

Kawasaki Quantum Summer Camp 2024

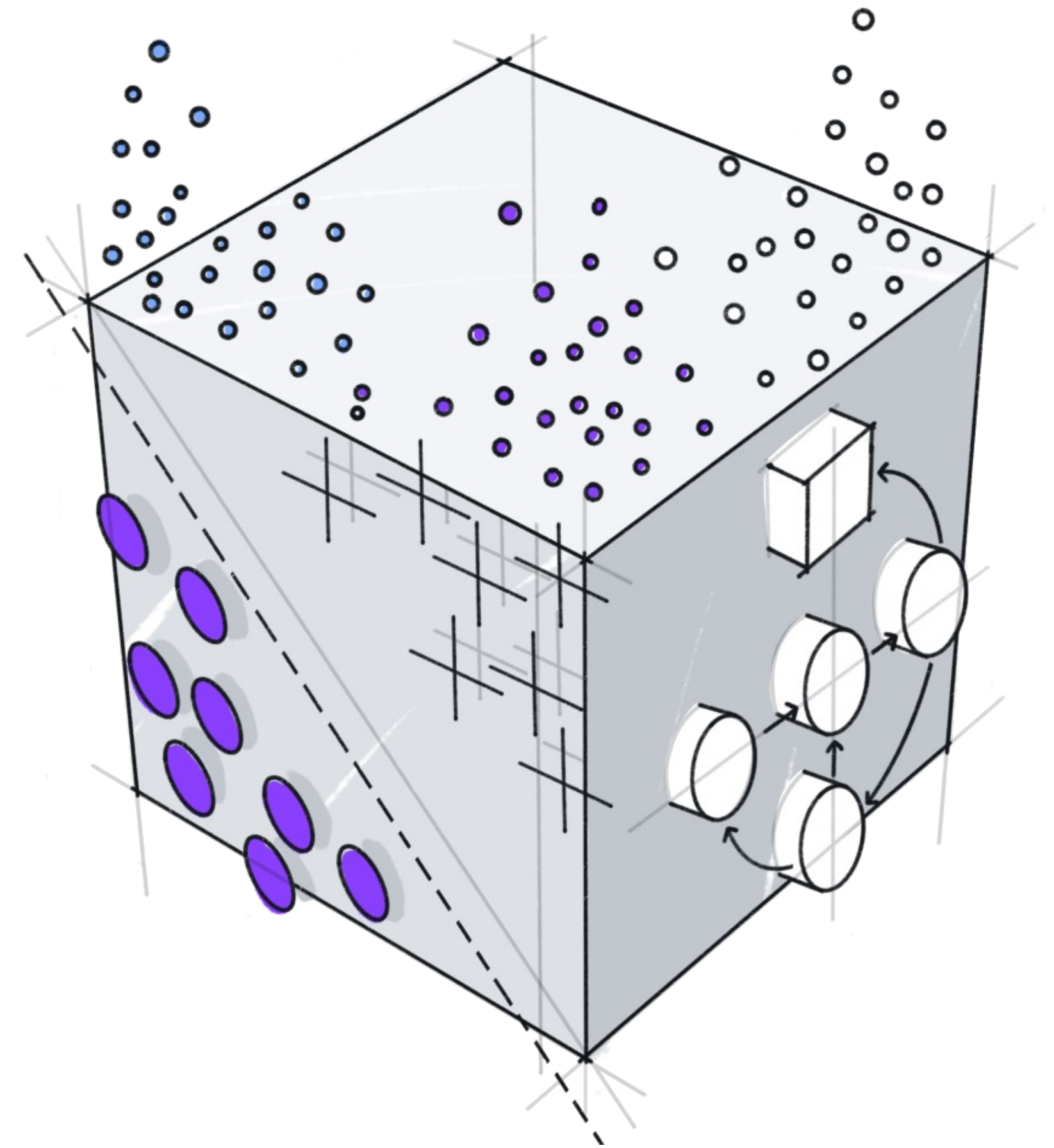
量子機械学習

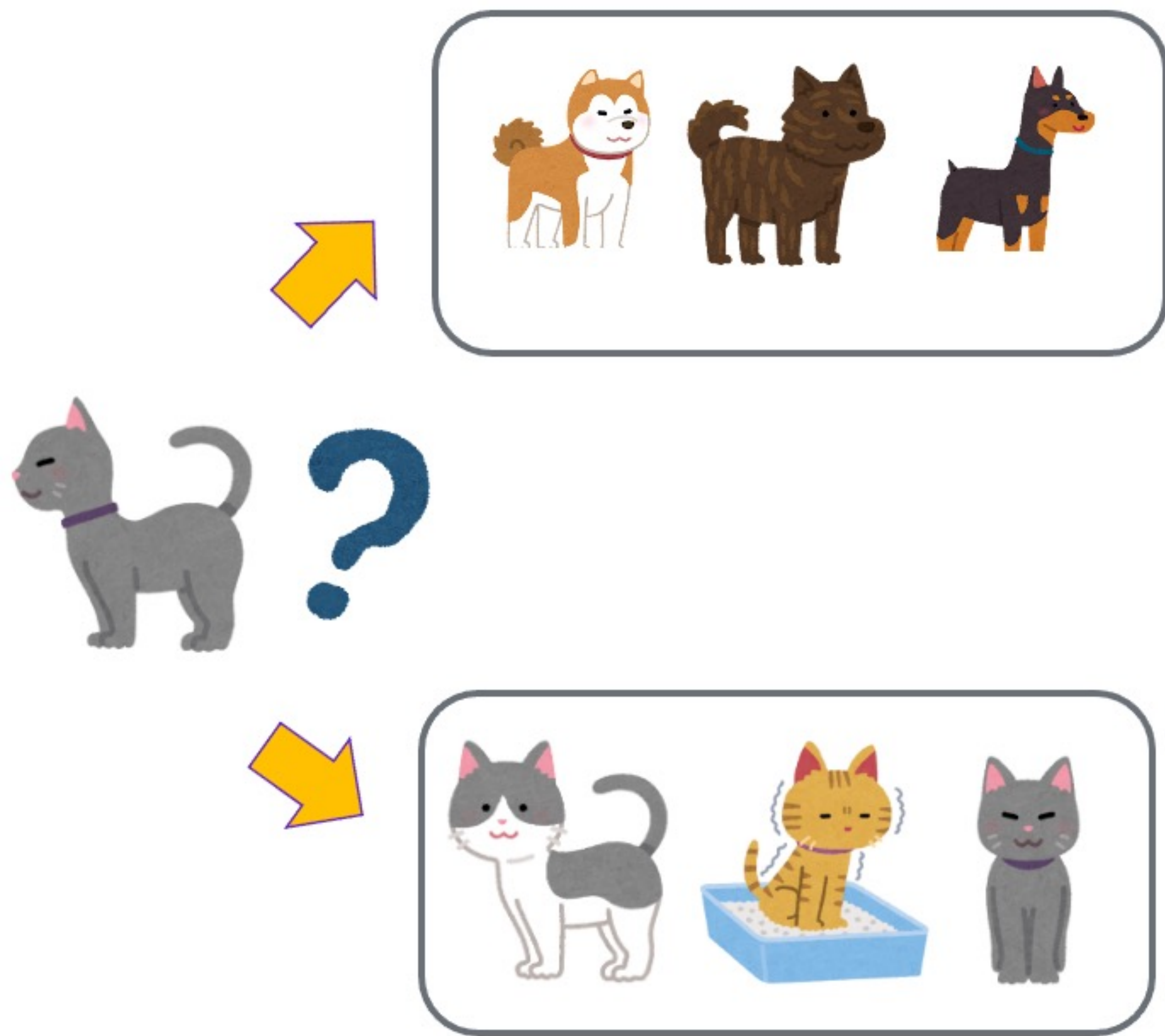
Jul 30, 2024

沼田 祈史

Kifumi Numata

IBM Quantum





NETFLIX

ホーム TV番組・ドラマ 映画 新作&人気作 マイリスト 言語別に検索



mamaさんにイチオシ!



