが, 本研究の範囲外である為詳細は割愛する.

7.3 CNMF モデルによる同変性を備えた分布変換

3章の深層生成モデルの節で述べた通り, CNF は, ある初期分布 $p(\mathbf{z}(t=0))$ に従うサンプル $\mathbf{z}(t=0)$ を, それぞれ1階微分方程式によって変換することによって変換後のサンプル $\mathbf{z}(t=T)$ を得るものであり, その際に変化後のサンプルが従う確率分布 $p(\mathbf{z}(t=T))$ を容易に計算できるという性質がある. CNF は次のように定式化される. 次の1階微分方程式

$$\frac{d}{dt}\mathbf{z} = f(\mathbf{z}, t) \quad (7.8)$$

を考える. ここで $f(\mathbf{z}, t)$ は \mathbf{z} と t に対してリプシッツ連続関数であるとする. このとき, サンプル $\mathbf{z}(t)$ が従う確率分布の変化は

$$\frac{d}{dt} \log p(\mathbf{z}(t)) = -\text{tr} \left(\frac{d}{d\mathbf{z}(t)} f \right) \quad (7.9)$$

に従う. ここで, 上式右辺の項 $\text{tr}(df/d\mathbf{z}(t))$ は変数 \mathbf{z} の瞬時変化 (instantaneous change) と呼ばれるている. $\text{tr}(\cdot)$ は行列の対角成分の和であるから, 微分方程式の変化を定める関数 f の \mathbf{z} による微分によって導かれる行列のうち, 確率の変化に貢献するのは行列の対角成分だけであるということが分かる.

CNF を MC シミュレーションに応用するために, 状態 \mathbf{z} として分子系内の全粒子座標 $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ を使用する. 状態 \mathbf{r} を1階微分方程式によって状態を変換させるとき, 変換は平行移動同変性・回転移動同変性・置換同変性の3つ同変性を満たす必要がある. ここで, これらの同変性の定義は次の通りである.

平行移動同変性

変換 $g(\cdot)$ が平行移動同変性を持つとは、任意の系の状態 \mathbf{r} と任意の平行移動ベクトル \mathbf{x} に対して、系の状態全体を \mathbf{x} だけ平行移動させた後に $g(\cdot)$ によって変換した状態と、系の状態を $g(\cdot)$ によって変換した状態全体を \mathbf{x} だけ平行移動させた状態が一致することである。すなわち、

$$g(\mathbf{r}) - \mathbf{x} = g(\mathbf{r} - \mathbf{x}) \quad (7.10)$$

が任意の \mathbf{r} , \mathbf{x} で成り立つことである。ただし、状態全体に対する平行移動 $\mathbf{r} - \mathbf{x}$ を $(\mathbf{r}_1 - \mathbf{x}, \mathbf{r}_2 - \mathbf{x}, \dots, \mathbf{r}_N - \mathbf{x})$ と定義する。

回転移動同変性

同様に、変換 $g(\cdot)$ が回転移動同変性を持つとは、任意の系の状態 \mathbf{x} と任意の回転行列 \mathbf{R} に対して、系の状態全体を \mathbf{R} だけ回転させた後に $g(\cdot)$ によって変換した状態と、系の状態を $g(\cdot)$ によって変換した状態全体を \mathbf{R} だけ回転移動させた状態が一致することである。すなわち、

$$\mathbf{R}g(\mathbf{r}) = g(\mathbf{R}\mathbf{r}) \quad (7.11)$$

が任意の \mathbf{r} , \mathbf{R} で成り立つことである。ただし、状態全体に対する回転移動 $\mathbf{R}\mathbf{r}$ を $(\mathbf{R}\mathbf{r}_1, \mathbf{R}\mathbf{r}_2, \dots, \mathbf{R}\mathbf{r}_N)$ と定義する。

置換同変性

また同様に、変換 $g(\cdot)$ が置換同変性を持つとは、任意の系の状態 \mathbf{x} と任意の置換 π に対して、系の状態の順序を置換 π によって交換した後に $g(\cdot)$ によって変換した状態と、系の状態を $g(\cdot)$ によって変換した状態の順序を置換 π によって交換した状態が一致することである。すなわち、

$$\pi g(\mathbf{r}) = g(\pi\mathbf{r}) \quad (7.12)$$

が任意の \mathbf{r} , π で成り立つことである。ただし、状態に対する置換とは、 $\pi\mathbf{r} = (\mathbf{r}_{\pi(1)}, \mathbf{r}_{\pi(2)}, \dots, \mathbf{r}_{\pi(N)})$ のことであり、置換列 $(\pi(1), \pi(2), \dots, \pi(N))$ は 1 から N までの整数を重複なく 1 つずつ選んだものである。

分子系の状態において、上記 3 つの同変性を満たす必要がある理由は、分子の状態 \mathbf{r}

が平行移動・回転・置換によって区別がつかないためである*2。本来区別のつかない状態が、変換によって異なる確率を持つてしまうことは直観的におかしいことであるし、点群を扱う機械学習の先行研究 [84] においてもこの同変性を満たさないモデルでは著しい性能低下がみられることが報告されているから、この性質は必要である。

これら 3 つの同変性を満たすために、本研究では Continuous Normalizing Molecular Flow (CNMF) を提案する。CNMF は、次の一階微分方程式

$$\frac{d}{dt}\mathbf{r}_i = \sum_{i \neq j} f_{ij}(\|\mathbf{r}_i - \mathbf{r}_j\|)(\mathbf{r}_i - \mathbf{r}_j) \quad (7.13)$$

によって状態を変換する。CNMF は原子間の 2 体間相互作用 f_{ij} によって状態変化するものであり、この相互作用は座標系や粒子番号に依存していないため、3 つの同変性を満たす。また、これは CNF の一種であるから、変換による確率分布の変化も計算することができる。上記の一階微分方程式に従う場合、状態 \mathbf{r} の瞬時変化は解析的に

$$\text{tr} \left(\frac{d}{d\mathbf{r}(t)} f \right) = \sum_{i \neq j} D f_{ij}(r_{ij}) + r_{ij} \frac{d}{dr_{ij}} f_{ij}(r_{ij}) \quad (7.14)$$

と展開できる。ここで、 D は空間の次元であり、また $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ である。したがって、

$$\frac{d}{dt} \log p(\mathbf{r}(t)) = \sum_{i \neq j} D f_{ij}(r_{ij}) + r_{ij} \frac{d}{dr_{ij}} f_{ij}(r_{ij}) \quad (7.15)$$

を数値積分することで $p(\mathbf{r}(t))$ を求めることができる。

ここで、CNMF はあらゆる分布を表現できる能力 (普遍性, Universality) を持っているわけではないことに注意する必要がある。普遍性を導入するためには、3 体間以上の相互作用や、グラフ畳み込み演算による相互作用などを導入することが考えられる。しかしながら、そのような複雑な相互作用を導入すると式 7.15 の瞬時変化の計算量が増加し、以降に述べるサンプリングアルゴリズムの効率性が低下するため、本研究では CNMF は 2 体間相互作用のみであるとしている。

*2 ここでいう置換とは、データの順序を変えるということであり、異種粒子からなる系でも成り立つ。

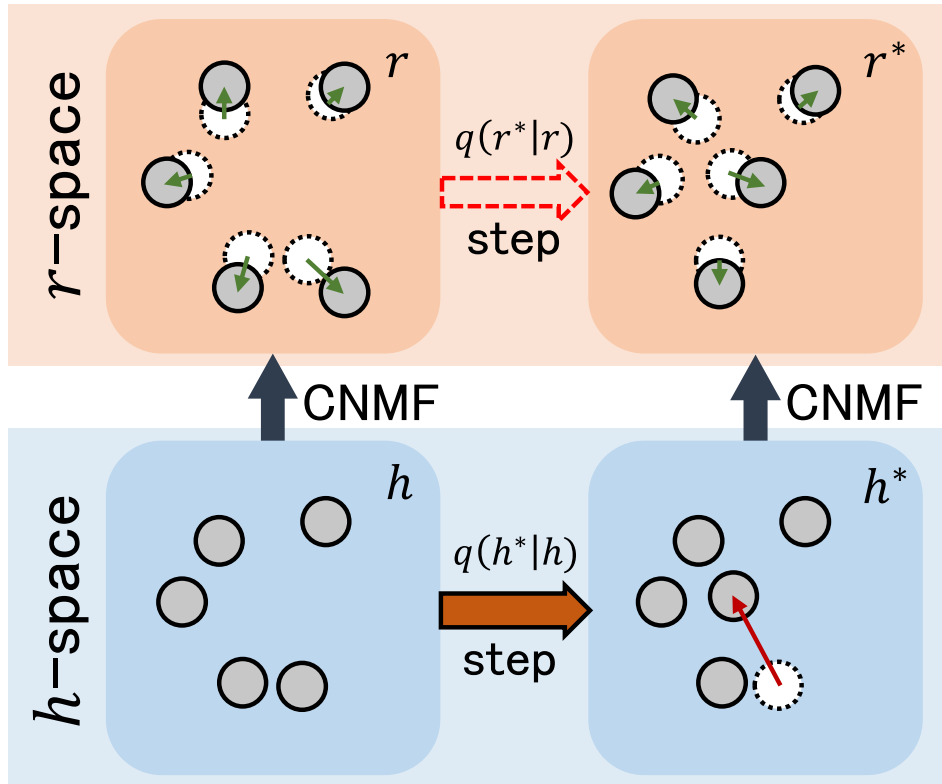


図 7.2 MFMH 法の模式図. 実空間 r で提案分布を用いる代わりに, 潜在空間 h で提案分布を用い, CNMF 法を用いて候補状態を変化させることで, 実空間上の提案分布から不都合な配置を取り除く.

7.4 MFMH 法による CNMF モデルに基づくサンプリングアルゴリズム

CNMF モデルを用いて提案分布を適切に変換することで, 効率的なサンプリングを実現したい. CNMF モデルに基づく具体的な変換方法は, f_{ij} によって決定される. f_{ij} の具体的な構成法は次節以降に譲るとして, この節では, f_{ij} が定まっているときに CNMF を用いて目的の分布をサンプリングするアルゴリズムである, MFMH 法を提案する.

MFMH 法では, MH 法で通常考える (実) 状態空間 r に加えて, これに対応する潜在状態空間 h も導入し, 潜在状態 h が CNMF モデルによって変換されたものが実状態 r になると考える. すなわち, 式 7.13 において時刻 0 の状態 $r(t=0)$ が潜在状態 h に相当し, そこから一定時間 T だけ積分した状態 $r(T)$ が実状態 r に相当する. これを模式的に表したものが図 7.2 である. 潜在状態空間を導入することで, 提案分布をより効率的な分

布へ変換することができる可能性がある．具体的には，MH 法において使用していた提案分布 $q(\mathbf{r}^*|\mathbf{r})$ を，MFMH 法では潜在空間上の提案分布 $q(\mathbf{h}^*|\mathbf{h})$ として用いる．潜在空間上の提案分布から得られた潜在空間上の候補状態のサンプル \mathbf{h}^* を，CNMF により実状態に変換することで，実空間上の候補状態 \mathbf{r}^* を得るということである．この変換によって，実状態空間上の提案分布に相当する分布 $q(\mathbf{r}^*|\mathbf{r})$ には，適切な関数 f_{ij} の選択によって原子同士が近接する高エネルギー配置が取り除かれていることを期待する．

CNMF は変換後の確率分布が計算できるため，潜在状態空間上の提案分布 $q(\mathbf{h}^*|\mathbf{h})$ が与えられたとき，実空間上の提案分布 $q(\mathbf{r}^*|\mathbf{r})$ は

$$q(\mathbf{r}^*|\mathbf{r}) = p(\mathbf{h}^* \rightarrow \mathbf{r}^*)q(\mathbf{h}^*|\mathbf{h}) \quad (7.16)$$

という関係にあることがわかる．ただし $p(\mathbf{h}^* \rightarrow \mathbf{r}^*)$ は式 7.15 の確率変化によって次のように計算される値である

$$p(\mathbf{h}^* \rightarrow \mathbf{r}^*) = \frac{p(\mathbf{r}(T) = \mathbf{r}^*)}{p(\mathbf{r}(0) = \mathbf{h}^*)}. \quad (7.17)$$

したがって，実空間上でサンプリングされた状態の確率の値が分かるから，採択率（式 7.6）を正しく計算できる．

まとめると，MFMH 法によるサンプリングアルゴリズムの全体は次のようになる．

1. 現在のステップ数を $i = 0$ として，適当な潜在状態の初期状態 $\mathbf{h}^{(0)}$ を設定する．
2. CNMF モデルに基づく一階微分方程式について，現在の潜在状態を初期値として時間 T まで数値積分を行い，時刻 T における値を現在の実状態 $\mathbf{r}^{(0)}$ とする（ただし T はアルゴリズム全体を通して一定とする）．
3. 以下のステップを十分なサンプルが集まるまで繰り返す．
4. 現在の潜在状態 $\mathbf{h}^{(i)}$ をもとにして，潜在空間上の提案分布 $q(\mathbf{h}^*|\mathbf{h})$ に従う潜在空間上の候補状態 \mathbf{h}^* をサンプリングする．
5. CNMF モデルに基づく一階微分方程式について，潜在空間上の候補状態 \mathbf{h}^* を初期値として時間 T まで数値積分を行い，時刻 T における値を候補状態 \mathbf{r}^* とする．
6. 候補状態 \mathbf{r}^* を採択率 $A(\mathbf{r}^*, \mathbf{r}^{(i)})$ の確率で採択し，候補状態 \mathbf{r}^* を次の時刻の状態 $\mathbf{r}^{(i)}$ とする．同時に対応する潜在状態も更新する．採択されない場合は，現在の状態 $\mathbf{r}^{(i)}$ を次の時刻の状態 $\mathbf{r}^{(i+1)}$ とする．
7. 現在のステップ数を $i = i + 1$ と更新する．

7.5 ISF 法による等圧縮かつ近接配置のない提案分布の実現

この節では、CNMF モデルにおける変換を決定づける関数 f_{ij} の具体的な構成法のひとつとして、ISF 法を提案する。ISF 法は、元の提案分布から原子が近接するような配置を取り除く一方、それ以外の配置では確率分布が不変（あるいは一様圧縮）するような分布を解析的に構成するものである。

まず、確率分布が不変となるような CNMF モデルの条件について考える。確率分布が不変ということは、状態の瞬時変化

$$\text{tr} \left(\frac{d}{d\mathbf{r}(t)} f \right) = \sum_{i \neq j} D f_{ij}(r_{ij}) + r_{ij} \frac{d}{dr_{ij}} f_{ij}(r_{ij}) \quad (7.18)$$

がゼロになるということである。これをゼロにするための十分条件としては

$$0 = D f_{ij}(r_{ij}) + r_{ij} \frac{d}{dr_{ij}} f_{ij}(r_{ij}) \quad (7.19)$$

を満たすように f_{ij} を選べばよい。ここで、これは個別の相互作用について瞬時変化が 0 となるようにしているが、これは必要条件ではなく、個別では相互作用の瞬時変化が 0 にならずとも、総合的に瞬時変化がゼロになるように f_{ij} を構成することは可能だろう。しかし、解くべき問題の自由度が大きくなってしまったため、個別の相互作用の瞬時変化がすべてゼロになるように設定している。上式は微分方程式であるから、これを解くと

$$f_{ij} = 1/r^D \quad (7.20)$$

を得る（定数倍の違いを含む）。このように求めた関数 f_{ij} を D 次元上の Inverse square flow (ISF) と名づける。この名前の理由としては、 r/r^D が逆二乗則の形式と一致するからである。

さて、この関数 $f_{ij} = 1/r^D$ がどのような性質を持っているか確認するために、簡易的な実験を行う。今、系を 2 次元空間として、図 7.3(a) のように正方形のそれぞれの頂点と正方形の中心に合計 5 つの粒子がおかれている状況を考える。ここで、中心の粒子が自身の位置を中心とした一様分布で移動しようとしている。この一様分布が図 7.3(b) で表されている。この一様分布には、中央の粒子が移動する先に正方形の頂点にある粒子と接近してしまう配置が含まれている。よって、これが MH 法のサンプリング過程であれば、そのような配置は一切採択されることはなく、無駄になる。この分布に ISF 関数による CNMF モデルを適用して変化させると、分布は図 7.3(c), 図 7.3(d) と変化していく。た

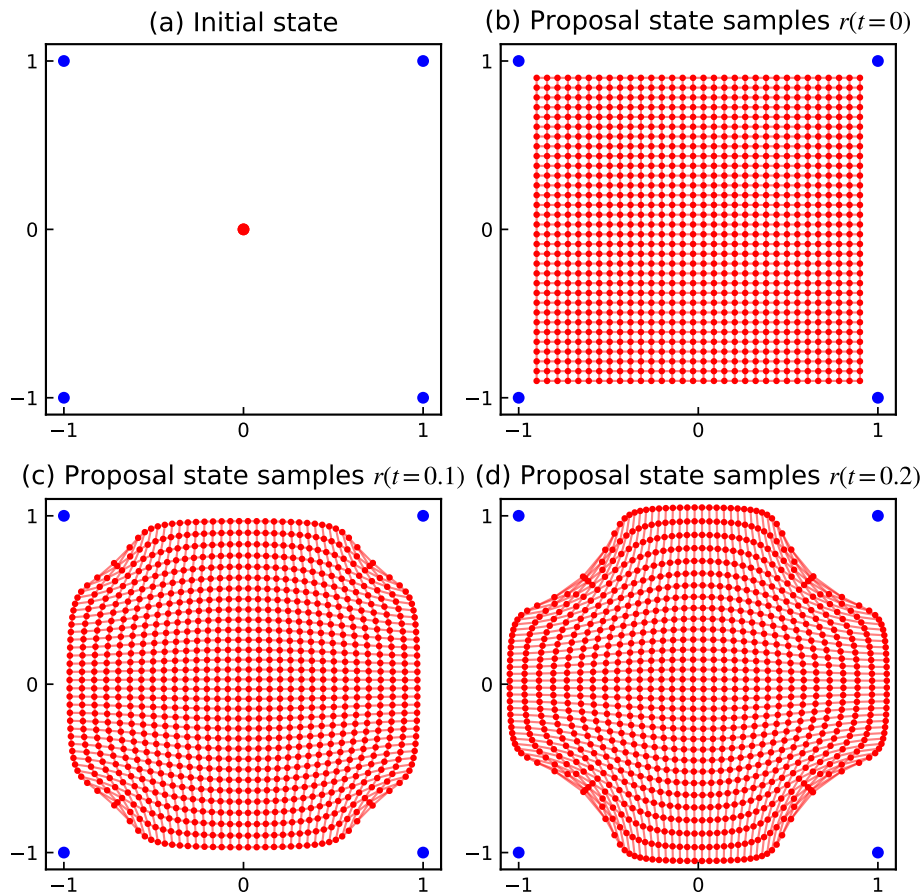


図 7.3 ISF 法による確率分布の変換の様子。(a)5つの粒子の配置を示している。(b)真ん中の赤い点が移動するとして、移動先を赤い点で示している。移動先は青い点で示す粒子の近接配置を含む。(c)ISF 法によって変換された点 ($T = 0.1$)。近接配置が少し取り除かれている。ただし、青い点は便宜的に固定している。(d)ISF 法によってさらに変換された点 ($T = 0.2$)。近接配置がさらに取り除かれている一方、確率分布が一定にならなければならないので縦横に押し出されている様子が分かる。