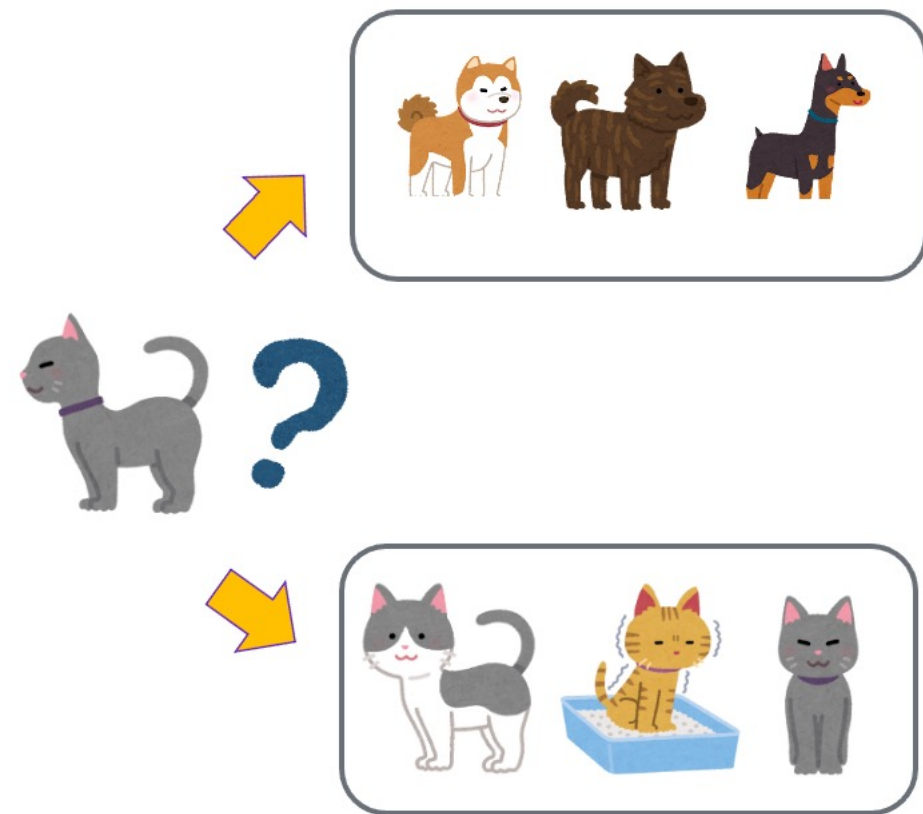
あなたにピッタリのオススメ作品





機械学習

教師あり学習



猫や犬の写真がラベル付けされた集合から、新しい猫や犬の写真を識別する。

教師なし学習



映画の視聴履歴に基づいて視聴者をグループ分けし、新しい映画を推薦する。