だし、視覚的なわかりやすさのために、正方形の頂点の粒子は固定している（本来は全粒子が移動する）。図 7.3(b-d) の分布変化を観察すると、ISF には利点となる 2 つの性質があることが分かる。1 つ目は、提案分布は変形しているものの体積変化がゼロ、すなわち確率分布が一様であるままになっていることである。これは ISF の構成方法から明らかではあるが、変形後も元の確率の値が変わらないため、採択率の計算に影響を与えないという利点である。2 つ目は、変形後の確率分布は、粒子が近接する配置が取り除かれているということである。この性質により、無駄となる近接原子配置を候補状態として試すことがなくなり、効率の向上につながると考えられる。

ただし、ISF 法が周期境界条件のある系に適用される際には注意すべきことがある。周期境界条件下では、系の体積は一定であるので、ISF の性質として提案分布の変化がゼロであるということと、粒子が近接する配置が取り除かれるという性質の両方を満たすことはできないはずである。なぜなら、取り除かれた分の確率体積がどこかに現れている必要があるからである。この問題は、周期境界条件下における ISF 法を用いた場合の状態の瞬時変化を理論的に解析することで、実際の挙動を確かめることができる。以降の解析は系の次元 D を 3 に限るとする。

周期境界条件においては、系には系内だけではなく系の周期セルにある粒子の影響を受けることを踏まえて、CNMF は次の一階微分方程式

$$\frac{d}{dt}\mathbf{r}_i = \sum_{\mathbf{n} \in \mathbb{Z}^3} \sum_{i \neq j} f_{ij}(\|\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}\|)(\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}) \quad (7.21)$$

となる。ただし、ここでは簡単のためにセルサイズは単位長さの立方体であるとしており、 $\mathbf{n} (\in \mathbb{Z}^3)$ は周期セルからの影響を表す格子ベクトルである。今、 f_{ij} は ISF 法に基づくから、

$$\frac{d}{dt}\mathbf{r}_i = \sum_{\mathbf{n} \in \mathbb{Z}^3} \sum_{i \neq j} \|\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}\|^{-3}(\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}) \quad (7.22)$$

と展開できる。これは、分子シミュレーションにおいてクーロン力の総和の形式をしており、単に \mathbf{n} について総和をとるだけでは収束しないことが知られている。これを収束させるために、分子シミュレーションでは Ewald の方法 (Ewald summation)[85] として知られる方法を用いている。

Ewald の方法では、次のクーロンポテンシャルの周期和

$$h^p(\mathbf{r}) = \sum_{\mathbf{n} \in \mathbb{Z}^3} \|\mathbf{r} - \mathbf{n}\|^{-1} \quad (7.23)$$

について、まず次のように展開する。

$$h^p(\mathbf{r}) = \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{1}{\|\mathbf{r} - \mathbf{n}\|} \quad (7.24)$$

$$= \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{\operatorname{erfc}(G\|\mathbf{r} - \mathbf{n}\|)}{\|\mathbf{r} - \mathbf{n}\|} + \frac{\operatorname{erf}(G\|\mathbf{r} - \mathbf{n}\|)}{\|\mathbf{r} - \mathbf{n}\|}. \quad (7.25)$$

ここで $\operatorname{erf}(\cdot)$ および $\operatorname{erfc}(\cdot)$ はそれぞれガウスの誤差関数とガウスの相補誤差関数であり、 G は 0 より大きい任意の定数である。さらに、総和内の右辺を 3 次元フーリエ級数展開す

ることで

$$= \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{\operatorname{erfc}(G\|\mathbf{r} - \mathbf{n}\|)}{\|\mathbf{r} - \mathbf{n}\|} + 4\pi \sum_{\mathbf{k} \neq \mathbf{0} \in \mathbb{Z}^3} \frac{e^{-\|\mathbf{k}\|^2/4G^2} e^{-i\mathbf{k} \cdot \mathbf{r}}}{\|\mathbf{k}\|} \quad (7.26)$$

とする。すなわち，Ewaldの方法では，近距離力と遠距離力を分離し，近距離力を実空間，遠距離力を端数空間で総和を取る。このように展開することで，クーロン力の総和を収束させることができる。

さて，もともと考えていた周期境界条件における ISF 法に基づく CNMF モデル（式 7.20）は，クーロンポテンシャルの周期和の勾配が

$$\nabla_{\mathbf{r}} h^p(\mathbf{r}) = \nabla_{\mathbf{r}} \sum_{\mathbf{n} \in \mathbb{Z}^3} \|\mathbf{r} - \mathbf{n}\|^{-1} \quad (7.27)$$

$$= - \sum_{\mathbf{n} \in \mathbb{Z}^3} \|\mathbf{r} - \mathbf{n}\|^{-3} (\mathbf{r} - \mathbf{n}) \quad (7.28)$$

と計算できることから，クーロンポテンシャルの周期和 $h^p(\mathbf{r})$ を用いて

$$\frac{d\mathbf{r}_i}{dt} = - \nabla_{\mathbf{r}} \sum_{i \neq j} h^p(\mathbf{r}_i - \mathbf{r}_j) \quad (7.29)$$

と表現できる。これに従う状態の瞬時変化は h^p を用いて

$$\operatorname{tr} \left(\frac{d}{d\mathbf{r}(t)} f \right) = \sum_{i \neq j} \frac{2}{r_{ij}} \frac{d}{dr_{ij}} h_{ij} + \frac{d^2}{dr_{ij}^2} h_{ij} \quad (7.30)$$

と計算できる。

ここで， h^p の Ewald の方法による展開について，端数空間の総和の項が無かったと仮定すると

$$h^p(\mathbf{r}) = \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{\operatorname{erfc}(G\|\mathbf{r} - \mathbf{n}\|)}{\|\mathbf{r} - \mathbf{n}\|} \quad (7.31)$$

となるから，これを状態の瞬時変化の式（式 7.30）に代入することで

$$\operatorname{tr} \left(\frac{d}{d\mathbf{r}(t)} f \right) = - \sum_{i \neq j} \frac{4G^3}{\sqrt{\pi}} \sum_{\mathbf{n} \in \mathbb{Z}^3} e^{-G^2\|\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}\|^2} \quad (7.32)$$

と計算できる。実際， $G \rightarrow 0$ の極限では，ガウスの誤差関数の性質から，端数空間の総

和の項はゼロに収束する．よって，状態の瞬時変化は厳密に

$$\mathrm{tr} \left(\frac{d}{d\mathbf{r}(t)} f \right) = - \sum_{i \neq j} \lim_{G \rightarrow 0} \frac{4G^3}{\sqrt{\pi}} \sum_{\mathbf{n} \in \mathbb{Z}^3} e^{-G^2 \|\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}\|^2} \quad (7.33)$$

$$= - \sum_{i \neq j} \frac{4}{\sqrt{\pi}} \int_{\mathbf{x} \in \mathbb{R}^3} e^{-\|\mathbf{x}\|^2} \quad (7.34)$$

$$= - \sum_{i \neq j} \frac{4}{\sqrt{\pi}} \pi^{3/2} = - \sum_{i \neq j} 4\pi = -4\pi N(N-1) \quad (7.35)$$

と計算できる．ここで，式変形の過程でガウス積分を実行している．従って，瞬時変化は粒子数にのみに依存する定数であることが分かった．

以上の計算から，自由境界条件のときに瞬時変化がゼロ（確率不変）となる ISF 法は，周期境界条件の場合では瞬時変化が定数，すなわち確率が等率圧縮することが示されたことになる．周期境界条件の場合であっても，粒子が近接する配置が取り除かれるという性質は変わらないから，CNMF を周期境界条件において適用する場合は，確率不変ではなく粒子数に依存した等率圧縮であることにさえ気を付ければよいことになる．これは，粒子数が変動しないカノニカルアンサンブルシミュレーションでは周期境界条件と自由境界条件を区別する必要がないことを示している．なぜなら，採択率の計算式

$$A(\mathbf{r}^* | \mathbf{r}_i) = \max \left(\frac{p(\mathbf{r}^*) q(\mathbf{r}^i | \mathbf{r}^*)}{p(\mathbf{r}^i) q(\mathbf{r}^* | \mathbf{r}^i)}, 1 \right) \quad (7.36)$$

について，式 7.16 を代入すると

$$A(\mathbf{r}^* | \mathbf{r}_i) = \max \left(\frac{p(\mathbf{r}^*) p(\mathbf{h}^i \rightarrow \mathbf{r}^i) q(\mathbf{h}^i | \mathbf{h}^*)}{p(\mathbf{r}^i) p(\mathbf{h}^* \rightarrow \mathbf{r}^*) q(\mathbf{h}^* | \mathbf{h}^i)}, 1 \right) \quad (7.37)$$

であり，さらに等率圧縮の式（式 7.35）を代入すると

$$A(\mathbf{r}^* | \mathbf{r}_i) = \max \left(\frac{p(\mathbf{r}^*) e^{4\pi N(N-1)T} q(\mathbf{h}^i | \mathbf{h}^*)}{p(\mathbf{r}^i) e^{4\pi N(N-1)T} q(\mathbf{h}^* | \mathbf{h}^i)}, 1 \right) \quad (7.38)$$

$$= \max \left(\frac{p(\mathbf{r}^*) q(\mathbf{h}^i | \mathbf{h}^*)}{p(\mathbf{r}^i) q(\mathbf{h}^* | \mathbf{h}^i)}, 1 \right) \quad (7.39)$$

となり，CNMF による変換の影響が打ち消されるからである．ここで N, T はそれぞれ系に含む粒子数と CNMF の数値積分時間（一定）である．

一方，粒子数が変動するグランドカノニカルアンサンブルシミュレーションでは，採択

率が

$$A(\mathbf{r}^*|\mathbf{r}_i) = \max \left(\frac{p(\mathbf{r}^*)}{p(\mathbf{r}^i)} \frac{e^{4\pi N^i(N^i-1)T} q(\mathbf{h}^i|\mathbf{h}^*)}{e^{4\pi N^*(N^*-1)T} q(\mathbf{h}^*|\mathbf{h}^i)}, 1 \right) \quad (7.40)$$

$$(7.41)$$

となって一般に打ち消されるとはいえない。ここで N^i , N^* はそれぞれ時刻 i における状態の粒子数, 候補状態の粒子数である。グランドカノニカルアンサンブルシミュレーションにおいても変換の影響が打ち消されるようにする方法として, 粒子数 N に応じて CNMF の数値積分時間 T の方を変更することも考えられる。すなわち, ある基準積分長 T_0 および基準粒子数 N_0 に対して, CNMF の数値積分時間 T_N を

$$T_N = T_0 \frac{N_0(N_0 - 1)}{N(N - 1)} \quad (7.42)$$

と設定すれば,

$$A(\mathbf{r}^*|\mathbf{r}_i) = \max \left(\frac{p(\mathbf{r}^*)}{p(\mathbf{r}^i)} \frac{e^{4\pi N^i(N^i-1)T_N^i} q(\mathbf{h}^i|\mathbf{h}^*)}{e^{4\pi N^*(N^*-1)T_N^*} q(\mathbf{h}^*|\mathbf{h}^i)}, 1 \right) \quad (7.43)$$

$$= \max \left(\frac{p(\mathbf{r}^*)}{p(\mathbf{r}^i)} \frac{q(\mathbf{h}^i|\mathbf{h}^*)}{q(\mathbf{h}^*|\mathbf{h}^i)}, 1 \right) \quad (7.44)$$

となって粒子数の影響が打ち消される。ここで, T_N^i , T_N^* は, それぞれ時刻 i における状態の粒子数に対応する数値積分時間, 候補状態の粒子数に対応する数値積分時間である。グランドカノニカルアンサンブルシミュレーションにおいてはどちらの方法でも正しいシミュレーションができることには変わりはないが, 本研究では後者を採用することにする。

7.6 SL 法による系ごとに最適な提案分布の実現

この節では, CNMF モデルにおける変換を決定づける関数 f_{ij} の具体的な構成法のひとつである SL(Self-Learning) 法について述べる。ISF 法では, CNMF が近接配置を取り除きながらも他の状態の確率が不変となるように解析的に f_{ij} を構成したが, より効率向上を達成できる適切な f_{ij} が存在する可能性がある。そこで, SL 法では, 機械学習によって, 最適な f_{ij} を見つけることにする。

ところで, 最適な f_{ij} を見つけるためには, 何らかの最適性指標が必要となる。最適性指標は, 本研究の目的である「効率的に独立なサンプリングを実行できること」に関連付

いたものである必要があるが、サンプルの独立性を直接的に最適化の指標にしようとする
と、そのような指標自体の計算が難しくなってしまう、その結果最適化が困難になるとい
う問題が生じる。そこで、より計算が容易な指標を設計する。

まず、採択率 $A(\mathbf{r}^*|\mathbf{r}^i)$ について考えると、提案分布が変わらないのであれば、より状
態が更新されやすいという観点から、採択率は高い方が良い。採択率が 1、すなわち必ず
採択されるためには、採択率の式から、

$$1 = \frac{p(\mathbf{r}^*) q(\mathbf{r}^i|\mathbf{r}^*)}{p(\mathbf{r}^i) q(\mathbf{r}^*|\mathbf{r}^i)} \quad (7.45)$$

が成り立てばよい。上式を満たす十分条件として、

$$q(\mathbf{r}^*|\mathbf{r}^i) = p(\mathbf{r}^*) \quad (7.46)$$

が考えられる。これは現在の状態に全く依存せず候補状態がボルツマン分布に従うとい
うことである。すなわち、提案分布が目標の分布に一致していれば、採択率は最大になる。
このように選んだ提案分布が高速に独立なサンプリングを実行可能である場合、目的の分
布に対して効率的に独立なサンプリングを実行できることも明らかだろう。なぜなら、提
案分布が目的の分布そのものであって、提案分布は独立にサンプリングできるのであるか
ら、「効率的に独立なサンプリングを実行できること」を達成できているからである。

しかしながら、目的の分布そのものは高速に独立なサンプリングができないため、
MH 法により提案分布を代理で用いていたのであるから、これは不可能である。そこで、
MFMH 法による提案分布を、なるべくボルツマン分布に一致させることを考える。ここ
で、MFMH 法では潜在空間上の提案分布を CNMF モデルによって変換させるから、潜
在空間上の提案分布と CNMF モデルの関数 f_{ij} の両方が変更可能である。潜在空間上の
提案分布として要求される性質を考えると、(1) (はじめから) 効率的にサンプリングでき
るものである必要があることと、(2) CNMF の変換によってボルツマン分布に変換するわ
けであるから、いくら CNMF が確率分布を変形してくれるとはいっても、系全体に大局
的に広がるボルツマン分布を構成するためには、潜在空間上の提案分布も系全体をサン
プリングするものでなければならないはずである。このような 2 つの性質を満たす分布のう
ちのひとつとして、本研究では、系全体に広がる一様分布 $p(\mathbf{h}) = 1$ が挙げられる*³。そ
こで、SL 法では系全体に広がる一様分布 $p(\mathbf{h})$ を潜在空間の分布として、実空間の分布を
なるべくボルツマン分布に近づけるような確率分布変換方法 f_{ij} を獲得することにする。

*³ もちろん、より適切な潜在空間上の提案分布の設定がある可能性がある。

ここで注意点として、CNMF モデルは一様分布 $p(\mathbf{h})$ をボルツマン分布に厳密に近づけることができるほど強力な変形が可能なモデルではないので、実際に潜在空間上の提案分布を系全体の一様分布にしてサンプリングが効率的にできるということを主張しているわけではない。あくまで学習の評価基準として、なるべく望ましい変換に近づけるためのプロセスを行っているのであって、実際にサンプリング時には、(MH 法がそうであるのと同様の理由で) 潜在空間上の提案分布は局所的な移動を行う提案分布を選ぶことが適切であると考えられる。将来の展望として、一様分布をボルツマン分布に近似精度よく変換できる方法が現れた場合には、潜在空間上の提案分布として大局的な一様分布を選ぶことが可能になるかもしれない。

以上の議論より最適化指標として、一様分布 $p(\mathbf{h})$ をなるべくボルツマン分布に近い確率分布を近づけるような変換方法 f_{ij} を獲得するために、潜在空間上の一様分布を CNMF によって変換した分布とボルツマン分布の間のクロスエントロピー誤差関数

$$L(\boldsymbol{\theta}) = - \int p_B(\mathbf{r}^*) p(\mathbf{h} \rightarrow \mathbf{r}; \boldsymbol{\theta}) p(\mathbf{h}) d\mathbf{h} \quad (7.47)$$

を用いる。ここで、 $p(\mathbf{h})$ は潜在空間上の系全体に渡る一様分布を提案分布としたものであり、 $p(\mathbf{h} \rightarrow \mathbf{r}; \boldsymbol{\theta}) p(\mathbf{h})$ は潜在空間上の提案分布を CNMF により実空間へ変換した際の分布となっている。ただし、実空間へ変換方法を決定する f_{ij} にはモデル化のためのパラメータ $\boldsymbol{\theta}$ が存在していると考えているため、変換にパラメータ $\boldsymbol{\theta}$ を含んでいる。クロスエントロピー誤差関数は、2つの分布が近いほど小さくなる関数であり、この損失関数 $L(\boldsymbol{\theta})$ を最小化するパラメータ $\boldsymbol{\theta}$ を求めることで、潜在空間上の提案分布を実空間へ変換した時の分布がボルツマン分布になるような変換方法 f_{ij} を獲得することができる。上式をさらに展開すると、潜在空間上の提案分布 $p(\mathbf{h})$ は系全体に渡る一様分布を考えているから、 $p(\mathbf{h}) \propto 1$ であり、定数倍の違いを無視すれば

$$L(\boldsymbol{\theta}) = - \int p_B(\mathbf{r}^*) p(\mathbf{h} \rightarrow \mathbf{r}; \boldsymbol{\theta}) d\mathbf{h} \quad (7.48)$$

とできる。また、上式中の積分はボルツマン分布に対する期待値

$$L(\boldsymbol{\theta}) = - \langle p(\mathbf{h} \rightarrow \mathbf{r}; \boldsymbol{\theta}) \rangle_{p_B(\mathbf{r}^*)} \quad (7.49)$$

とみなすことができるから、損失関数をボルツマン分布に従うサンプル平均

$$L(\boldsymbol{\theta}) \simeq \sum_{\mathbf{r} \sim p_B(\mathbf{r}^*)} p(\mathbf{h} \rightarrow \mathbf{r}; \boldsymbol{\theta}) \quad (7.50)$$

によって近似できる．ここで，このサンプル平均は実状態のサンプル \mathbf{r} が与えられているが，総和内は潜在空間上の提案分布を実空間へ変換した際の確率の変化率となっているから，実状態のサンプル \mathbf{r} に対応する潜在状態 \mathbf{h} を知る必要がある*4．よって，式 7.13 を逆向きに積分して，実状態のサンプル \mathbf{r} からそれに対応する潜在状態 \mathbf{h} を求め，同時に式 7.15 も逆向きに積分することで，確率変化率を求めることにする．

また，ボルツマン分布に従うサンプル平均をとるということは，ボルツマン分布に従うサンプルが先に得られている必要がある．そこで SL 法の学習の際には，最適化されていない状態の f_{ij} を用いて MFMH 法によるボルツマン分布からのサンプリングを行いながら，同時に並行して，得られたサンプルを用いて損失関数 $L(\boldsymbol{\theta})$ を最小化を進める， f_{ij} の最適化を行う．これが SL 法の名前の自己学習 (Self-Learning) の理由である．

SL 法が最適化する関数 f_{ij} は，(1) 計算量が大きくなるように，短距離相互作用となっていること，(2) CNF の要件から， f_{ij} はその微分が連続であること，の 2 つの条件を満たしている必要がある．本研究の実験においては，上記の条件を満たすような関数 f_{ij} のモデルとして

$$f_{ij}(r_{ij}, \boldsymbol{\theta}) = \begin{cases} \sum_{k=2}^{64} \theta_k (1 - r_{ij}/r_c)^k & (r_{ij} < r_c) \\ 0 & (r_{ij} \leq r_c) \end{cases} \quad (7.51)$$

を用いることにする．ただし，このモデルでは 64 次までの多項式を用いているが，64 次以上の多項式があっても獲得されるモデルにほとんど違いがないことを予備実験で確認している．ここで $\boldsymbol{\theta} = (\theta_2, \theta_3, \dots, \theta_{64})$ がこのモデルのパラメータであり， r_c は相互作用の影響範囲の最大値として， $r_c = 2$ と設定している．

7.7 LJ 粒子系を用いた MC シミュレーションの効率評価

MFMH 法の効率の評価のため，カノニカルアンサンブルシミュレーションおよびグランドカノニカルアンサンブルシミュレーションの双方において比較検証を行う．

7.7.1 カノニカルアンサンブルシミュレーションによる実験

MFMH 法の効率の評価のため，カノニカルアンサンブル下の LJ 粒子系を用いてシミュレーションを行った．まず，シミュレーションの対象となる系の設定について述べ

*4 CNMF の通常のプロセスでは，潜在状態 \mathbf{h} が先に合って，そこから数値積分を通して実状態 \mathbf{r} とその際の確率変化率を求めている．

る。系は1種類のLJ粒子256個からなり、その2体間相互作用は

$$U(r) = 4(r^{-12} - r^{-6}) \quad (7.52)$$

である。ただし、距離 r が3以上の場合は $U(r) = 0$ とするカットオフ半径を導入している。ただし r は粒子同士の無次元化距離である。また、系はカノニカルアンサンブルに従い、対応するボルツマン分布は

$$p_B(\mathbf{r}) \propto \exp \left[-\beta \sum_{i \neq j} U(r_{ij}) \right] \quad (7.53)$$

となる。ここで β は系の逆温度であり、 $\beta = 1/4$ と設定している。系には周期境界条件が設定されており、周期セルの大きさは一辺 $2^{8/3}$ の立方体となっている*⁵。

以上の系に対して、MH法、ISF法に基づくMFMH法、SL法に基づくMFMH法のそれぞれを用いて 10^6 ステップのシミュレーションを行った。シミュレーションの効率評価のために、次の2つの評価基準を導入している。1つ目は平均採択率 R_A であり、生成した候補状態の数 ($= 10^6$) に対して候補状態が採択された数の割合を平均採択率と定義する。平均採択率 R_A はその値が高いほど生成した候補状態が無駄にならずに済むという点を評価している。ただし、平均採択率が高い場合でも、系の状態がほとんど変化していない場合は独立なサンプリングができていないとは言えない。そこで2つ目として、効率性を直接評価する基準、粒子の平均二乗変位の比例定数 $\langle C(t)/t \rangle$ を利用する。ここで、粒子の平均二乗変位 $C(t)$ を

$$C(t) = \langle \|\mathbf{r}(t+\tau)_i - \mathbf{r}(t)_i\|^2 \rangle_{i,\tau} \quad (7.54)$$

と定義する。すなわち、粒子がステップ数 t のあいだに移動した距離の二乗の平均を、 $C(t)$ としている。一定のステップで移動できた量が大きければ大きいほど、状態の変化量が大きいわけであるから、同じステップ数 t において $C(t)$ が大きい方が、より独立なサンプルを取得する効率が高いと考えられる。よって、粒子の運動がブラウン運動に近い運動となるときは、平均二乗変位はステップ数に比例することも踏まえて、この比例定数 $\langle C(t)/t \rangle$ を効率性の評価基準として用いることにする。

次に、各手法の設定について述べる。MH法については、用いる提案分布として、一様ランダムに選ばれた粒子をそれを中心とした標準偏差 σ の無相関正規分布に従うように動かすことにする。標準偏差 σ の大きさについては、後の実験結果のとおり、小さいと採

*⁵ 密度が丁度1になるように設定している。

採択率が上昇する代わりに候補状態の移動量が低下し、大きいと候補状態の移動量は大きくなるが採択率が低下するというトレードオフの関係にある。よって、いくつかの σ について同様のシミュレーションを実行することにする。ISF 法および SL 法についても、提案分布は比較のため MH 法で用いた提案分布と全く同じものを潜在空間上で用いることにする。

ISF 法については、CNMF の積分時間によって排除される近接配置領域の大きさが異なる。積分時間が小さすぎると、排除領域が小さくなり不必要な高エネルギー領域が残ってしまうが、逆に大きすぎると、サンプルとして実現されるべき配置も取り除くことになってしまう。よって、適切な積分時間を求める必要がある。正しい計算が行える積分時間の上限は、次のような簡単な計算によって見積もることができる。潜在空間上に粒子がただ 2 つだけ存在する場合を考えると、一般性を失わずにその座標を $(-x_0/2, 0, 0)$ と $(x_0/2, 0, 0)$ とできる。この配置を初期状態にとって、ISF 法の積分を積分時間 T として変形させると、変形後の座標は

$$x_T = \sqrt[3]{6T + x_0^3} \quad (7.55)$$

を用いて $(-x_T/2, 0, 0), (x_T/2, 0, 0)$ となる。つまり、潜在空間において x_0 の距離だけ離れた粒子は実空間では x_T の距離になる。これは、実空間で $\sqrt[3]{6T}$ 以下の粒子ペアは存在できなくなることを意味するから、シミュレーション実行時にとり得る最小の原子間距離が $\sqrt[3]{6T}$ 以上となるように T を選ばばよい。この見積りを今回の系の実験結果に照らし合わせると、本実験においてはおよそ $T = 0.1$ を得た。よって CNMF の積分時間は $T = 0.1$ を用いることにする。

SL 法については、学習によって f_{ij} を求める必要があるため、予備実験として 10^5 ステップの計算を行い、式 7.51 のパラメータを学習しておく。予備実験の提案分布の移動半径は $\sigma = 0.2$ を採用した。この値は MH 法において適正な値であると知られているものである [37]。SL 法における CNMF の積分時間は 1 に固定する。これは、CNMF の積分時間が C 倍になるのは f_{ij} を C 倍するのと同様であるから、積分時間の自由度を f_{ij} に押し付けても一般性を失わないためである。予備実験の結果を図 7.4 に示す。図 7.4(b) はステップ数に対する平均採択率 R_A の変化を示しており、学習開始から 5 万ステップほどで採択率は 35 % に達し、MH 法によるベースラインの平均採択率 11 % の約 3 倍に到達していることが分かる。図 7.4(c) はステップ数に対する平均二乗変位の比例定数の変化を示している。こちらも平均採択率同様、5 万ステップほどで最大値に達し、ベースラインと比べて約 6 倍もの値を達成していることが分かる。また、5 万ステップ時点でのシミュレーション結果として、エネルギーヒストグラムを図 7.4(a) に示しており、良好な一

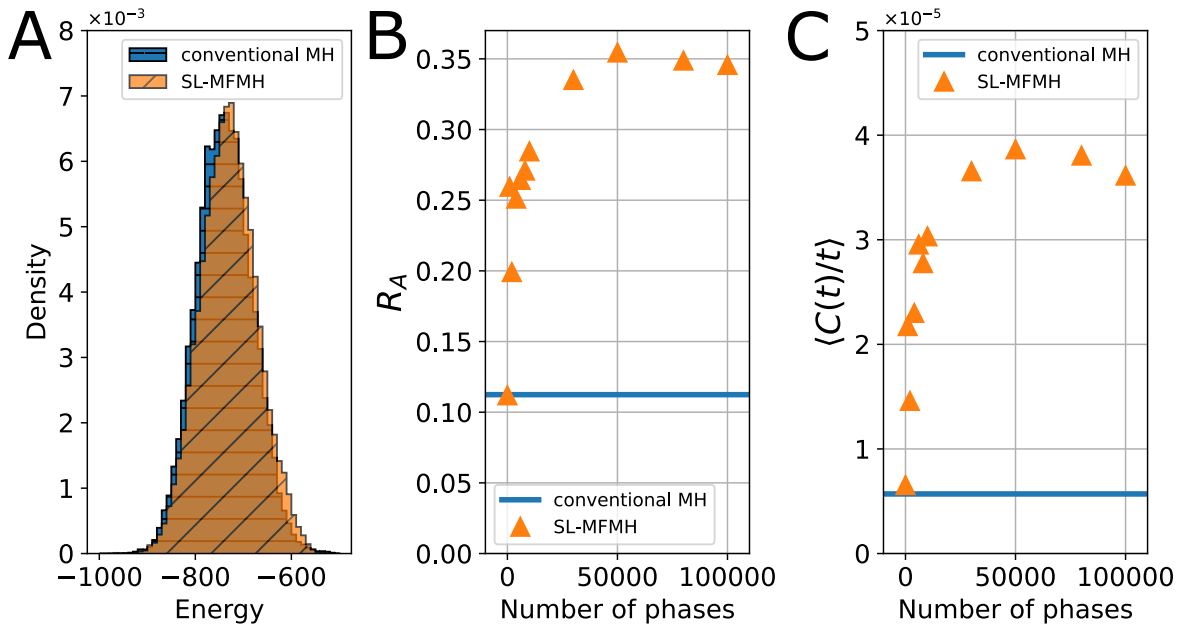


図 7.4 SL 法の学習過程 (予備実験). (a) 5×10^4 ステップ目における SL 法によってサンプリングされた状態のエネルギーヒストグラムと、MH 法によるエネルギーヒストグラムの比較. (b) 学習に伴う採択率 R_A の変化の様子. (c) 学習に伴う平均二乗変位の比例定数 $\langle C(t)/t \rangle$ の変化の様子.

致を示していることが分かる. よって, 5 万ステップ経過時点におけるパラメータを f_{ij} を最適値であると判断し, 以降の本実験ではこれを用いることにする.

さて, 本実験の結果として MH 法, ISF 法, SL 法の計算結果を図 7.5 に示す. 図 7.5 は各手法および移動半径 σ の変化に伴い, 採択率 R_A , および平均二乗変位の比例定数 $\langle C(t)/t \rangle$ がどのように変化するかをプロットしたものである. 図 7.5(a) は $R_A \langle C(t)/t \rangle$, (b) は $\sigma - R_A$, (c) は $\sigma - \langle C(t)/t \rangle$ を示している. 図 7.5(b), 図 7.5(c) より, 採択率と平均二乗変位の比例定数どちらをとっても, 全ての移動半径の領域 σ について, 高い順に SL 法, ISF 法, MH 法, となっている. 特に, 平均二乗変位の比例定数の最大値は, SL 法が MH 法の約 20 倍にも達していることが分かる. このことから, 学習によって提案分布を最適化する SL 法が, 実際に優れた効率を達成できていることが分かる. ただし, ISF 法についても, 系に特別に最適化されていないにもかかわらず SL 法に近い採択率・比例定数を達成できている点では, 汎用性の高さが伺える.

採択率と平均二乗変位を比較する図 7.5(a) では, SL 法において興味深い挙動が確認される. 移動半径 σ が大きい領域において, 通常の MH 法では採択率の著しい低下が起きるため, 平均二乗変位についても大きく低下している. 一方, SL 法においては, 採択率

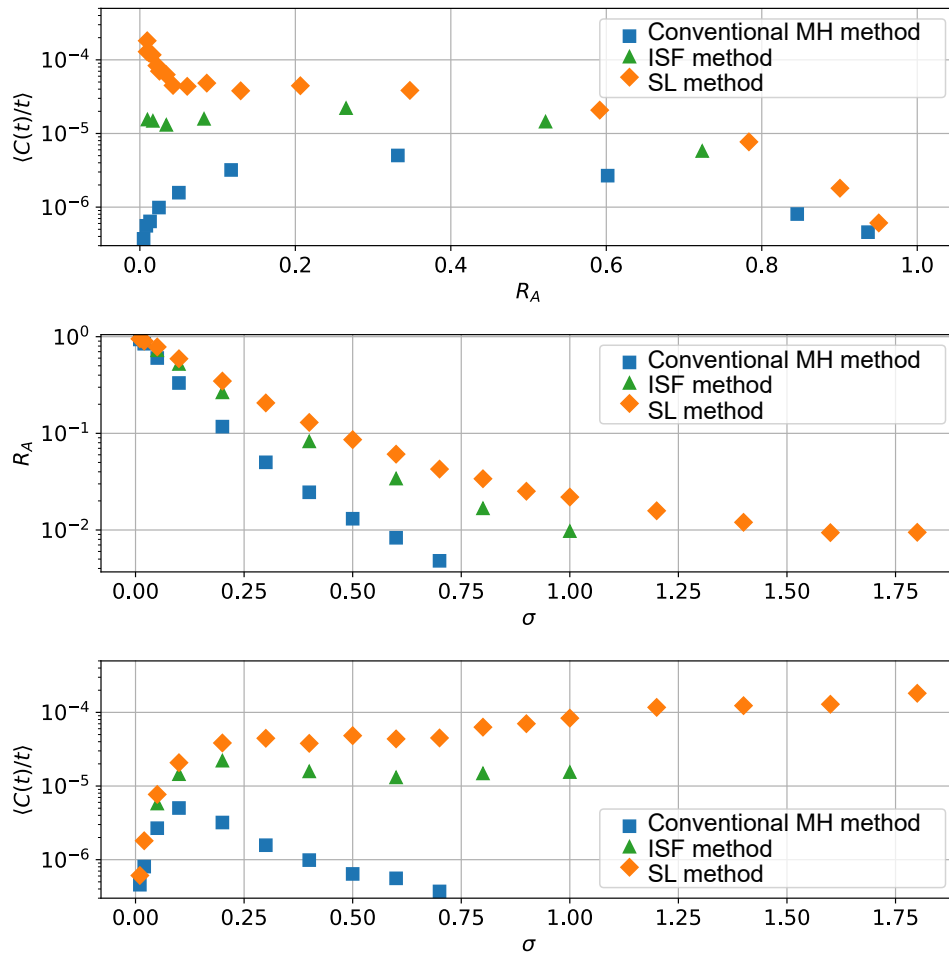


図 7.5 カノニカルアンサンブルシミュレーション下の LJ 粒子系を用いた効率評価. MH 法, ISF 法, SL 法の各手法において提案分布の移動半径 σ を変更した際の効率評価指標の変化の様子を示している. 各図はそれぞれ (a) 採択率 R_A と平均二乗変位の比例定数 $\langle C(t)/t \rangle$, (b) 移動半径 σ と採択率 R_A , (c) 移動半径 σ と平均二乗変位の比例定数 $\langle C(t)/t \rangle$ の変化の様子である.

の低下は確認されるものの, その低下は緩慢であり, 結果的に比例定数が大きく上昇していることが分かる (図左上の領域).

以上の結果から, カノニカルアンサンブルにおける提案手法の効率向上を定量的に確認することができた.