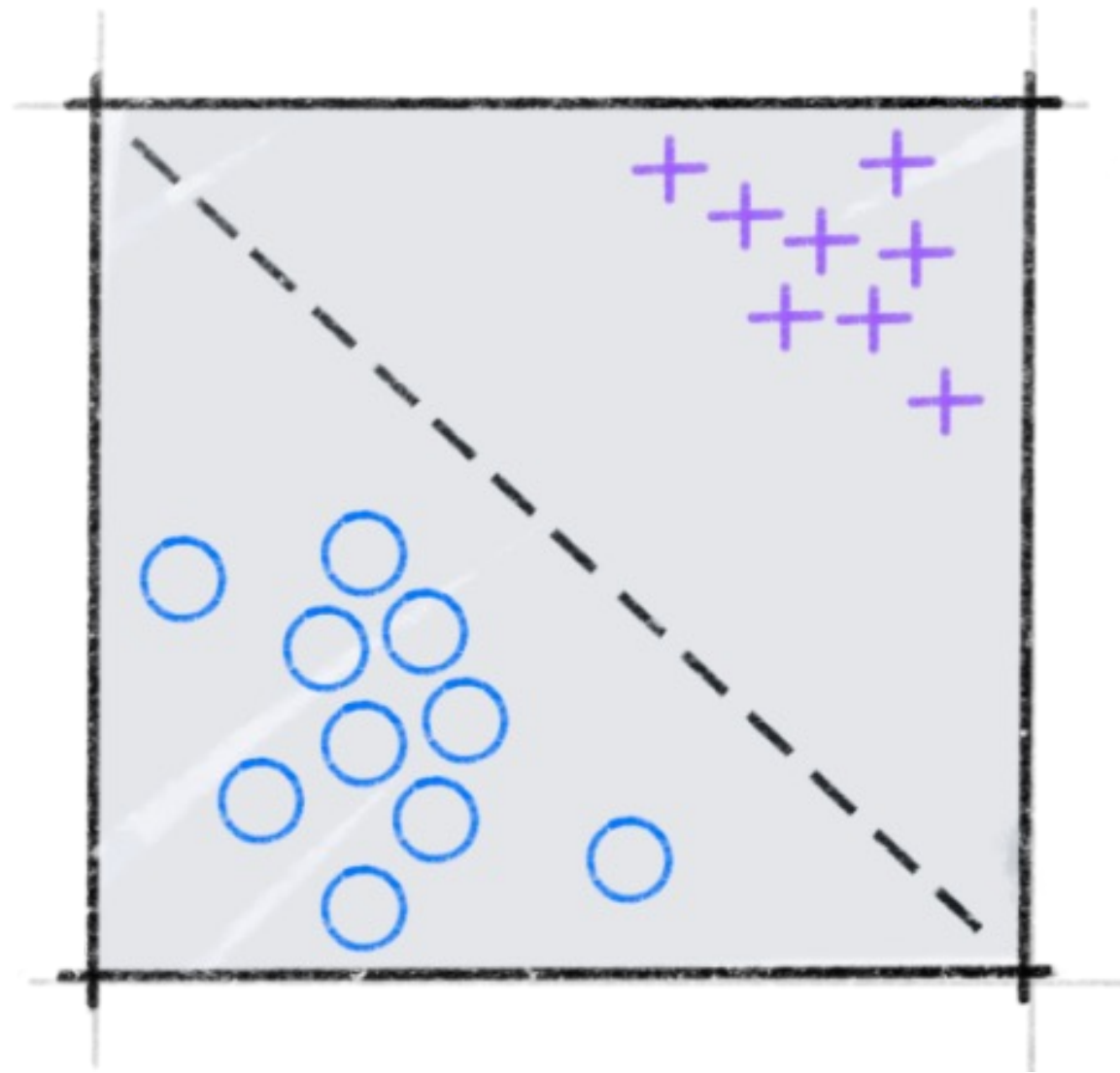
強化学習



囲碁のプレイ方法をアルゴリズムで学習する。

機械学習

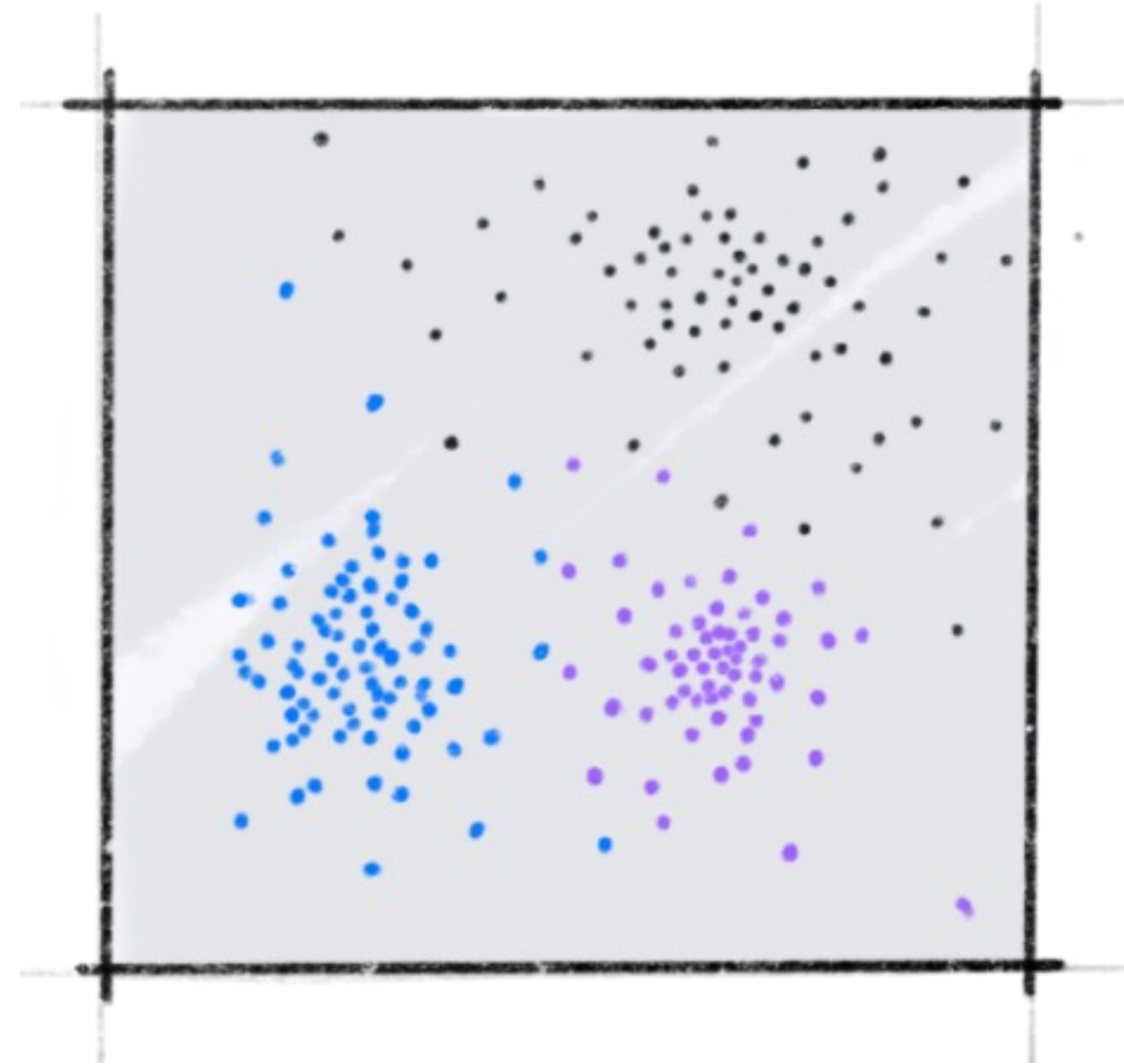
教師あり学習



ラベル付きデータ (x_i, y_i) :
マッピングする関数 $y = f(x)$ を
学習。

猫や犬の写真がラベル付け