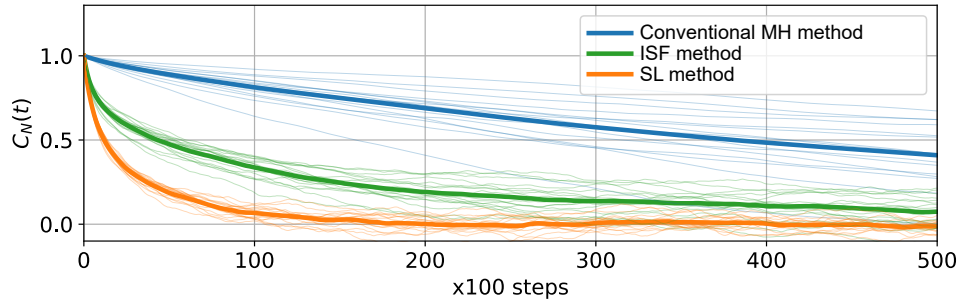


図 7.6 グランドカノニカルアンサンブルシミュレーション下の LJ 粒子系を用いた効率評価. MH 法, ISF 法, SL 法の各手法において, 粒子数のステップ数自己相関関数 $C_N(t)$ をそれぞれ示している. 各手法ごとに 10 回の独立なシミュレーションを実行しており, 細線はそれぞれの実験の自己相関関数に対応している. これら 10 回の平均を太線で示している.

7.7.2 グランドカノニカルアンサンブルシミュレーションによる実験

MFMH 法の効率の評価のため, グランドカノニカルアンサンブル下の LJ 粒子系を用いてシミュレーションを行った. まず, シミュレーションの対象となる系の設定について述べる.

系に含まれる粒子は前節のカノニカルアンサンブルと同じ相互作用をもつ LJ 粒子である. また, 系はグランドカノニカルアンサンブルに従い, 対応するボルツマン分布は

$$p_B(\mathbf{r}) \propto \exp \left[-\beta \sum_{i \neq j} U(r_{ij}) + \mu N \right] \quad (7.56)$$

となる. ここで β は系の逆温度であり, $\beta = 1/4$ と設定している. また μ は系の化学ポテンシャルであり, 平均の粒子数が前節のカノニカルアンサンブルとほぼ同等の 256 近くとなるように $\mu = 7.7$ を選んでいる. 系には周期境界条件が設定されており, 周期セルの大きさは一辺 $2^{8/3}$ の立方体となっている.

以上の系に対して, MH 法, ISF 法に基づく MFMH 法, SL 法に基づく MFMH 法のそれぞれを用いて 10^6 ステップのシミュレーションを行った. ただし SL 法のパラメータはカノニカルアンサンブルシミュレーションの実験で求めたものを用いる. グランドカノニカルアンサンブルでは, カノニカルアンサンブルとは異なり, 粒子数が確率的に挿入・削除されることによって, 粒子数が変動する. MC シミュレーションでは, なるべく多様な状態をサンプリングする必要があるから, なるべく短いステップ数でさまざまな粒子数

表 7.1 各手法による実験結果から得られた粒子数の自己相関関数の時定数と平均採択率

Method	Relaxation steps of $C_N(t)$	R_A
Conventional MH	60.7 k±24.1 k	0.100 %±0.007 %
ISF	8.91 k±20.8 k	2.25 %±0.07 %
SL	1.93 k±0.50 k	4.02 %±0.06 %

の状態が得られる方が望ましい。この観点からグランドカノニカルアンサンブルシミュレーションの効率性を評価するために、粒子数のステップ数自己相関関数

$$C_N(t) = \frac{\langle (N(t+\tau) - \hat{N})(N(\tau) - \hat{N}) \rangle_\tau}{\langle (N(\tau) - \hat{N})^2 \rangle_\tau} \quad (7.57)$$

を用いる。ここで $N(t)$ はステップ数 t における状態の粒子数であり、また $\hat{N} = \langle N(\tau) \rangle_\tau$ は平均の粒子数である。この自己相関関数 $C_N(t)$ は、あるステップ数 t だけ離れた状態間の粒子数の相関係数（3章参照）を表しており、グランドカノニカルアンサンブルに従う独立なサンプル間の相関はゼロになることから、自己相関関数 $C_N(t)$ が早い速度で減少する方が、効率的であると言える。

MH法、ISF法、SL法の計算結果に基づく粒子数のステップ数自己相関関数 $C_N(t)$ を図 7.6 に示す。図 7.6 から、各手法の自己相関関数はステップ数とともに減少しており、いずれの手法も長いステップ数をかけることによって独立なサンプルを得られていることが分かる。各手法の違いを観察すると、SL法・ISF法・MH法の順に早い速度で自己相関関数が減少しており、この順でサンプリングの効率が高いことが分かる。この減少速度を定量的に見積もるために、自己相関関数の時定数、すなわち自己相関関数の値が e^{-1} を下回るステップ幅を計算したところ、表 7.1 の通りになった。表 7.1 には、自己相関関数の時定数に加えて粒子の挿入・削除の平均採択率 R_A も記載している。表 7.1 より、ISF法ではMH法の約6倍、SL法ではMH法の約30倍のステップ数で独立なサンプルを得られているということになる。これは、グランドカノニカルアンサンブルシミュレーションにおいても提案手法の有効性を示している。

7.8 結論と今後の展望

本章で行った研究について、結論と今後の展望を述べる。

7.8.1 結論

本章では MC シミュレーションを機械学習によって高速化する手法について調べ、以下の通り報告した。

1. MH 法の提案分布から近接配置を取り除く枠組みとして、粒子系における確率分布を変換する CNMF 法を提案した。
2. CNMF 法を用いてサンプリングを行うアルゴリズムとして MFMH 法を提案した。
3. CNMF 法における具体的な分布変換の構成法として、解析的にもとめた ISF 法を提案した。
4. CNMF 法における具体的な分布変換の構成法として、自己学習によりパラメータを決定する SL 法を提案した。
5. その結果、LJ 粒子系において、カノニカルアンサンブルシミュレーションではステップあたり最大 20 倍の粒子移動量を達成し、グランドカノニカルアンサンブルシミュレーションでは最大 30 倍の速度で粒子数に関する独立なサンプルを得ることに成功した。

7.8.2 今後の展望

今後の展望として、提案手法を用いたシミュレーションをより複雑な系へと適用して、既存の MC シミュレーションでは不可能であった計算を行うことができるようになると考えられる。

また、本研究の提案手法自体の改善方策としては、CNMF は 2 体間相互作用のみで分布変換をしており、分布の表現能力は比較的小さいものである。そのため、分布変換を学習する SL 法においても、潜在空間においてもボルツマン分布の複雑な形状が残ってしまうため、ランダムに選んだ粒子を微小に移動する潜在空間上の提案分布を用いる必要がある。しかしながら、3 体間以上の相互作用や、グラフ畳み込み演算による相互作用などを導入することによって、ボルツマン分布を強力的に平らな分布へと変換することができれば、ボルツマン分布からの独立なサンプリングが実現する可能性がある。

第 8 章

ゲート式量子コンピュータによる MC シミュレーションの高速化に向けた検討

本章では、量子コンピュータを活用して任意の分布から高速にサンプリングをする手法について調べた研究を報告する。

8.1 本章の目的と概要

ベイズ推定における事後分布や統計力学におけるボルツマン分布など、複雑な分布から独立なサンプルを高速に得たいという需要がある。しかしながら、前章で述べた通り分布形状が複雑であればあるほど独立なサンプリングを得るのが難しいという問題がある。

基本的かつ汎用的なサンプリングアルゴリズムの一つである MH 法は、目的とする分布の代わりに提案分布とよばれる別の分布からサンプリングを行い、それは適切に棄却採択をすることで目的の分布を得るというアルゴリズムなのであった。また、MH 法の性質から、効率的なサンプリングのためには提案分布はは大別して次の 3 つの性質を満たす必要がある。(1) 提案分布それ自体は高速にサンプリングできるものであること、(2) サンプリングした値においての提案分布の確率の値が簡単に計算できること、(3) 提案分布が目的とする分布となるべく近いこと、である。前章では、この 3 つの性質を満たしながらも機械学習によって適切な分布を構成する CNMF 法を提案することによって、MC シミュレーションを高速化した。

前章の中でも、パラメータによって可変な提案分布を設定して、サンプリングの過程で

提案分布を目的の分布へ近づけていくことによりパラメータを決定することで、サンプリングを効率を高める方式の自己学習 (SL) モンテカルロ法は、近年分子シミュレーションのみならず様々な分野のモンテカルロサンプリングで用いられている [86, 87, 88, 89]. しかしながら、この自己学習の考え方を汎用的に適用しようとする、高速にサンプリングすることが可能であるようなパラメータ付き汎用提案分布を構成する必要がある. もともとサンプリングが難しい分布を簡単にサンプリングするための提案分布が、高い汎用性を要求した結果、サンプリングが難しくなってしまうのは、元も子もないのである.

ところで、量子コンピュータにおける状態である量子状態は、原理的に確率的であり、量子状態に対応する確率分布からのサンプリングは、単に測定をするだけで実行できる. 実際、ボソンサンプリング [90, 91] に代表されるように、古典コンピュータでは容易にサンプリングすることが難しい確率分布を、量子コンピュータでは高速にサンプリングできる量子超越性が知られている. したがって、これらの量子状態の性質を利用して、汎用的で効率的なサンプリングアルゴリズムを構成できないかを検討するのは妥当だろう.

そこで本研究では、提案分布の構成のために量子コンピュータを利用することにより、汎用的な自己学習モンテカルロアルゴリズムを構築することを目的とする. ただし、量子コンピュータによる提案分布としては様々なモデルが考えられるが、本研究では研究の方向性を限定するために、特に量子フーリエ変換回路 (Quantum Fourier transform, QFT)[92] を用いたモデルを調べることにする.

次節以降では、まず、量子コンピュータを用いた汎用的な自己学習サンプリングアルゴリズムを示している. これは QFT 回路に限らず量子コンピュータを用いた場合の自己学習方式の MH アルゴリズムについて定式化している. ここでは、このアルゴリズムの効率性として重要である上述の提案分布の 3 つの性質についても議論している.

次に、1次元の確率分布を対象として、QFT を用いた自己学習 MH アルゴリズムである 1次元 QFT サンプラーについての定式化を行っている. その次に、このアルゴリズムが一定の制約下においては古典コンピュータで効率的に再現できることを示している. このように、量子アルゴリズムをある制約下において古典コンピュータ上で効率的に実行できるようなアルゴリズムを構成することを、quantum-inspired アルゴリズムと呼び、近年量子コンピュータ研究の 1 分野として注目を集めている [93, 94, 95]. さらにその次には、QFT サンプラーを多次元化することで、多次元分布に対しても適用できる手法を述べている.

これらの提案手法を評価するために、いくつかの模擬的な確率分布に対して、QFT サンプラーを適用し、ベースラインとなる MH アルゴリズムとの比較を行っている. その結果、1次元の分布に対して適用する 1次元 QFT サンプラーの性能評価と、多次元分布

に対しての多次元 QFT サンプラーの性能評価のそれぞれについて、汎用的に効率向上が達成できていることが確認された。最後に、結果の考察と今後の展望について述べる。

8.2 量子コンピュータを用いた汎用的な自己学習サンプリングアルゴリズムの定式化

量子コンピュータの基本的な考え方と、量子コンピュータを利用した汎用的な自己学習サンプリングアルゴリズムの定式化の方法について述べる。

8.2.1 自己学習サンプリングアルゴリズムについて

この節では、量子コンピュータを用いた自己学習 MH 法の汎用的な形式について議論する。 $p(x)$ をサンプリングを行いたい目標の分布として、それに対応する MH 法の提案分布を $q(x)$ とする。MH 法では、目標の分布 $p(x)$ からのサンプリングを直接得ることが難しいとして、提案分布 $q(x)$ からのサンプリングを行う。さらに、提案分布から得られた候補サンプルを確率的に棄却・採択することで、採択されたサンプルが目標の分布 $p(x)$ に従うようにする。より具体的には、採択された一番直近の状態を \mathbf{r} 、提案分布からの候補状態を \mathbf{r}^* としたとき、この候補状態 \mathbf{r}^* を採択率

$$A(\mathbf{r}, \mathbf{r}^*) = \min \left(\frac{p(\mathbf{r}^*) q(\mathbf{r})}{p(\mathbf{r}) q(\mathbf{r}^*)}, 1 \right) \quad (8.1)$$

の確率で採択する。採択されなければ直近の状態をそのまま次の状態として用いる。ただし、一般的には提案分布として直前の状態に依存した形の提案分布 $q(\mathbf{r}^*|\mathbf{r})$ を用いるが、本章では独立なサンプリングを高速に実行することを実現したいため、提案分布自体が直前の状態に依存しない独立なサンプリングをするもの $q(\mathbf{r}^*)$ に議論を限ることにした。

このプロセスを繰り返し実行することにより得られたと採択サンプルは、目標の分布 $p(x)$ に従うことが詳細つり合い条件を満たすことから証明できる。上式からわかる通り、目標の分布 $p(x)$ については、値 x が与えられたときの確率の値（正規化定数を除く）は高速に計算できるが、 $p(x)$ からのサンプリングが難しいという状況を仮定している。この仮定は、ボルツマン分布やベイズモデリングなどで一般的に成り立つ過程である。

さて、このプロセスに関して、提案分布と目標分布が一致するとき ($q(x) \equiv p(x)$)、採択率は 1 となり、全てのサンプルが採択される。このとき、提案分布をは直前の状態に依存しないものを選んでいたので、全て独立なサンプリングとなっているから、モンテカルロ法の効率は最大になる。しかしながら、 $p(x)$ のサンプリングが難しいからという理由

で MH 法では提案分布を代わりに使用しているわけであるから，提案分布を目標分布と一致させることは一般にはできない．このことから，なるべく効率よくプロセスを上手く回すためには，提案分布それ自体は高速にサンプリングできるものであって，かつ提案分布と目標分布がなるべく近いものである必要があるということになる．加えて，提案分布からサンプリングされた候補状態から，詳細つり合い条件による採択率（式 8.1）を計算しなければならないから，サンプリングした値における提案分布の確率の値が簡単に計算できることが必要である．

よってまとめると，提案分布としては次の 3 つの性質

1. 提案分布それ自体は高速にサンプリングできるものであること
2. サンプリングした値における提案分布の確率の値が簡単に計算できること
3. 提案分布が目的とする分布となるべく近いこと

が要求される．そこで，自己学習サンプリングアルゴリズムの観点からは，パラメータ θ によって決定づけられる提案分布 $q(\theta)$ があって，有効な範囲のどのパラメータを選んでも上記 3 つの性質を満たすようなものを設計する必要がある．ひとたびそのような提案分布が構成できれば，パラメータ θ を調整し，提案分布を目標の分布 $p(x)$ に最も近づけることで効率の良いサンプリングが達成できると考えられる．

8.2.2 量子コンピュータに特有な概念と表記について

量子コンピュータを用いた議論の前に，量子コンピュータに特有な概念と表記について簡単に整理する．量子コンピュータでは，0 か 1 のどちらかをとる古典的なビットとは異なり，0 と 1 の状態の重ね合わされた状態をとることができる量子ビットを状態として持つ．このため，量子コンピュータにおいては，古典コンピュータでにおいて 0 に相当する量子ビットを $|0\rangle$ ，1 に相当する量子ビットを $|1\rangle$ とすれば，量子ビットは

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (8.2)$$

と表現される．ここで α, β は複素数であり， $\|\alpha\|^2 + \|\beta\|^2 = 1$ を満たす範囲に制限されている．単に 0 と 1 の（実数）内分点の状態を取るのではなく，複素数和の形を取っていることに注意する．結局，1 量子ビットは α, β の 2 つの複素自由度を持つことになるから，上式の量子状態 $|\phi\rangle$ は

$$|\phi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (8.3)$$

とベクトル表記することができる。この表記によれば、 $|0\rangle = (1, 0)$, $|1\rangle = (0, 1)$ である。

量子状態は、その重ね合わせ状態を直接測定することはできない。これは、量子力学的に基づく重ね合わせ状態は、測定によってただ一つの基底に収縮してしまうためである。この収縮は、量子状態に依存した確率的な振る舞いを示し、式 8.2 の状態を基底 $(|0\rangle, |1\rangle)$ で測定すると、0 が確率 $\|\alpha\|^2$, 1 が確率 $\|\beta\|^2$ で測定されることになる。このような量子状態の確率的な特性は、古典力学では見られないものである。

複数の量子ビットがある場合、その量子状態はとり得る古典的状态のすべての組み合わせを重ね合わせた状態を取る。すなわち、2 量子ビットの場合は、 $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, $|1\rangle \otimes |1\rangle$ の 4 つ組の基底状態によって、2 量子ビット状態

$$|\phi\rangle = \alpha_{00} |0\rangle \otimes |0\rangle + \alpha_{01} |0\rangle \otimes |1\rangle + \alpha_{10} |1\rangle \otimes |0\rangle + \alpha_{11} |1\rangle \otimes |1\rangle \quad (8.4)$$

が定義される。ここで、 α_{ij} は複素数であり、測定の際には 00, 01, 10, 11 の基底状態がそれぞれ $\|\alpha_{00}\|^2$, $\|\alpha_{01}\|^2$, $\|\alpha_{10}\|^2$, $\|\alpha_{11}\|^2$, の確率で測定されることになる。また、このことから、2 量子ビット状態のベクトル表現は、

$$|\phi\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \quad (8.5)$$

となる。加えて、2 ビットの量子状態を簡潔に表記するために、次の表記

$$|0\rangle \otimes |0\rangle \equiv |00\rangle \equiv |0\rangle \quad (8.6)$$

$$|0\rangle \otimes |1\rangle \equiv |01\rangle \equiv |1\rangle \quad (8.7)$$

$$|1\rangle \otimes |0\rangle \equiv |10\rangle \equiv |2\rangle \quad (8.8)$$

$$|1\rangle \otimes |1\rangle \equiv |11\rangle \equiv |3\rangle \quad (8.9)$$

が用いられることがある。以上の議論は、3 ビット以上の量子状態にも同様に適用される。たとえば、 N 量子ビットの状態は、ベクトル表現では 2^N 成分の複素ベクトルとなる。

これらの量子状態に対して、量子コンピュータは、量子状態のベクトル表現に対して任意のユニタリ行列 U を掛けることによって得られる量子状態を生成することができる。すなわち、ある量子状態 $|\phi\rangle$ があって、あるユニタリ行列 U を用いて

$$|\psi\rangle = U |\phi\rangle \quad (8.10)$$

という状態を得ることができる。ここで、量子状態の演算に暗黙的にベクトル表現が適用されていることに注意されたい。量子コンピュータは、量子状態に対して適切なユニタリ行列を適用することで、目的の計算を行うものである。

8.2.3 量子コンピュータを用いた自己学習サンプリングアルゴリズム

さて、話を戻して、量子コンピュータを用いた場合の自己学習サンプリングアルゴリズムはどうなるかを議論する。量子コンピュータにおいては（一般に）基底状態 $|0^N\rangle = |0\rangle^{\otimes N}$ が初期状態として与えられて、これにパラメータ付きのユニタリゲート $U(\theta)$ を適用して量子状態

$$|\psi(\theta)\rangle = U(\theta) |0^N\rangle \quad (8.11)$$

を得る。ここではパラメータ θ を複素数であると仮定している。このように得られた量子状態 $|\psi(\theta)\rangle$ を測定することによって得られるサンプルが従う確率分布を提案分布 $q(x; \theta)$ とすれば、この提案分布は

$$q(x; \theta) = \|\langle x | \psi(\theta) \rangle\|^2 = \|\langle x | U(\theta) |0^N\rangle\|^2 \quad (8.12)$$

に従う。ここで x は 0 から $2^N - 1$ までのいずれかの整数である。よって、0 から $(2^N - 1)$ までの整数を状態空間とする提案分布を構成したことになる。自己学習サンプリングアルゴリズムでは、提案分布 $q(x; \theta)$ を目的の分布 $p(x)$ になるべく一致させる必要があるから、 θ を最適化することが必要なのであった。量子コンピュータを用いた場合では、以下の通り導入する損失関数を勾配法を用いて最小化する。

まず、ある固定のパラメータ θ が与えられたとして、量子状態 $|\psi(\theta)\rangle$ を測定することによって、提案分布 $q(x; \theta)$ に従うサンプルを得る。これらのサンプルは式 8.1 に基づく採択率によって棄却・採択され、結局目的の分布 $p(x)$ に従うサンプルを得る。ここで、量子コンピュータでは、測定がそのままサンプリングとなるから、3つの性質のうち1つ目「提案分布それ自体は高速にサンプリングできるものであること」は原理的に満たされることになる。一方、2つ目の「サンプリングした値における提案分布の確率の値が簡単に計算できること」については、用いるユニタリゲート $U(\theta)$ によってその容易さが変わるだろう。ただし、一般には振幅推定アルゴリズムによって $q(x; \theta) = \|\langle x | \psi(\theta) \rangle\|^2$ を計算できることが知られている。3つ目の性質として、提案分布集合の中に目的分布と近いものが含まれているかどうかは、もちろん、選択するユニタリゲート $U(\theta)$ の種類による、本研究ではそのような性質を持つ汎用的なユニタリゲートとして、QFTゲートを選択している。QFTゲートの詳細は次節に述べることとして、ここでは与えられた提案分布集合の中から適切な提案分布を選ぶ方法について述べる。

提案分布 $q(x; \theta)$ を目的の分布 $p(x)$ になるべく一致させるために、提案分布と目的の分

布がどれだけ一致しているかの指標として次のクロスエントロピー誤差関数

$$L(\boldsymbol{\theta}) = - \sum_x p(x) \log q(x; \boldsymbol{\theta}) \quad (8.13)$$

を損失関数として用いることにする。クロスエントロピー誤差関数は2つの分布 $p(x)$ と $q(x)$ が一致するときに限り最小値を取る、分布間の差の比較として一般的に用いられている損失関数の一つである。従って、この損失関数が最小となるようなパラメータ $\boldsymbol{\theta}$ を求めることで、提案分布集合の中から適切な提案分布を選べる。そこで、損失関数を最小とするパラメータを用いるために、最急降下法を用いた探索アルゴリズムを使用する。最急降下法は、パラメータ空間において損失関数の勾配の逆方向に少しずつ進むことによって最適化を行う方法であるが、今回は損失関数のパラメータが複素数であることから、勾配の計算について注意する必要がある。あるパラメータ $\boldsymbol{\theta}$ において、パラメータが $\boldsymbol{\theta}' = \boldsymbol{\theta} + \delta\boldsymbol{\theta}$ と変化したとき、損失関数の変化 δL は

$$\delta L = \left(\frac{\partial}{\partial \boldsymbol{\theta}} L \right)^T \delta\boldsymbol{\theta} + \left(\frac{\partial}{\partial \boldsymbol{\theta}^*} L \right)^T \delta\boldsymbol{\theta}^* \quad (8.14)$$

$$= \left(\frac{\partial}{\partial \boldsymbol{\theta}} L \right)^T \delta\boldsymbol{\theta} + \left(\frac{\partial}{\partial \boldsymbol{\theta}} L \right)^\dagger \delta\boldsymbol{\theta}^* \quad (8.15)$$

となる。従って、パラメータ空間に損失関数の値を小さくするためには、

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \left(\frac{\partial}{\partial \boldsymbol{\theta}} L \right)^* \quad (8.16)$$

と更新すればよいことがわかる。損失関数の勾配の代わりに、勾配の複素共役が用いられている点が、実数空間における最急降下法と異なる。続けて、式 8.13 のクロスエントロピー誤差関数の勾配は、

$$\frac{\partial}{\partial \boldsymbol{\theta}} L(\boldsymbol{\theta}) = - \sum_x \frac{p(x)}{q(x; \boldsymbol{\theta})} \frac{\partial}{\partial \boldsymbol{\theta}} q(x; \boldsymbol{\theta}) \quad (8.17)$$

$$= - \sum_x q(x; \boldsymbol{\theta}) \frac{p(x)}{q(x; \boldsymbol{\theta})^2} \frac{\partial}{\partial \boldsymbol{\theta}} q(x; \boldsymbol{\theta}) \quad (8.18)$$

$$= - \left\langle \frac{p(x)}{q(x; \boldsymbol{\theta})^2} \frac{\partial}{\partial \boldsymbol{\theta}} q(x; \boldsymbol{\theta}) \right\rangle_{q(x; \boldsymbol{\theta})} \quad (8.19)$$

と計算できる。上式の最後の式は、提案分布のサンプルに対する期待値となっているから、提案分布 $q(x; \boldsymbol{\theta})$ からのサンプリング列 r_1, r_2, \dots, r_B によってこれを近似すれば、

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \sim - \frac{1}{B} \sum_{i=1}^B \frac{p(x)}{q(x; \boldsymbol{\theta})^2} \frac{\partial}{\partial \boldsymbol{\theta}} q(x; \boldsymbol{\theta}) \quad (8.20)$$

と計算することができる。ここで B は近似の際のサンプル数である。つまり、提案分布 $q(x; \boldsymbol{\theta})$ から得られた候補状態のサンプルをもとに、この勾配近似を計算し、棄却採択プロセスと同時に式 8.20 の勾配によるパラメータの更新を行う。ここで、パラメータ $\boldsymbol{\theta}$ における提案分布の値 $q(x; \boldsymbol{\theta})$ の勾配は、

$$\frac{\partial}{\partial \boldsymbol{\theta}} q(x; \boldsymbol{\theta}) = \left[\left\| \langle x | \left(\frac{\partial}{\partial \theta_1} \mathbf{U}(\boldsymbol{\theta}) \right) |0^N\rangle \right\|^2, \dots, \left\| \langle x | \left(\frac{\partial}{\partial \theta_m} \mathbf{U}(\boldsymbol{\theta}) \right) |0^N\rangle \right\|^2 \right] \quad (8.21)$$

と計算できる。

8.3 1次元 QFT サンプラー

ここまで、量子コンピュータを用いた自己学習サンプリングアルゴリズムの一般形について述べてきたが、このアルゴリズムを実際に実行するにはユニタリ行列 $\mathbf{U}(\boldsymbol{\theta})$ を具体的に決定する必要がある。本研究では、上記の量子版自己学習サンプリングアルゴリズムの概念実証として、QFT 回路と呼ばれるユニタリ行列を選び実装を進めていく。本研究においては、以降に述べる通り提案分布としての望ましい性質を持っているため、QFT 回路を選択しているが、QFT 回路より適切な構成が存在することについては十分な可能性があると考えている。

本節では、QFT 回路を 1 次元分布のサンプラーとして構成する。2 次元以上の分布については、別途以降の節に述べることにする。QFT 回路は、パラメータを量子状態へ埋め込んだ次の状態

$$|\phi(\boldsymbol{\theta})\rangle = \theta_0 |0\rangle + \theta_1 |1\rangle + \dots + \theta_{2^M-1} |2^M - 1\rangle \quad (8.22)$$

を用いて、QFT 回路への入力状態を

$$|\text{in}\rangle = |\phi(\boldsymbol{\theta})\rangle \otimes |0^{N-M}\rangle \quad (8.23)$$

として得られる状態

$$|\psi(\boldsymbol{\theta})\rangle = \mathbf{U}_{QFT} |\text{in}\rangle \quad (8.24)$$

からのサンプリングを提案分布として用いる (図 8.1)。ここでパラメータ $\boldsymbol{\theta}$ は 2^M 次元複素ベクトルであり、状態 $|\phi(\boldsymbol{\theta})\rangle$ は M 量子ビット状態により表現されている。したがって 2^M 個のパラメータを持つ。パラメータ $\boldsymbol{\theta}$ が実数ではなく複素数であるのは、実数のみの入力であると QFT 回路が偶関数分布しか表現できないためである。本研究では、パ

ラメータを持つ状態として各計算基底の重み和という形で入力状態を表現しているが、ハードウェア実装効率の良いその他の状態表現 [96] を用いても問題ないと考えられる。一方、測定すべき状態 $|\psi(\boldsymbol{\theta})\rangle$ は N 量子ビット状態により表現されており、 2^N 個の状態 $|0\rangle, |1\rangle, \dots, |2^N - 1\rangle$ を計算基底に持ち、測定結果 x は 0 から $2^N - 1$ までのいずれかの整数となる。ここで、QFT 回路は

$$(U_{QFT})_{ij} = \frac{1}{\sqrt{2^N}} e^{i2\pi kj/2^N} \quad (8.25)$$

と定義されるユニタリ行列である。

以上の設定から、提案分布として用いる状態 $|\psi(\boldsymbol{\theta})\rangle$ のサンプリングは、確率分布

$$q_{QFT}(x; \boldsymbol{\theta}) = \|\langle x | \psi(\boldsymbol{\theta}) \rangle\|^2 = \|\langle x | U_{QFT} | \text{in} \rangle\|^2 \quad (8.26)$$

に従うことになる。入力状態は式 8.23 の通り後ろの $N - M$ ビットがすべてゼロ状態 $|0\rangle$ となっているから、これはさらに

$$q_{QFT}(x; \boldsymbol{\theta}) = \mathbf{u}_x \boldsymbol{\theta} \quad (8.27)$$

と展開できる。ただし式中のベクトル \mathbf{u}_x は、QFT ゲート行列はの x 行目のベクトルの初めの 2^M 要素を取ってきたベクトル

$$(\mathbf{u}_x)_j = \frac{1}{\sqrt{2^N}} e^{i2\pi xj/2^N} \quad (8.28)$$

である。また、損失関数の勾配の計算に必要な提案分布に対するパラメータの勾配は、

$$\frac{\partial}{\partial \boldsymbol{\theta}} q_{QFT}(x; \boldsymbol{\theta}) = (\mathbf{u}_x^\dagger \boldsymbol{\theta}^*) \mathbf{u}_x \quad (8.29)$$

と計算することができるから、式 8.20 より、損失関数の勾配は

$$\left(\frac{\partial}{\partial \boldsymbol{\theta}} L(\boldsymbol{\theta}) \right)^* \simeq -\frac{1}{B} \sum_{i=1}^B \frac{p(r_i)}{q_{QFT}(r_i; \boldsymbol{\theta})^2} (\mathbf{u}_x^\dagger \boldsymbol{\theta}^*) \mathbf{u}_x \quad (8.30)$$

と計算できる。ここで (r_1, r_2, \dots, r_B) は提案分布 q_{QFT} から得られたサンプル列である。これを前節の通り学習と棄却採択を同時に行うことによって、効率的に $p(x)$ に従うサンプル集合を得る。これを、(1次元) QFT サンプラーと呼ぶことにする。

さて、ここで QFT サンプラーが提案分布としてのふさわしい 3 つの性質を満たしているかについて確認する。1 つ目の「提案分布それ自体は高速にサンプリングできるものであること」については、前節のとおり量子コンピュータを用いたサンプリングは原理的に

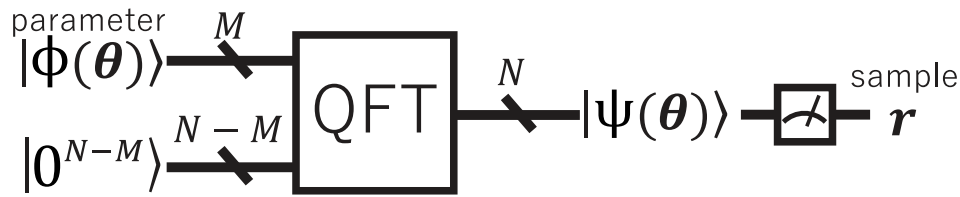


図 8.1 1次元 QFT サンプラーの模式図． ([97] より引用) M 量子ビットのパラメータベクトルから N 量子ビットの量子状態を生成し，測定することによりサンプリングを行う．

高速である．2つ目の「サンプリングした値における提案分布の確率の値が簡単に計算できること」については，提案分布は単に式 8.27 の通りパラメータベクトルと測定値に対応するベクトルの内積を計算すればよいためとなっており，これは古典コンピュータで高速に計算できるから，QFT サンプラー特有の性質としてこれを満たしている．また，3つ目の「提案分布が目的とする分布となるべく近いこと」について，式 8.26 から，入力状態 $|\text{in}\rangle$ は提案分布関数 q_{QFT} のフーリエ級数となっていることが分かる．つまり，任意の提案分布関数 q_{QFT} は，入力状態 $|\text{in}\rangle$ を適切に選ぶことで表現できることになる．本研究の設定では，入力状態 $|\text{in}\rangle$ は最初の M 量子ビットのみパラメータが与えられ，それ以外はゼロ状態となっているが，これは提案分布関数 q_{QFT} のうち低周波成分のみをパラメータとして設定することに相当する．これにより任意性はなくなるが，分布の低周波成分を一致させることができる．従って，提案分布関数 q_{QFT} が目的の分布 $p(x)$ と近いものを選ぶことになる．以上より，3つの条件を満たしていることが分かる．

8.4 古典コンピュータによる Quantum-inspired QFT サンプラー実装

上述の QFT サンプラーは，量子コンピュータ上で QFT 回路を適用する必要がある．QFT 回路は，フォールトトレラントな量子計算^{*1}が要求され現在の量子コンピュータでは直接実行できない．しかしながら， M が N より十分に小さいときは，QFT 回路の適応的測定 [98] と呼ばれる手法によって古典コンピュータによって高速に実行が可能である．

本来， N 量子ビットの量子回路を古典コンピュータ上でシミュレーションするためには，一般に 2^N 次元の状態ベクトルに対する $2^N \times 2^N$ の行列演算を計算する必要がある

^{*1} 量子計算の際に発生する雑音を補正して正確な計算を行うことができる方法．現時点でそのような計算を実行できる量子コンピュータは存在しない．

るから、少なく見積もったとしても $O(2^N)$ 以上の計算量が必要である。これは例えば $N = 100$ の時は 2^{100} の計算となって、もはや古典コンピュータで扱える計算量ではない。ただし、一定の条件下にある量子計算および量子測定は、古典コンピュータでも十分に扱える計算量となる場合がある。具体的には、QFT 回路は、適応的測定法を用いることにより、パラメータベクトルの量子ビット数 M と全体の量子ビット数 N に対して $O(2^M + N)$ の計算量で計算することができる。よって、 M が比較的小さい範囲では、古典コンピュータを用いて高速に計算できる。

従って、提案分布としてのふさわしい3つの性質として唯一量子コンピュータの能力が必要であった性質 (1) 「提案分布それ自体は高速にサンプリングできるものであること」についても、 M が十分に小さい範囲では古典コンピュータで実現可能であるということになる。よって、QFT サンプラーを古典コンピュータで高速に実行することにより、Quantum-inspired アルゴリズムとしての QFT サンプラーを導入し、以降はこれを用いることにする。

8.5 多段化による多次元 QFT サンプラー

前節までに述べた QFT サンプラーは 1 次元分布にのみ対応しており、多次元分布については別途拡張を考える必要がある。拡張の方向性としては、QFT 回路を拡張して 2 次元 QFT 回路を採用することも考えられるが、本研究においては QFT サンプラーを各次元ごとに積み重ねる、多段 QFT サンプラーを提案する。多段 QFT サンプラーは図 8.2 に示すように構成される。 D 次元の候補状態 $\mathbf{x} = (x_1, x_2, \dots, x_D)$ を生成するために、まず 1 次元 QFT サンプラーによって x_1 をサンプリングする。次に x_2 を 1 次元 QFT サンプラーによりサンプリングするが、その際に、QFT サンプラーの入力パラメータ $\boldsymbol{\theta}$ がサンプル x_1 に依存するように決定する。すなわち、何かしらの関数モデル $f_2(x_1, \boldsymbol{\theta}_2)$ を用いて、2 次元目の状態 x_2 を

$$q(x_2) = q_{QFT}(x_2; \mathbf{f}_2(x_1, \boldsymbol{\theta}_2)) \quad (8.31)$$

として、 x_1 に間接的に依存するように決定する。ここで、2 次元目の QFT サンプラーにパラメータベクトルとして入力されるのは、 $\boldsymbol{\theta}_2$ ではなく $\mathbf{f}_2(x_1, \boldsymbol{\theta}_2)$ となることに注意する。したがって、パラメータ $\boldsymbol{\theta}_2$ は f_2 のが持つパラメータの設定に依ることになる。同様に、3 次元目以降の状態もそれまでにサンプリングされた次元の状態間接的に依存するような 1 次元 QFT サンプラーによってサンプリングされる。これをまとめると、図 8.2 の

多段階 QFT サンプラーによる提案分布が構成される。このとき、提案分布の確率分布は

$$q(\mathbf{x}, \boldsymbol{\theta}) = \prod_{k=1}^D q_{QFT}(x_k; \mathbf{f}_k(x_1, \dots, x_{k-1}, \boldsymbol{\theta}_k)) \quad (8.32)$$