された集合から、新しい猫
や犬の写真を識別する。

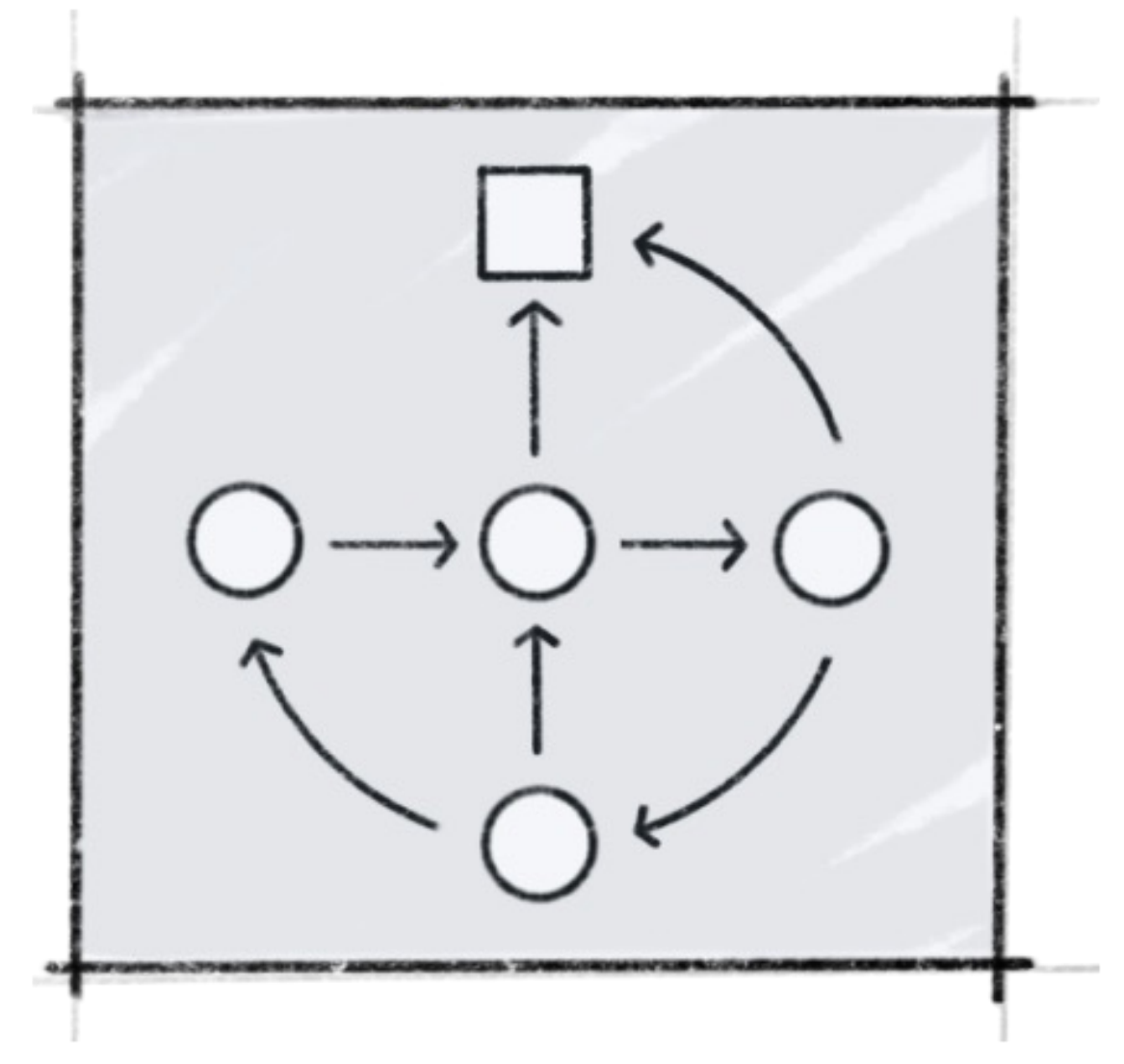
教師なし学習



ラベルのないデータ :
何らかの構造を学習。

映画の視聴履歴に基づいて視聴
者をグループ分けし、新しい映
画を推薦する。

強化学習

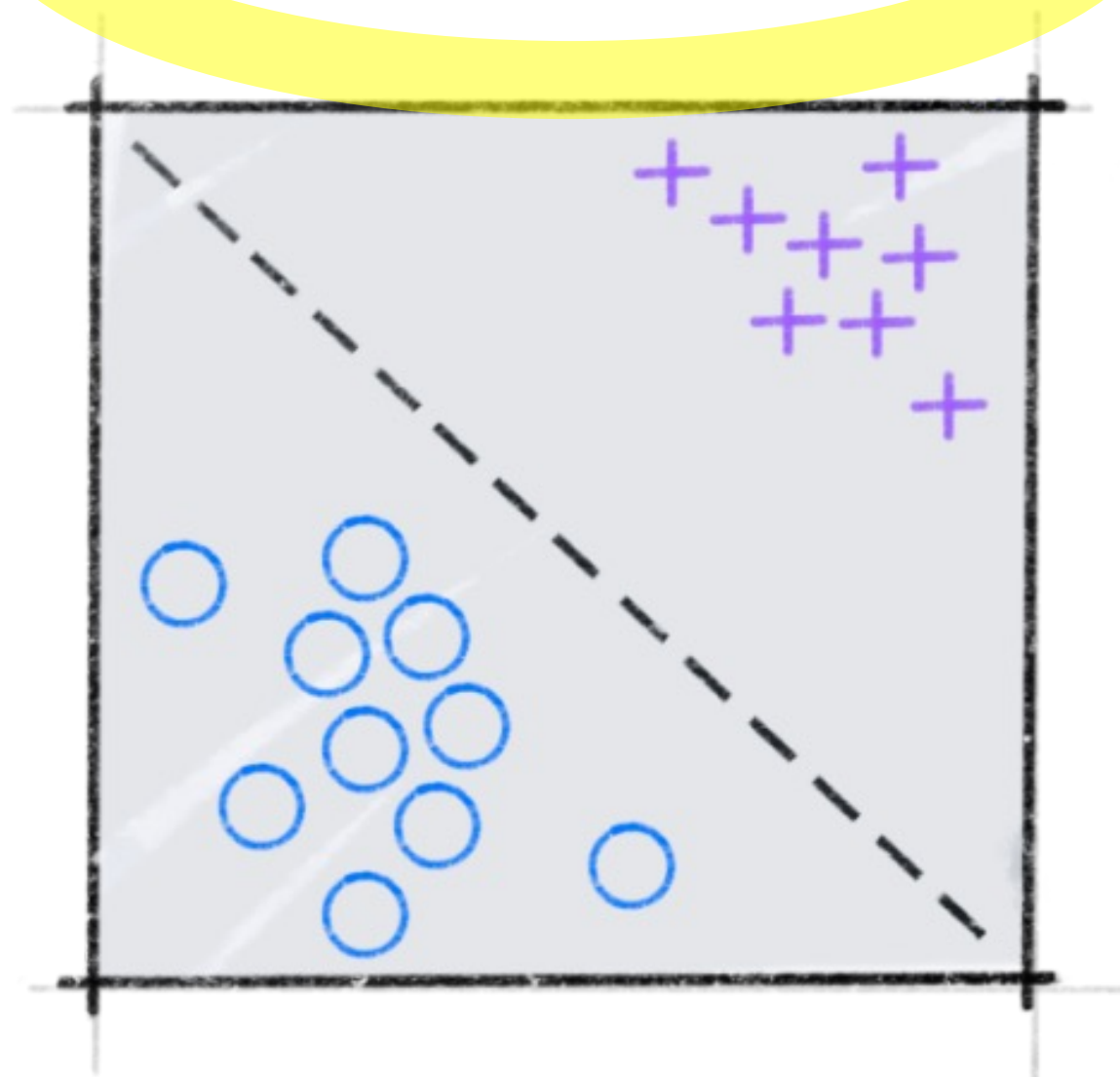