となる。ここで $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_D)$ であるとした。このようにすることで、 $\mathbf{f}_1, \dots, \mathbf{f}_D$ を上手く設計すれば、様々な D 次元提案分布を表現できるはずである。

多次元サンプラーの学習も、目的の分布 $p(\mathbf{x})$ と上記の提案分布 $q(\mathbf{x}, \boldsymbol{\theta})$ とのクロスエントロピー誤差関数（式 8.13）を最小化するようにパラメータ $\boldsymbol{\theta}$ を決定すればよい。なお、多次元サンプラーにおけるクロスエントロピー誤差関数の勾配は、

$$\left(\frac{\partial}{\partial \boldsymbol{\theta}_k} L(\boldsymbol{\theta}) \right)^* \simeq -\frac{1}{B} \sum_{i=1}^B \frac{p(\mathbf{r}_i)}{q(\mathbf{r}_i; \boldsymbol{\theta})} \frac{\left(\mathbf{u}_{(\mathbf{r}_i)_k}^T \mathbf{f}_k \right) \mathbf{u}_{(\mathbf{r}_i)_k}^*}{q_{QFT}((\mathbf{r}_i)_k; \boldsymbol{\theta})} \frac{\partial}{\partial \boldsymbol{\theta}_k} \mathbf{f}_k \quad (8.33)$$

と展開できる。ここで、 $\mathbf{r}_1, \dots, \mathbf{r}_B$ は、 D 次元 QFT サンプラーによって生成された候補状態のサンプル列である。

多次元 QFT サンプラーの提案分布を決定づける関数 $\mathbf{f}_1, \dots, \mathbf{f}_B$ については、様々なモデルが考えられる。本研究では、適切なモデルを比較検討するため、恒等 (Identity) モデル・線形基底線形回帰 (LBLR) モデル・非線形基底線形回帰 (NBLR) モデル・ニューラルネットワーク (NN) モデルの 4 つのモデルを検討する。各モデルの定義は、次のようになっている。

恒等 (Identity) モデル

恒等 (Identity) モデルは、次のように定義される

$$\mathbf{f}_k(x_1, \dots, x_{k-1}, \boldsymbol{\theta}_k) = \text{Norm}(\boldsymbol{\theta}_k). \quad (8.34)$$

ここで、 $\boldsymbol{\theta}$ は 2^M 次元の複素数ベクトルであって、 $\text{Norm}(\boldsymbol{\theta}) = \boldsymbol{\theta} / \|\boldsymbol{\theta}\| = \boldsymbol{\theta} / \sqrt{\boldsymbol{\theta}^\dagger \boldsymbol{\theta}}$ である。このモデルは、 x_1, \dots, x_{k-1} とは関係なくパラメータ $\boldsymbol{\theta}_k$ を直接出力しているから、単に 1 次元 QFT サンプラーを各次元に用いているだけである。つまり、恒等モデルによる提案分布は、各次元が独立な確率分布となっていて、以降の多次元サンプラーのベースラインと考える。

線形基底線形回帰 (LBLR)

線形基底線形回帰 (LBLR) は、次のように定義される

$$\mathbf{f}_k(x_1, \dots, x_{k-1}, \boldsymbol{\theta}_k) = \text{Norm}(\mathbf{w}_1^k x_1 + \dots + \mathbf{w}_{k-1}^k x_{k-1} + \mathbf{b}^k) \quad (8.35)$$

ここで、 $\boldsymbol{\theta}_k = (\mathbf{w}_1^k, \dots, \mathbf{w}_{k-1}^k, \mathbf{b}^k)$ であり、それぞれは 2^M 次元の複素ベクトルである。このモデルは、入力を線形回帰するものである。

非線形基底線形回帰 (NBLR) モデル

非線形基底線形回帰 (NBLR) モデルは、次のように定義される

$$\mathbf{f}_k(x_1, \dots, x_{k-1}, \boldsymbol{\theta}_k) = \text{Norm} \left(\sum_{j=1}^J \mathbf{w}_j^k \phi_j^k(x_1, \dots, x_{k-1}) + \mathbf{b}^k \right) \quad (8.36)$$

ここで、 $\boldsymbol{\theta}_k = (\mathbf{w}_1^k, \dots, \mathbf{w}_J^k, \mathbf{b}^k)$ であり、それぞれは 2^M 次元の複素ベクトルである。このモデルは、入力を特徴量ベクトルに変換してから線形回帰する。選択される特徴量 ϕ_j^k は、対象とする実験によって変えることにする。

ニューラルネットワーク (NN) モデル

ニューラルネットワーク (NN) モデルは、次のように定義される

$$\mathbf{f}_k(x_1, \dots, x_{k-1}, \boldsymbol{\theta}_k) = \text{Norm} \circ \text{RI} \circ g_4^k \circ \text{Sg} \circ g_3^k \circ \text{Sg} \circ g_2^k \circ \text{Sg} \circ g_1^k(x_1, \dots, x_{k-1}) \quad (8.37)$$

ここで、 g_j^k はニューラルネットワークの全結合層であり、 $\text{Sg}(\cdot)$ はシグモイド関数である。また、 g_1^k, g_2^k, g_3^k の出力次元は 256、 g_4^k の出力次元は 2^{M+1} である。また、関数 $\text{RI}(\cdot)$ は、入力ベクトルを 2 つに分割したときの前半を \mathbf{p} 、後半を \mathbf{q} としたときに、 $\mathbf{p} + i\mathbf{q}$ を出力する関数であり、実数ベクトルを複素ベクトルへ変換する関数である。関数 $\text{RI}(\cdot)$ はニューラルネットワークが実数ベクトルを出力するものであるために必要となっている。このモデルのパラメータ $\boldsymbol{\theta}_k$ は、使用するニューラルネットワーク（全 5 層の全結合ニューラルネットワーク）の全ての学習すべきパラメータであるとする。

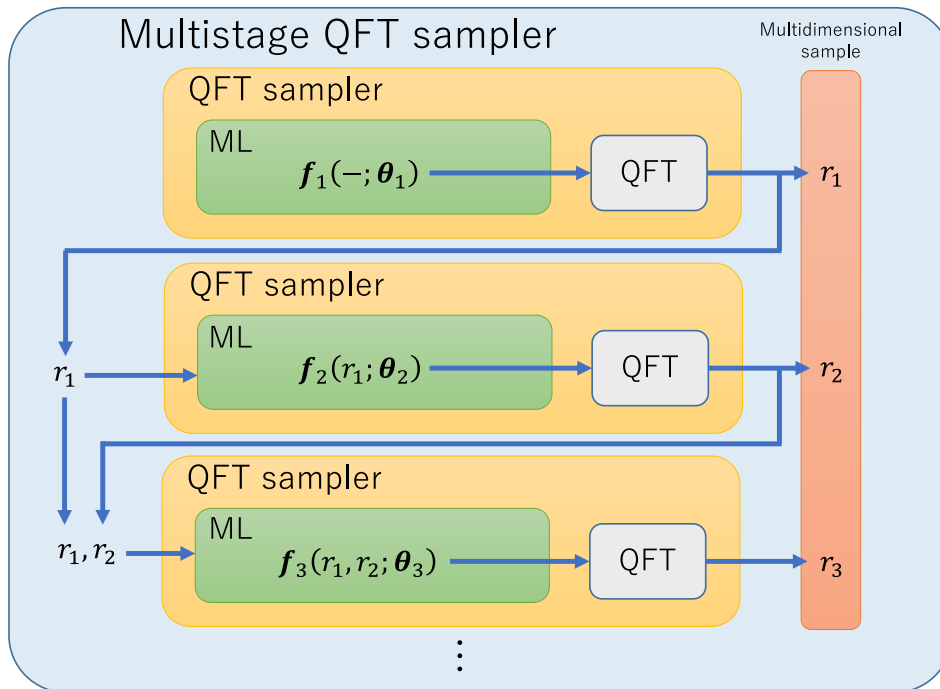


図 8.2 多段 QFT サンプラーの模式図. ([97] より引用) 各成分のごとに 1 次元 QFT サンプラーが使用され, それぞれのパラメータベクトルはそれより前の成分のサンプル結果に依存するようにする.

8.6 QFT サンプラーの性能評価実験

QFT サンプラーを 1 次元分布および 2 次元分布のサンプリングに適用し, 性能を評価する.

8.6.1 1 次元 QFT サンプラーの実験

QFT サンプラーの性能評価のため, 1 次元分布に対して 1 次元 QFT サンプラーを適用することにより, 適切な提案分布の獲得と効率向上が達成できるかを確認する.

この実験では, 検証する 1 次元 QFT サンプラーの設定として $N = 10$ かつ $M = 4$ と設定した (これは前述の通り Quantum-inspired アルゴリズムにて, 古典コンピュータを用いて高速に計算可能である). よって, サンプリングの対象となる分布 $p(x)$ は $x = 0$ から $1023 (= 2^{10} - 1)$ までの整数を状態空間として持ち, 提案分布は 2^4 個の複素パラメータを持つことになる.

サンプリングの対象となる分布 $p(x)$ としては、図 8.3 の (a)-(e) の上部枠内にプロットされている赤破線の分布に示す 5 つの異なる分布を用いた。それぞれ、分布形状はおおよそ (a) 標準偏差が大きな正規分布様の分布、(b) 標準偏差が小さな正規分布様の分布、(c) 3 つの大小の山がある分布、(d) 矩形波様の分布、(e) 2 つの大小の山がある分布、となっている。これらの分布に対して、それぞれ別個に QFT サンプラーを適用し、学習とサンプリングを実行した。

学習とサンプリングの繰り返し回数は 40000 とし、1 回あたり $B = 32$ のサンプル（候補状態）を生成した。生成されたサンプルは MH 法の採択・棄却と損失関数の勾配の計算に用いた。求めた勾配は、勾配法的一种である Momentum 法 [99] によってパラメータ更新に使用した。ただし、Momentum 法とは、パラメータを

$$\theta \leftarrow \theta - \alpha m', m \leftarrow \mu m + (1 - \mu) \left(\frac{\partial}{\partial \theta} L(\theta) \right)^* \quad (8.38)$$

と更新するアルゴリズムである。ここで α, μ は学習アルゴリズムのハイパーパラメータであり、 $\alpha = 0.01, \mu = 0.9$ と設定した。また、 m は勾配の移動平均に関する変数である。

学習の過程にともなう提案分布の変化をを図 8.3 の (a)-(e) の上部枠内に黒実線で示しており、また、最終的に得られた提案分布は特別に青実線で示した。最終的な提案分布（青実線）と目的の分布（赤破線）との定量的な差を示すために、各図下部に目標分布 $p(x)$ と提案分布 $q_{QFT}(x)$ のクロスエントロピー誤差（損失関数）、同 Wasserstein 距離、および MH 法における平均採択率を示している。Wasserstein 距離は、損失関数とは別の視点から分布の一致性を見るための指標として導入しており、クロスエントロピー誤差とは異なり完全な分布一致で距離が 0 となる*2。

提案分布と目的の分布（赤破線）の形状を比較すると、最終的に得られた提案分布（青実線）は (a)-(c) および (e) において非常によく一致していることが分かる。また、(d) については大まかにはよく一致しているが、高周波振動を持つ誤差が生じていることが分かる。これは、1 次元 QFT サンプラーを古典コンピュータ上で実行するにあたって、分布の低周波成分のみをパラメータ化したことにより、矩形波のフーリエ級数のうち高周波成分が無視されたことによると考えられる。また、矩形波は不連続点を持つから、ギブスの現象による効果も含むと考えられる。しかしながら、定量的には採択率については (a)-(e) すべてともに 0.9 以上を達成している。本研究では前述の通り、提案分布として直前の分布に依存しないもの（グローバルアップデート）を考えているため、前後の採択サンプル間の相関はほぼ無いに等しく、平均採択率がそのまま効率を表している。以上より、1 次

*2 クロスエントロピー誤差は分布が一致していても自己エントロピーの分だけ残る。

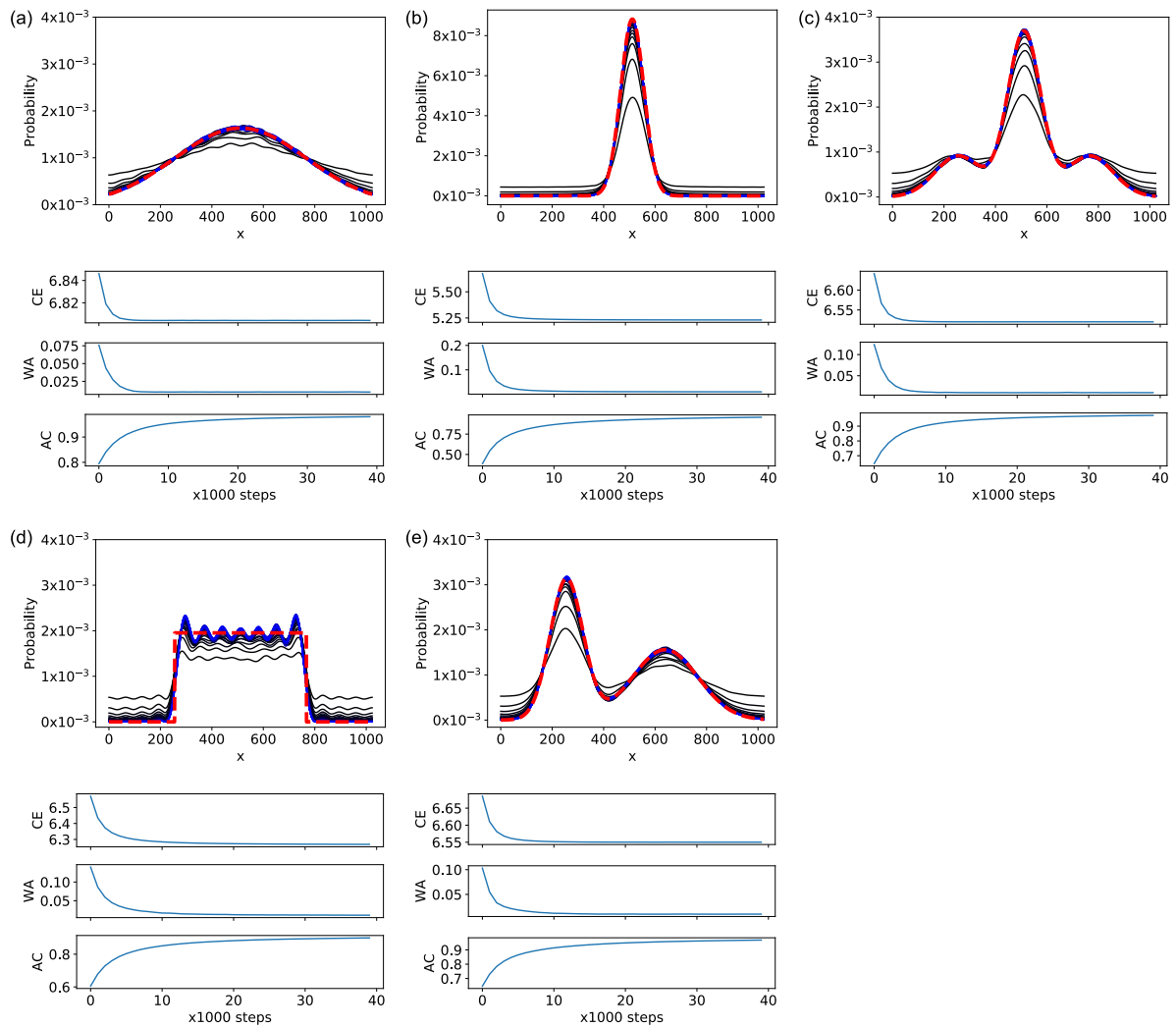


図 8.3 1次元分布に対する1次元QFTサンプラーの学習結果とサンプリング効率. ([97]より引用) 各図(a)-(e)はそれぞれ異なる分布に対して1次元QFTサンプラーを適用した結果を示している. 各図上部は目的とする分布(赤破線), 提案分布の学習経過(黒実線), 最終的に獲得した提案分布(青線)を示している. 各図下部は上から順に学習に伴うクロスエントロピー誤差関数(損失関数)の変化, Wasserstein距離の変化, 平均採択率の変化, それぞれを示す.

元QFTサンプラーによって, 様々な1次元分布のサンプリング効率改善を達成したことを示した.