行動に応じて報酬が得られる環
境で、期待される報酬を最大
化。

囲碁のプレイ方法をアルゴリズム
で学習する。

機械学習

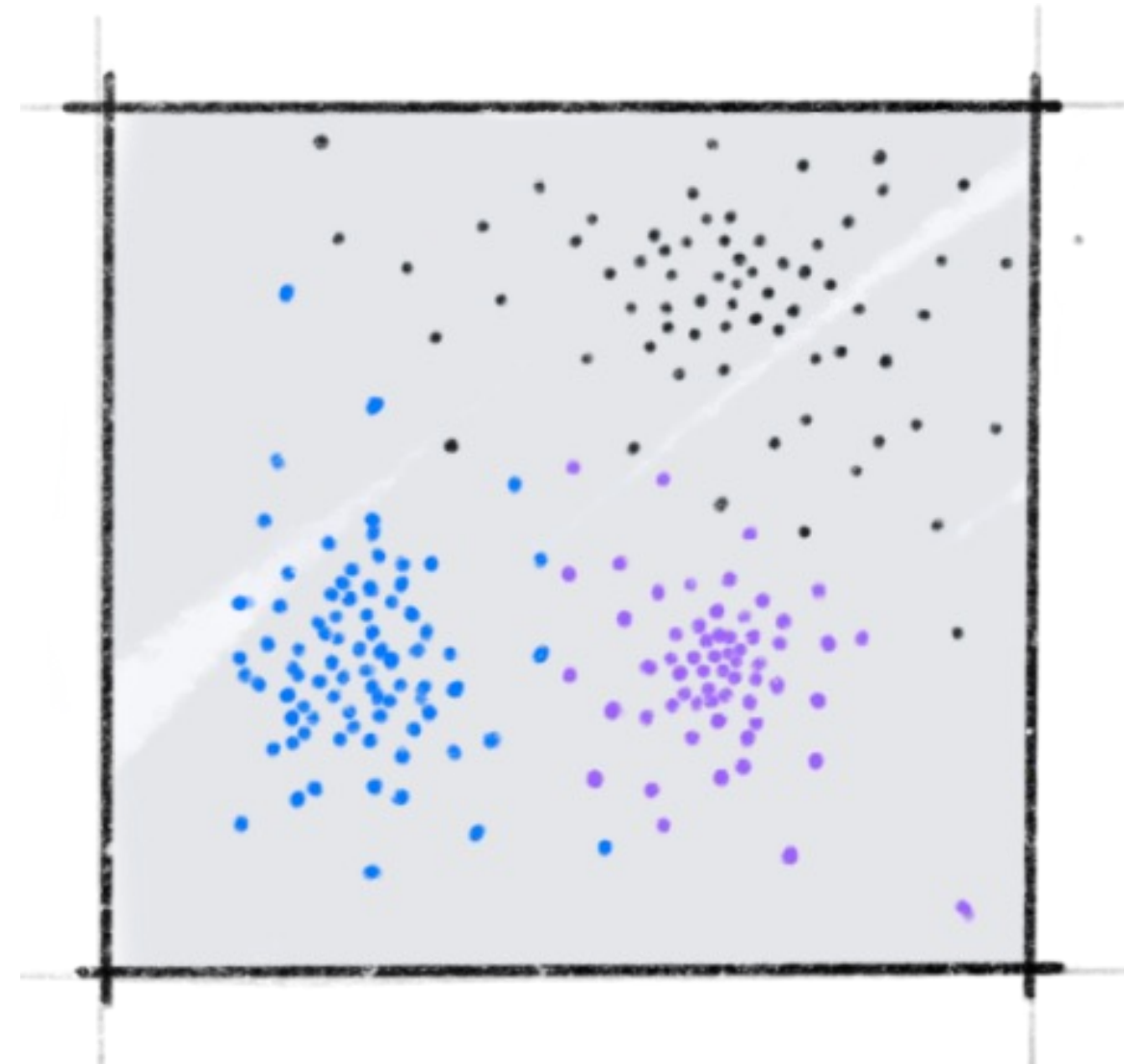
教師あり学習



ラベル付きデータ (x_i, y_i) :
マッピングする関数 $y = f(x)$ を
学習。

例) 猫や犬の写真がラベル付けされた
集合から、新しい猫や犬の写真を識別す
る。

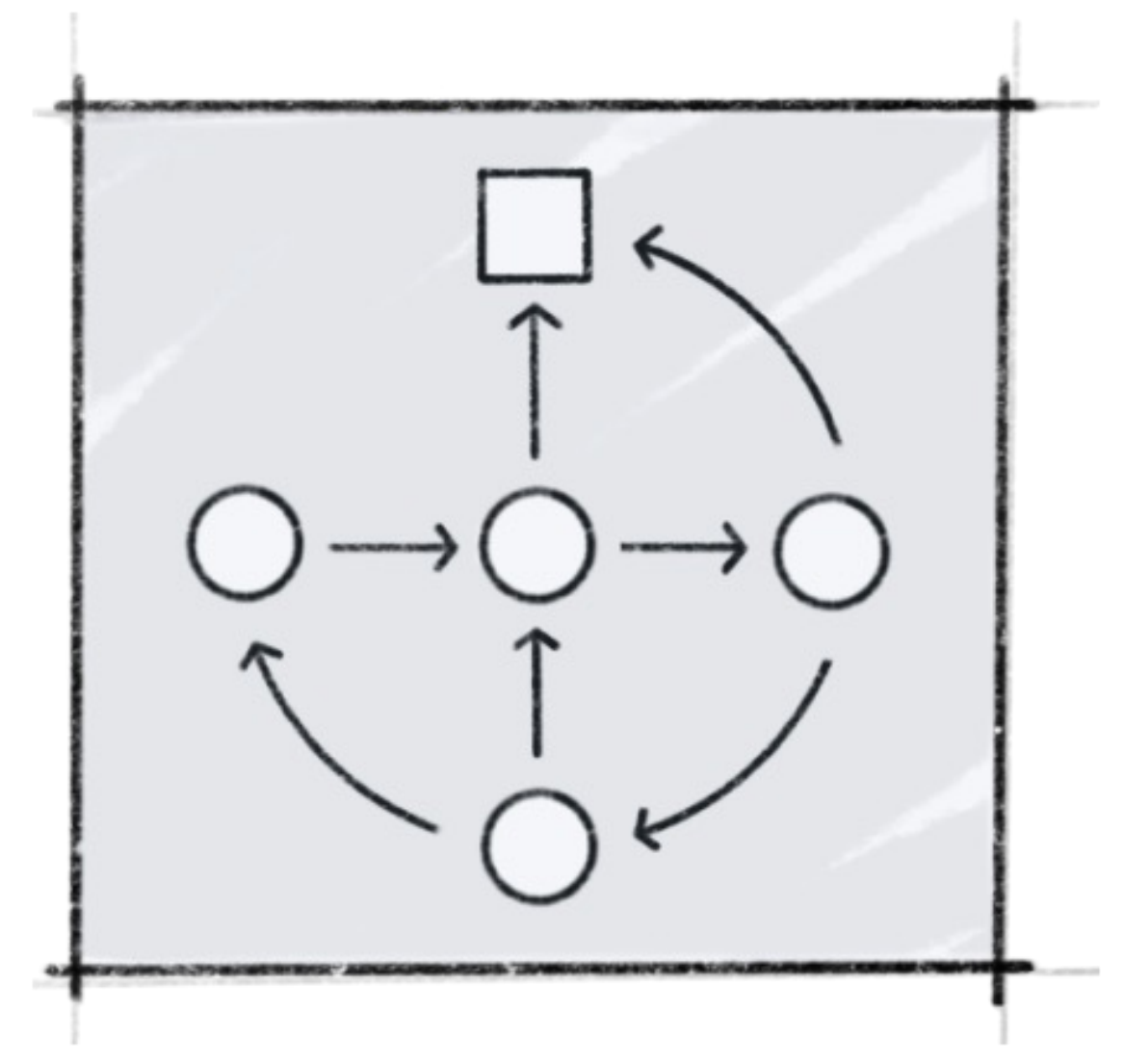
教師なし学習



ラベルのないデータ :
何らかの構造を学習。

例) 映画の視聴履歴に基づいて視聴者を
グループ分けし、新しい映画を推薦す
る。