8.6.2 多次元 QFT サンプラーの実験

QFT サンプラーの性能評価のため、2次元分布に対して多次元 QFT サンプラーを適用することにより、適切な提案分布の獲得と効率向上が達成できるかを確認する。

この実験では、2次元分布を対象とするため、2段の1次元 QFT サンプラーからなる次の提案分布

$$q(\mathbf{x}) = q_{QFT}(x_2, \mathbf{f}_2(x_1; \boldsymbol{\theta}_2))q_{QFT}(x_1, \mathbf{f}_1(\boldsymbol{\theta}_1)) \quad (8.39)$$

を用いる。ここで、 \mathbf{f}_1 については1次元 QFT サンプラーと同様のため Identity モデルを使用し、 \mathbf{f}_2 については1次元目のサンプル x_1 に依存して2次元目 x_2 を決定する必要があるため、Identity, LBLR, NBLR, NN の4つの方法をそれぞれ個別に検証する。また、NBLR で使用する非線形特徴量としては、次の4次元特徴量

$$\phi_1(x_1) = \hat{x}, \phi_2(x_1) = \hat{x}^2, \phi_3(x_1) = \hat{x}^3, \phi_4(x_1) = \sqrt{\|\hat{x}\|} \quad (8.40)$$

を用いている。ただし $\hat{x} = 2x_1/(2^N - 1) - 1$ あるとする。これら4つの方法は、左から順に計算量が大きくなっていくため、高い表現力を持つ NN だけでなく、それ以外の手法で高い効率のサンプリングを達成できるかどうかを確認する。

多段階 QFT サンプラーを構成する1次元 QFT サンプラーの設定としては、1次元分布の実験と同様の設定 ($N = 10$ かつ $M = 4$) とした。よって、サンプリングの対象となる分布 $p(\mathbf{x})$ は0から1023 ($= 2^{10} - 1$) までの2つの整数の組 ($\mathbf{x} = (x_1, x_2)$) を状態空間として持つことになる。サンプリングの対象となる分布 $p(\mathbf{x})$ として、図 8.4 の (a)-(g) の上部一番右にプロットされている分布に示す7つの異なる分布を用いた。それぞれの分布の形状は、(a) 円周上に台がある分布、(b) 相関のある正規分布、(c) 相関のない正規分布、(d) LBLR で表現することが可能な分布、(e) 相関のある2つの山をもつ混合正規分布、(f) 市松模様状の分布、(g) 顔のような形の分布、となっている。これらの分布に対してそれぞれ別個に多段階 QFT サンプラーが適用され、それぞれ学習とサンプリングを並行に実行した。学習とサンプリングの繰り返し回数は400,000とし、1回あたり $B = 32$ のサンプル (候補状態) を生成した。生成されたサンプルは MH 法の採択・棄却と損失関数の勾配の計算に用いた。求めた勾配は、Momentum 法 ($\alpha = 0.01, \mu = 0.9$) によってパラメータを更新した。

Identity, LBLR, NBLR, NN の各手法によって最終的に得られた提案分布を図 8.4(a)-(g) 上部左から順にそれぞれ示している。また、各手法による学習の過程での目標分布 $p(x)$ と提案分布 $q_{QFT}(x)$ のクロスエントロピー誤差 (損失関数)、同 Wasserstein 距離、

および MH 法における平均採択率を図 8.4(a)-(g) 下部に示している。図 8.4 から、それぞれの結果は、次に述べる通りとなった。まず、Identity 法については、各次元で独立な分布しか表現することができないため、無相関な分布である (c) のみ目標分布と一致する提案分布を獲得できているものの、その他 6 つの分布については相関のない提案分布が獲得されてしまっている。この影響は特に (f) の分布に顕著に現れており、初期の一様分布から一切学習が進行していない様子を観察できる。次に、各次元の相関を扱うための最も簡単なモデルである LBLR 法では、(b)・(d)・(e) などの分布は良く表現できているが、(a)・(f)・(g) などのより複雑であると考えられる分布については正しく表現できていない。一番表現力の高い NN 法については、(a)-(g) いずれの分布に対しても良く一致した提案分布を獲得しており、採択率もすべて 0.8 を上回っていることが分かる。次点として NBLR 法が挙げられ、NN 法より多少採択率は低下するがほとんど遜色ない効率性を示している。このことから、計算量の大きい NN 法と比べてかなり小さな計算量しか持たない NBLR 法を用いることで、効果的に効率のよいサンプリングを達成できることがわかる。以上の結果から、様々な 2 次元分布に対して、計算量の小さい NBLR 法を用いることで十分高い効率でサンプリングが実行できることが確認された。

8.7 結論と今後の展望

本章で行った研究について、結論と今後の展望を述べる。

8.7.1 結論

本章では MC シミュレーションのために量子コンピュータを用いてサンプリングを行う手法について検討し、以下の通り報告した。

1. 量子コンピュータを用いた汎用的な自己学習サンプリングアルゴリズムの定式化を行った。
2. その中でも、QFT 回路を用いることで Quantum-inspired なアルゴリズムを構成し、古典コンピュータにおける新たなサンプリングアルゴリズムを示した。
3. その結果、QFT サンプラーを用いて様々な分布のサンプリング高速化が達成できることを示した。

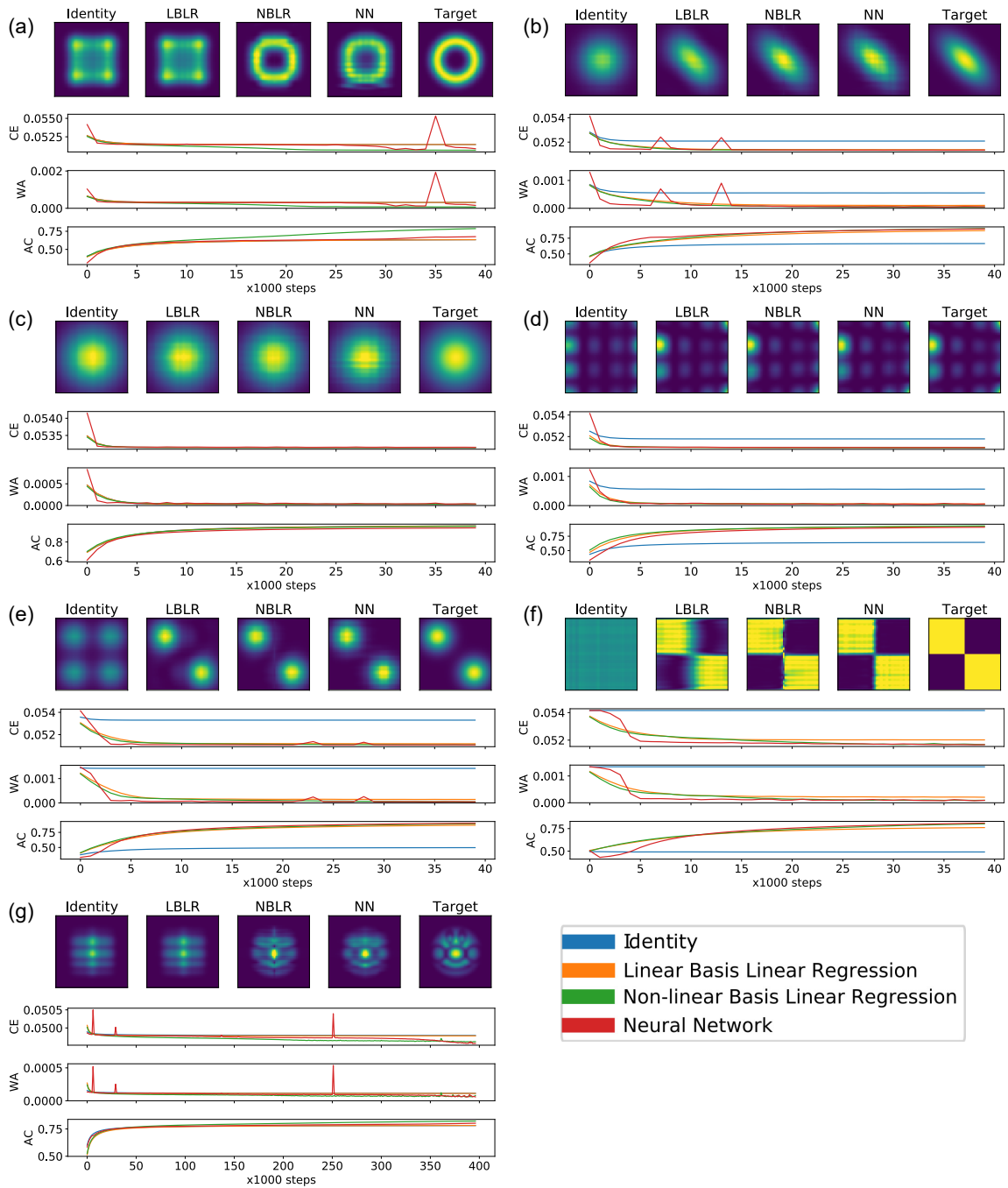


図 8.4 2次元分布に対する多次元次元 QFT サンプラーの学習結果とサンプリング効率． ([97] より引用) 各図 (a)-(g) はそれぞれ異なる 2次元分布に対して多次元 QFT サンプラーを適用した結果を示している． 各図上部は左から順に Identity 法・LBLR 法・NBLR 法・NN 法によって獲得した最終的な提案分布，目的とする 2次元分布，を示している． 各図下部は上から順に学習に伴うクロスエントロピー誤差関数（損失関数）の変化，Wasserstein 距離の変化，平均採択率の変化，を各手法についてそれぞれ示す．

8.7.2 今後の展望

本研究では QFT 回路を提案分布に用いることで量子コンピュータによるサンプリングアルゴリズムを構成した。しかしながら、提案分布に求められる 3 つの性質を満たす、より表現力の高い回路が存在すると考えられ、そのようなアルゴリズムの研究を進めている。

また、本研究では多次元分布のサンプリングのために 1 次元の QFT 回路を多段で組み合わせることによりサンプリングを行ったが、この方式であると高次元分布に対する自己学習が難しくなると考えられる。今後、量子コンピュータの性能が飛躍的に高まると予想されるため、これらの性能を効率的に活用できるサンプリングアルゴリズムの開発が期待される。

第9章

結論

本論文では、機械学習および量子コンピュータを応用することによって分子シミュレーションを用いた研究開発を高速化・効率化する方法について述べた。

分子シミュレーションは、MD シミュレーションと MC シミュレーションに大別され、MD シミュレーションは分子の時間発展を計算し、MC シミュレーションは統計力学に基づく確率分布をサンプリングする。分子シミュレーションは既に材料開発・創薬を含む広範な分野で活用されているものの、様々な課題が残っている。本研究では特に重要と思われる次の課題

- (課題 1) MD シミュレーションを用いる場合、時間発展は逐次計算であるから並列化が難しく、長時間にわたるシミュレーションを実行することが困難であること。
- (課題 2) MD シミュレーション実行結果から、有意義な知見を取り出すためには、専門家の高度な知識に基づく解析が必要となる難易度の高いものであること。
- (課題 3) MC シミュレーションを用いる場合、ボルツマン分布のような複雑な構造を持つ確率分布を効率的にサンプリングすることが困難であること

を取り上げ、機械学習および量子コンピュータの活用による課題解決を目指した。

第5章では、深層生成モデルによる部分系の確率的時間発展を実行する代理モデルを導入し、短時間の MD シミュレーションから長時間の MD シミュレーションを予測することによって課題 1 の解決を試みた。代理モデルの導入にあたって、長時間予測による誤差を可能な限り低減するための手法を提案した。その結果、ポリエチレン溶融体の平均二乗変位の計算を 15 倍高速化することに成功した。第6章では、複数の系の MD シミュレーション結果を深層学習により自動解析する手法を提案することによって、課題 2 の解決を試みた。その結果、部分系のダイナミクスを統計集団とみなして、各統計集団間の距離

の埋め込みと、距離の大小に貢献する個別のダイナミクスを半自動的に抽出する機械学習手法を開発した。この手法を用いることによってアミノ酸水溶液系において有用な解析を半自動で行うことに成功した。第7章では、深層生成モデルによって、複雑な構造をもつボルツマン分布に適した形へ提案分布を変形することによって、課題3の解決を試みた。提案分布を構成するにあたって、原子同士が近接する配置を取り除くための解析的な変形手法や、自己学習により適切な変形方法を自動的に獲得する手法を提案した。その結果、LJ粒子系において、カノニカルアンサンブルシミュレーションではステップあたり最大20倍の粒子移動量を達成し、グランドカノニカルアンサンブルシミュレーションでは最大30倍の速度で粒子数に関する独立なサンプルを得ることに成功した。第8章では、量子コンピュータによる情報処理に基づき、多様な分布を高速にサンプリング可能な方法を構築することによって、課題3の解決を試みた。量子コンピュータによるサンプリングアルゴリズムの構成にあたって、自己学習によりサンプリングを高速化する枠組みの整理を行い、かつQFT回路を用いて具体的かつQuantum-inspiredなアルゴリズムを構成した。その結果、様々な分布のサンプリング高速化に成功した。

今後の展望としては、次のような発展が考えられる。課題1においては、部分系の短時間シミュレーションから長期予測が可能であることを示した。しかしながら、どのように部分系を選べばよいかや、短時間シミュレーションの長さの必要量など、未解明な部分が多い。特に、部分系の選び方については、部分系として開放系、すなわち部分系として考える領域が時間発展によって分子が流出・流入する場合、を考慮することがさらなる効率向上に重要だと考える。そのような流出・流入を考慮する予測を行うためには、粒子の生成消滅を取り扱う新たな機械学習モデルを提案する必要がある。また、課題2においては、MDシミュレーション結果のダイナミクス自動解析手法を提案したが、その応用例は不十分である。本研究では、アミノ酸水溶液系および共著研究としてリガンド-タンパク質系の解析を取り上げたが、それ以外にも高分子のガラス転位や水分子の溶解現象など、ダイナミクスが重要なファクターとなると考えられている物質が多数知られており、本研究の提案手法による解析を進めていきたい。課題3においては、機械学習と量子コンピュータの双方からモンテカルロシミュレーションの効率向上について議論した。機械学習においては、ボルツマン分布をなるべく一様分布などの無相関分布に近づくように変換することが本質的に重要であり、分布変換の性能と変換の計算効率の両方をさらに向上させる機械学習アルゴリズムの開発が求められる。また、量子コンピュータにおいては、普遍的な確率分布に対して、量子超越性をもつサンプリングアルゴリズムの存在自体が未だ不明であり、さらなる研究が必要である。

謝辞

本研究の遂行および学位論文の執筆あたり、多くの方から多大なご支援・ご協力をいただきました。この場を借りて感謝の意を述べさせていただきます。

本研究は、著者が慶應義塾大学理工学部および同大学大学院理工学研究科前期・後期博士課程在学中に、泰岡顕治教授の指導のもと行いました。泰岡顕治教授には、学部の研究室配属時から著者の能力を最大限に引き出す研究指導をいただいたのみならず、研究生生活や将来の事などを含め様々な人生相談にも快く乗っていただきました。これからも多々ご迷惑をおかけすることがあるかと存じますが、引き続き宜しくお願い致します。

そして、学位審査にあたりお忙しい中副査を引き受けてくださりました矢向高弘准教授、山本直樹教授、深瀧康二教授に感謝いたします。矢向高弘准教授には、著者が運営委員として参加する AI・高度プログラミングコンソーシアムの立ち上げ時期から仕事をさせていただいております。学生教育やマネジメント手法などのご指導もいただきました。山本直樹教授には、著者が採択された IPA 未踏ターゲット事業（ゲート式量子コンピュータ部門）プロジェクトの担当 PM としてはじめてご指導をいただいてから、量子コンピュータに関する様々な議論をさせていただきました。深瀧康二教授は、著者が所属する専修の教員であり、学部生の頃から恐れ多くも何時でも気軽に疑問に答えて下さりました。

研究室にて丁寧かつ大変にご指導をいただいた、湯原大輔博士、友部勝文博士、野澤拓磨博士、Paul Brumby 特任助教にお礼申し上げます。湯原大輔博士には、これまでの研究生生活の大半においてお側で支えて下さり、研究テーマの精密化、研究ストーリーの構築方法、結果の解析方法の議論、論文執筆の方法など、地道な研究活動を具体的かつ的確にご指導いただきました。友部勝文博士には、研究室配属初年度、研究の進め方が良くわからない中で、はじめての論文をまとめる際に多大なご協力をいただきました。また、はじめての国際学会で困惑する中、先頭に立って助けてくださいましたことを覚えております。野澤拓磨博士には、親身に研究相談に乗っていただいたのみならず、休憩時の雑談を通して常に興味深いお話をお聞きすることができました。Paul Brumby 特任助教には、

著者の拙い英文の修正を常に快く修正していただきましたし、英文原稿の執筆における様々なアドバイスをいただきました。

本研究において様々なご協力をいただいた、泰岡研究室卒業生及び関係者の皆様に感謝致します。村松眞由専任講師には、機械学習を材料分野へ適用する研究チームに招待していただき、研究室の学生の面白い研究テーマを共に進めさせていただくことができました。また、様々な困り事の相談も引き受けて下さり、楽しい研究生活を送ることができました。荒井規允准教授には、本研究にて提案した手法を新たな系へ適用する共同研究をさせていただきました。東京大学金子敏宏特任研究員には、本研究を進めていく上での課題点をアドバイスいただきました。坂牧隆司博士には、本研究の実用性について様々な議論をさせていただきました。高橋和義博士には、高分子の専門家としての観点から様々なアドバイスをいただきました。野村昴太郎博士には、GPU やスーパーコンピュータにおける計算高速化の手法などを詳しく教えていただきました。Donguk Suh 博士には、本研究を詳しく理解していただき、また、研究室を楽しく過ごす方法を教わりました。工学院大学平塚将起准教授には、第一原理計算や水の振動に関する知見を惜しみなく教えていただきました。山本詠士専任講師には、研究をご一緒させていただく中、為になる厳しいご意見をいただくことができました。Stephen Fitz 特任助教には、深層学習の数学的側面について教えていただきました。平野秀典特任助教には、タンパク質シミュレーションに関する知見を教えていただきました。

各々の研究において親身にご指導をいただいた先生方に、感謝致します。神戸大学牧野淳一郎教授には、理化学研究所インターンにて受け入れていただいていたから、スーパーコンピュータの研究をさせていただいております。その中にご指導いただいた知見は本研究を遂行する上で欠くことのできないものでした。大阪大学藤井啓祐教授には、量子コンピュータの研究を進める上での貴重なアドバイスを多々いただきました。

研究室生活を共にした研究室の先輩・後輩の皆様にも、大変お世話になりました。森田一軌博士、山光隆一さん、今井尚子さん、阿由葉翔さん、入江将大さん、九十九慶之 Kevin さん、長谷川智大さん、中村真二郎さん、井上貫太郎さん、小林祐生さん、鈴木雄大さんには、先輩として研究室の生活も含めて様々なアドバイスをいただきました。ありがとうございます。

澤眞詩さん、加藤修三さん、平川和明さん、平山健太さん、松井佳菜子さん、山田悠斗さん、プアカイさん、石山将成さん、板倉慶直さん、小和口昌愛さん、谷口鷹さん、山田有紗さん、Araf al Rafi さん、Jean-Francois Cailleau さん、佐竹寛さん、関紘也さん、平出和也さん、辻之上弘晃さん、阿部晃基さん、川田稜さん、佐藤大稀さん、三輪玄輝さん、安田一希さん、児嶋佑太さん、佐々木翔唯さん、山寄祐輔さん、石合智貴さん、栗林

直信さん、佐竹晃典さん、菖蒲健太さん、金澤祐里奈さん、本田理央さん、吉村厚輝さんとは、楽しい研究生生活を送らせていただきました。

滝口啓介さんは、本論文の校正を手伝ってもらった、信頼できる友人です。中村太一さんは、共同研究者として研究を行った、頼りになる友人です。研究生生活を送るにあたって、同期であり大切な友人でもある岡田清志郎さん、小野祐為さん、荻野健太さんとの取り留めも無い会話が研究を続ける精神的な助けになりました。最後に、長い大学院生活を支えてくださった家族に感謝致します。ありがとうございました。

2022年2月

参考文献

- [1] Richard H Gee, Naida Lacevic, and Laurence E Fried. Atomistic simulations of spinodal phase separation preceding polymer crystallization. *Nature materials*, Vol. 5, No. 1, pp. 39–43, 2006.
- [2] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, Vol. 334, No. 6055, pp. 517–520, 2011.
- [3] Matthew R Walsh, Carolyn A Koh, E Dendy Sloan, Amadeu K Sum, and David T Wu. Microsecond simulations of spontaneous methane hydrate nucleation and growth. *Science*, Vol. 326, No. 5956, pp. 1095–1098, 2009.
- [4] Kenji Yasuoka and Mitsuhiro Matsumoto. Molecular dynamics simulation of homogeneous nucleation in supersaturated water vapor. *Fluid phase equilibria*, Vol. 144, No. 1-2, pp. 369–376, 1998.
- [5] S B Legoas, V R Coluci, S F Braga, P Z Coura, S O Dantas, and Douglas S Galvao. Molecular-dynamics simulations of carbon nanotubes as gigahertz oscillators. *Physical review letters*, Vol. 90, No. 5, p. 055504, 2003.
- [6] Qingwei Gao, Yudan Zhu, Yang Ruan, Yumeng Zhang, Wei Zhu, Xiaohua Lu, and Linghong Lu. Effect of adsorbed alcohol layers on the behavior of water molecules confined in a graphene nanoslit: a molecular dynamics study. *Langmuir*, Vol. 33, No. 42, pp. 11467–11474, 2017.
- [7] Yasushi Shibuta, Shinji Sakane, Eisuke Miyoshi, Shin Okita, Tomohiro Takaki, and Munekazu Ohno. Heterogeneity in homogeneous nucleation from billion-atom molecular dynamics simulation of solidification of pure metal. *Nature communications*, Vol. 8, No. 1, pp. 1–9, 2017.
- [8] Sarah R Needham, Selene K Roberts, Anton Arkhipov, Venkatesh P Mysore, Christopher J Tynan, Laura C Zanetti-Domingues, Eric T Kim, Valeria Losasso,

- Dimitrios Korovesis, Michael Hirsch, Daniel J Rolfe, David T Clarke, Martyn D Winn, Alireza Lajevardipour, Andrew H A Clayton, Linda J Pike, Michela Perani, Peter J Parker, Yibing Shan, David E Shaw, and Marisa L Martin-Fernandez. Egfr oligomerization organizes kinase-active dimers into competent signalling platforms. *Nature communications*, Vol. 7, No. 1, pp. 1–14, 2016.
- [9] Takeshi Aoyagi, Fumio Sawa, Tatsuya Shoji, Hiroo Fukunaga, Jun-ichi Takimoto, and Masao Doi. A general-purpose coarse-grained molecular dynamics program. *Computer physics communications*, Vol. 145, No. 2, pp. 267–279, 2002.
- [10] Soumil Y Joshi and Sanket A Deshmukh. A review of advancements in coarse-grained molecular dynamics simulations. *Molecular simulation*, Vol. 47, No. 10–11, pp. 786–803, 2021.
- [11] Tetsu Narumi, Yousuke Ohno, Noriaki Okimoto, Takahiro Koishi, Atsushi Suenaga, Noriyuki Futatsugi, Ryoko Yanai, Ryutaro Himeno, Shigenori Fujikawa, Makoto Taiji, and Mitsuru Ikei. A 55 tflops simulation of amyloid-forming peptides from yeast prion sup35 with the special-purpose computer system mdgrape-3. In *Proceedings of the 2006 ACM/IEEE conference on supercomputing*, pp. 49–es, 2006.
- [12] Sho Ayuba, Donguk Suh, Kentaro Nomura, Toshikazu Ebisuzaki, and Kenji Yasuoka. Kinetic analysis of homogeneous droplet nucleation using large-scale molecular dynamics simulations. *The journal of chemical physics*, Vol. 149, No. 4, p. 044504, 2018.
- [13] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, Vol. 98, No. 14, p. 146401, 2007.
- [14] Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, Vol. 9, No. 1, pp. 1–10, 2018.
- [15] Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the kohn-sham equations with machine learning. *Nature communications*, Vol. 8, No. 1, pp. 1–10, 2017.
- [16] Christoph Wehmeyer and Frank Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The journal of chemical physics*, Vol. 148, No. 24, p. 241703, 2018.

- [17] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, Vol. 9, No. 1, pp. 1–11, 2018.
- [18] João M L Ribeiro, Pablo Bravo, Yihang Wang, and Pratyush Tiwary. Reweighted autoencoded variational bayes for enhanced sampling (rave). *The journal of chemical physics*, Vol. 149, No. 7, p. 072301, 2018.
- [19] Thomas Stecher, Noam Bernstein, and Gábor Csányi. Free energy surface reconstruction from umbrella samples using gaussian process regression. *Journal of chemical theory and computation*, Vol. 10, No. 9, pp. 4079–4097, 2014.
- [20] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, Vol. 365, No. 6457, 2019.
- [21] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, Vol. 2, p. 79, 2018.
- [22] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj quantum information*, Vol. 6, No. 1, pp. 1–9, 2020.
- [23] Panagiotis K Barkoutsos, Jerome F Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J Egger, Matthias Troyer, Antonio Mezzacapo, et al. Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. *Physical review A*, Vol. 98, No. 2, p. 022322, 2018.
- [24] Marco Bentivegna, Nicolò Spagnolo, Chiara Vitelli, Fulvio Flamini, Niko Viggianiello, Ludovico Latmiral, Paolo Mataloni, Daniel J Brod, Ernesto F Galvão, Andrea Crespi, Roberta Ramponi, Roberto Osellameand, and Fabio Sciarrino. Experimental scattershot boson sampling. *Science advances*, Vol. 1, No. 3, p. e1400255, 2015.
- [25] 渡辺澄夫. ベイズ統計の理論と方法. コロナ社, 2012.
- [26] Svetlozar Todorov Rachev, R M. Duality theorems for kantorovich-rubinstein and wasserstein functionals. Warszawa: Instytut Matematyczny Polskiej Akademi Nauk, 1990.
- [27] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

- [28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [29] Ian Goodfellow, Jean Pouget-Abadie, and Mehdi Mirza. Generative adversarial nets, *Advances in neural information processing systems 27*. pp. 2672–2680, 2014.
- [30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [31] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.
- [32] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. *arXiv preprint arXiv:2002.04185*, 2020.
- [33] Naveen Kodali, Jacob Abernethy, James Hays, and Zolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.
- [34] Behnam Neyshabur, Srinadh Bhojanapalli, and Ayan Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv preprint arXiv:1705.07831*, 2017.
- [35] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.
- [36] Ricky T Q Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.
- [37] Daan Frenkel, Berend Smit, and Mark A Ratner. *Understanding molecular simulation: from algorithms to applications*, Vol. 2. Academic press San Diego, 1996.
- [38] Keith W Hastings. Monte carlo sampling methods using markov chains and their applications. Oxford University Press, 1970.
- [39] Jascha Sohl-Dickstein, Mayur Mudigonda, and Michael DeWeese. Hamiltonian monte carlo without detailed balance. In *International conference on machine learning*, pp. 719–726. PMLR, 2014.
- [40] Hidemaro Suwa and Synge Todo. Markov chain monte carlo method without detailed balance. *Physical review letters*, Vol. 105, No. 12, p. 120603, 2010.
- [41] Hiroaki Nishizawa and Hisashi Okumura. Comparison of replica-permutation molecular dynamics simulations with and without detailed balance condition. *Journal of the physical society of japan*, Vol. 84, No. 7, p. 074801, 2015.

- [42] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman J Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of computational physics*, Vol. 23, No. 3, pp. 327–341, 1977.
- [43] Berk Hess, Henk Bekker, Herman J Berendsen, and Johannes G Fraaije. LINCS: a linear constraint solver for molecular simulations. *Journal of computational chemistry*, Vol. 18, No. 12, pp. 1463–1472, 1997.
- [44] John E Lennard-Jones. Cohesion. *Proceedings of the physical society (1926-1948)*, Vol. 43, No. 5, p. 461, 1931.
- [45] Hendrik A Lorentz. Ueber die anwendung des satzes vom virial in der kinetischen theorie der gase. *Annalen der physik*, Vol. 248, No. 1, pp. 127–136, 1881.
- [46] Daniel Berthelot. Sur le mélange des gaz. *Comptes rendus*, Vol. 126, pp. 1703–1706, 1898.
- [47] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The journal of chemical physics*, Vol. 81, No. 1, pp. 511–519, 1984.
- [48] William G Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical review A*, Vol. 31, No. 3, p. 1695, 1985.
- [49] Hans C Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The journal of chemical physics*, Vol. 72, No. 4, pp. 2384–2393, 1980.
- [50] Michele Parrinello and Aneesur Rahman. Crystal structure and pair potentials: A molecular-dynamics study. *Physical review letters*, Vol. 45, No. 14, p. 1196, 1980.
- [51] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d’edp par un schéma en temps pararéel. *Comptes rendus series I mathematics*, Vol. 332, No. 7, pp. 661–668, 2001.
- [52] L’ubor Ladický, SoHyeon Jeong, Barbara Solenthaler, Marc Pollefeys, and Markus Gross. Data-driven fluid simulations using regression forests. *ACM transactions on graphics*, Vol. 34, No. 6, pp. 1–9, 2015.
- [53] Julia Ling, Andrew Kurzwawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of fluid mechanics*, Vol. 807, pp. 155–166, 2016.

- [54] Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E Charon, Gianni De Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS central science*, Vol. 5, No. 5, pp. 755–767, 2019.
- [55] So Takamoto, Chikashi Shinagawa, Daisuke Motoki, Kosuke Nakago, Wenwen Li, Iori Kurata, Taku Watanabe, Yoshihiro Yayama, Hiroki Iriguchi, Yusuke Asano, Tasuku Onodera, Takafumi Ishii, Takao Kudo, Hideki Ono, Ryohto Sawada, Ryuichiro Ishitani, Marc Ong, Taiki Yamaguchi, Toshiki Kataoka, Akihide Hayashi, and Takeshi Ibuka. PFP: Universal neural network potential for material discovery. *arXiv preprint arXiv:2106.14583*, 2021.
- [56] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.
- [57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [58] Roderic Lakes and Roderic S Lakes. *Viscoelastic materials*. Cambridge university press, 2009.
- [59] Erik Lindahl, Berk Hess, and David Van Der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Molecular modeling annual*, Vol. 7, No. 8, pp. 306–317, 2001.
- [60] Katie A Maerzke, Nathan E Schultz, Richard B Ross, and J Ilja Siepmann. Trappe-ua force field for acrylates and monte carlo simulations for their mixtures with alkanes and alcohols. *The journal of physical chemistry B*, Vol. 113, No. 18, pp. 6415–6425, 2009.
- [61] Ryo Kawada, Katsuhiko Endo, Daisuke Yuhara, and Kenji Yasuoka. MD-GAN with multi-particle input: the machine learning of long-time molecular behavior from short-time MD data *arXiv preprint arXiv:2202.00995*, 2022.
- [62] Rossend Rey, Klaus B Møller, and James T Hynes. Hydrogen bond dynamics in water and ultrafast infrared spectroscopy. *The journal of physical chemistry A*, Vol. 106, No. 50, pp. 11993–11996, 2002.
- [63] Biman Bagchi. Water dynamics in the hydration layer around proteins and micelles. *Chemical reviews*, Vol. 105, No. 9, pp. 3197–3219, 2005.
- [64] Damien Laage, Guillaume Stirnemann, Fabio Sterpone, Rossend Rey, and

- James T Hynes. Reorientation and allied dynamics in water and aqueous solutions. *Annual review of physical chemistry*, Vol. 62, pp. 395–416, 2011.
- [65] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. *In Advances in neural information processing systems 20*, Vol. 7, pp. 1433–1440, 2007.
- [66] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density ratio estimation: A comprehensive review (statistical experiment and its related topics). 数理解析研究所講究録, Vol. 1703, pp. 10–31, 2010.
- [67] Gabriel Peyré, Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and trends[®] in machine learning*, Vol. 11, No. 5-6, pp. 355–607, 2019.
- [68] Peter Dunn-Rankin, Gerald A Knezek, Susan R Wallace, and Shuqiang Zhang. *Scaling methods*. Psychology press, 2014.
- [69] Jan De Leeuw and Patrick Mair. Multidimensional scaling using majorization: Smacof in R. *Journal of statistical software*, Vol. 31, pp. 1–30, 2009.
- [70] Paul S Nerenberg, Brian Jo, Clare So, Ajay Tripathy, and Teresa Head-Gordon. Optimizing solute–water van der waals interactions to reproduce solvation free energies. *The journal of physical chemistry B*, Vol. 116, No. 15, pp. 4524–4534, 2012.
- [71] Dail E Chapman, Jonathan K Steck, and Paul S Nerenberg. Optimizing protein–protein van der waals interactions for the amber ff9x/ff12 force field. *Journal of chemical theory and computation*, Vol. 10, No. 1, pp. 273–281, 2014.
- [72] Hans W Horn, William C Swope, Jed W Pitera, Jeffrey D Madura, Thomas J Dick, Greg L Hura, and Teresa Head-Gordon Development of an improved four-site water model for biomolecular simulations: TIP4P-Ew. *Journal of chemical physics*, Vol. 120, pp. 9665–9678, 2004.
- [73] Anselm H C Horn. A consistent force field parameter set for zwitterionic amino acid residues. *Journal of molecular modeling*, Vol. 20, No. 11, pp. 1–14, 2014.
- [74] Ikki Yasuda, Katsuhiko Endo, Eiji Yamamoto, Yoshinori Hirano, and Kenji Yasuoka. Ligand-induced protein dynamics differences correlate with protein-ligand binding affinities: An unsupervised deep learning approach. *arXiv preprint arXiv:2109.01339*, 2021.

- [75] Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of mathematical imaging and vision*, Vol. 62, No. 3, pp. 352–364, 2020.
- [76] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.
- [77] Samuel J Greydanus, Misko Dzumba, and Jason Yosinski. Hamiltonian neural networks. *arXiv preprint arXiv:1906.01563*, 2019.
- [78] E C Neyts, B J Thijsse, M J Mees, K M Bal, and G Pourtois. Establishing uniform acceptance in force biased monte carlo simulations. *Journal of chemical theory and computation*, Vol. 8, No. 6, pp. 1865–1869, 2012.
- [79] M Rao and B J Berne. On the force bias monte carlo simulation of simple liquids. *The Journal of chemical physics*, Vol. 71, No. 1, pp. 129–132, 1979.
- [80] Bin Chen, J Ilja Siepmann, Kwang J Oh, and Michael L Klein. Aggregation-volume-bias monte carlo simulations of vapor-liquid nucleation barriers for lennard-jonesium. *The Journal of chemical physics*, Vol. 115, No. 23, pp. 10903–10913, 2001.
- [81] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, Vol. 314, No. 1-2, pp. 141–151, 1999.
- [82] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, Vol. 195, No. 2, pp. 216–222, 1987.
- [83] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of computational physics*, Vol. 23, No. 2, pp. 187–199, 1977.
- [84] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.
- [85] P P Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annals of physics*, Vol. 369, No. 3, pp. 253–287, Jan 1921.
- [86] Junwei Liu, Huitao Shen, Yang Qi, Zi Y Meng, and Liang Fu. Self-learning monte carlo method and cumulative update in fermion systems. *Physical review B*, Vol. 95, No. 24, p. 241104, 2017.

- [87] Huitao Shen, Junwei Liu, and Liang Fu. Self-learning monte carlo with deep neural networks. *Physical review B*, Vol. 97, No. 20, p. 205140, 2018.
- [88] Yuki Nagai, Huitao Shen, Yang Qi, Junwei Liu, and Liang Fu. Self-learning monte carlo method: Continuous-time algorithm. *Physical review B*, Vol. 96, No. 16, p. 161102, 2017.
- [89] Yuki Nagai, Masahiko Okumura, and Akinori Tanaka. Self-learning monte carlo method with behler-parrinello neural networks. *Physical review B*, Vol. 101, No. 11, p. 115111, 2020.
- [90] Max Tillmann, Borivoje Dakić, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther. Experimental boson sampling. *Nature photonics*, Vol. 7, No. 7, pp. 540–544, 2013.
- [91] Matthew A Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C Ralph, and Andrew G White. Photonic boson sampling in a tunable circuit. *Science*, Vol. 339, No. 6121, pp. 794–798, 2013.
- [92] Michele Mosca and Christof Zalka. Exact quantum fourier transforms and discrete logarithm algorithms. *International journal of quantum information*, Vol. 2, No. 01, pp. 91–100, 2004.
- [93] Juan M Arrazola, Alain Delgado, Bhaskar R Bardhan, and Seth Lloyd. Quantum-inspired algorithms in practice. *arXiv preprint arXiv:1905.10415*, 2019.
- [94] Wu Deng, Hailong Liu, Junjie Xu, Huimin Zhao, and Yingjie Song. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE transactions on instrumentation and measurement*, Vol. 69, No. 10, pp. 7319–7327, 2020.
- [95] Ahmed A A El-Latif, Bassem Abd-El-Atty, Irfan Mehmood, Khan Muhammad, Salvador E Venegas-Andraca, and Jialiang Peng. Quantum-inspired blockchain-based cybersecurity: securing smart edge utilities in iot-based smart cities. *Information processing & management*, Vol. 58, No. 4, p. 102549, 2021.
- [96] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, Vol. 549, No. 7671, pp. 242–246, 2017.
- [97] Katsuhiko Endo, Taichi Nakamura, Keisuke Fujii, and Naoki Yamamoto. Quantum self-learning monte carlo and quantum-inspired fourier transform sampler.

- Physical review research*, Vol. 2, No. 4, p. 043442, 2020.
- [98] Daniel E Browne. Efficient classical simulation of the quantum fourier transform. *New journal of physics*, Vol. 9, No. 5, p. 146, 2007.
- [99] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, Vol. 12, No. 1, pp. 145–151, 1999.