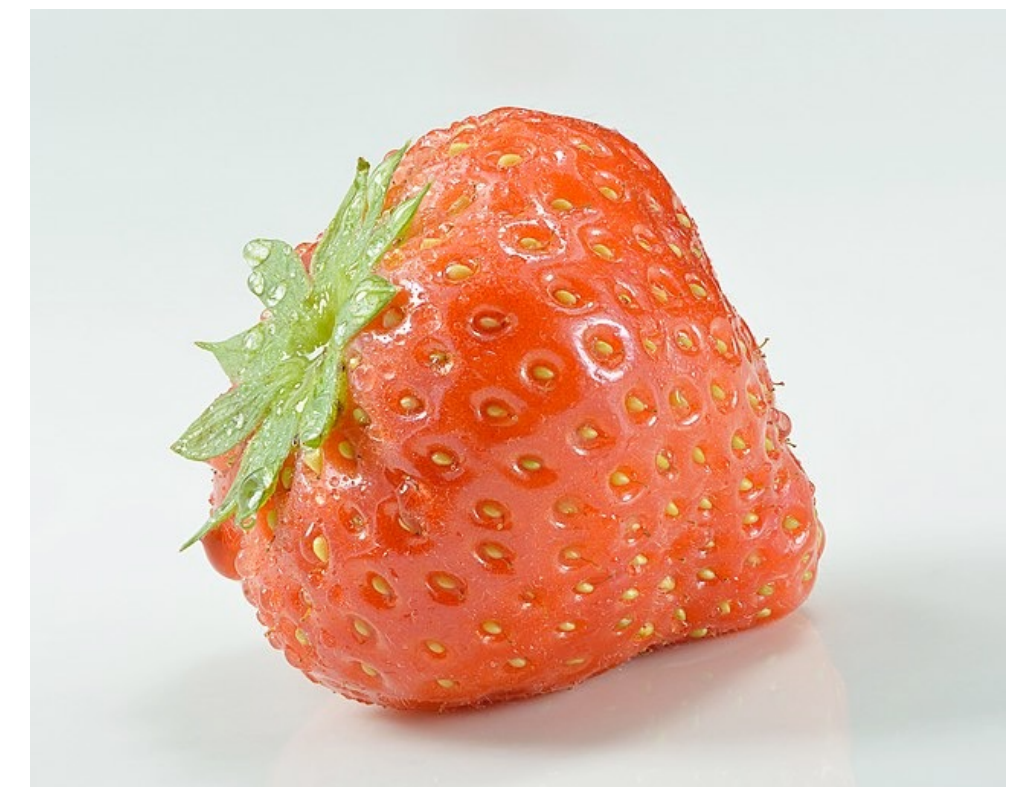
強化学習



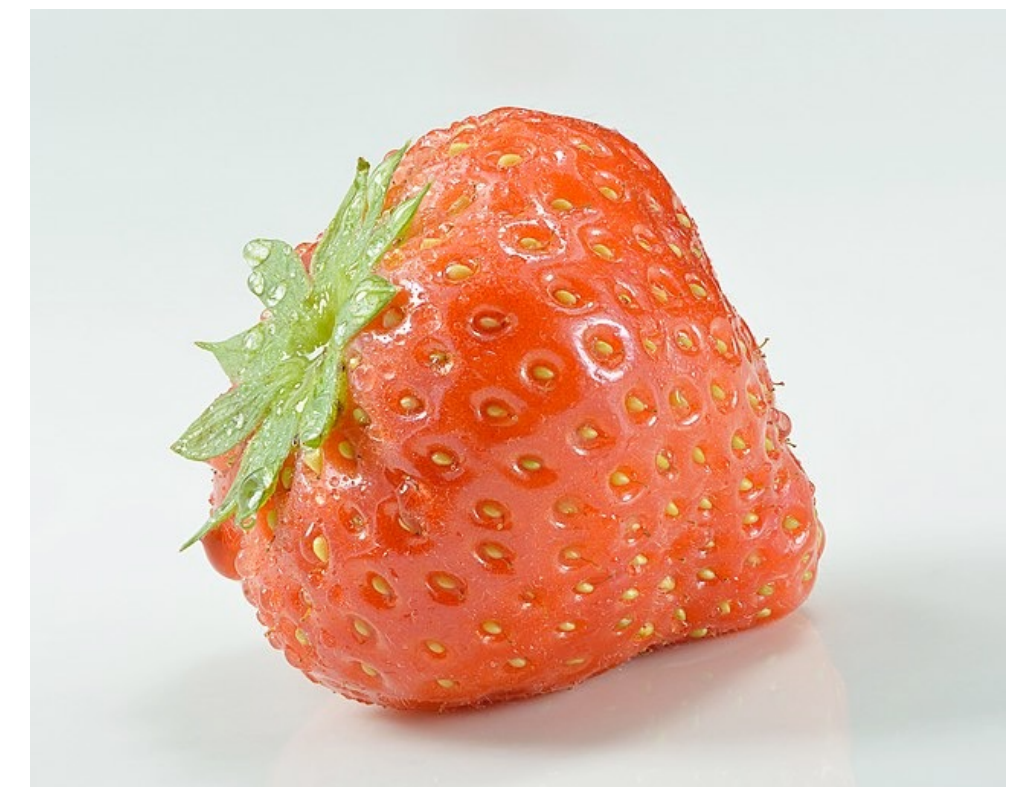
行動に応じて報酬が得られる環境
で、期待される報酬を最大化。

例) 「パックマン」のプレイ方法をアル
ゴリズムで学習する。

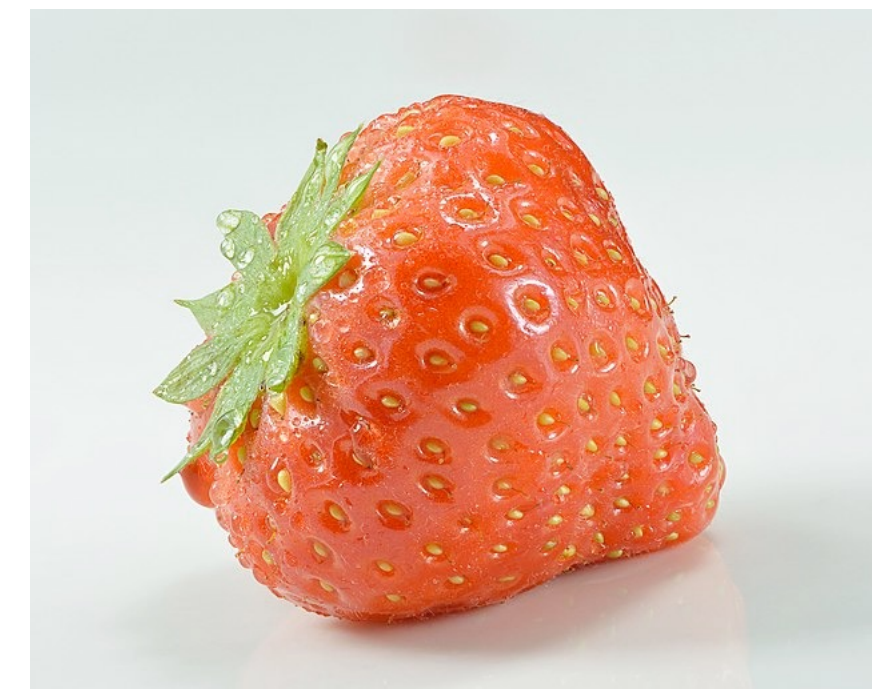
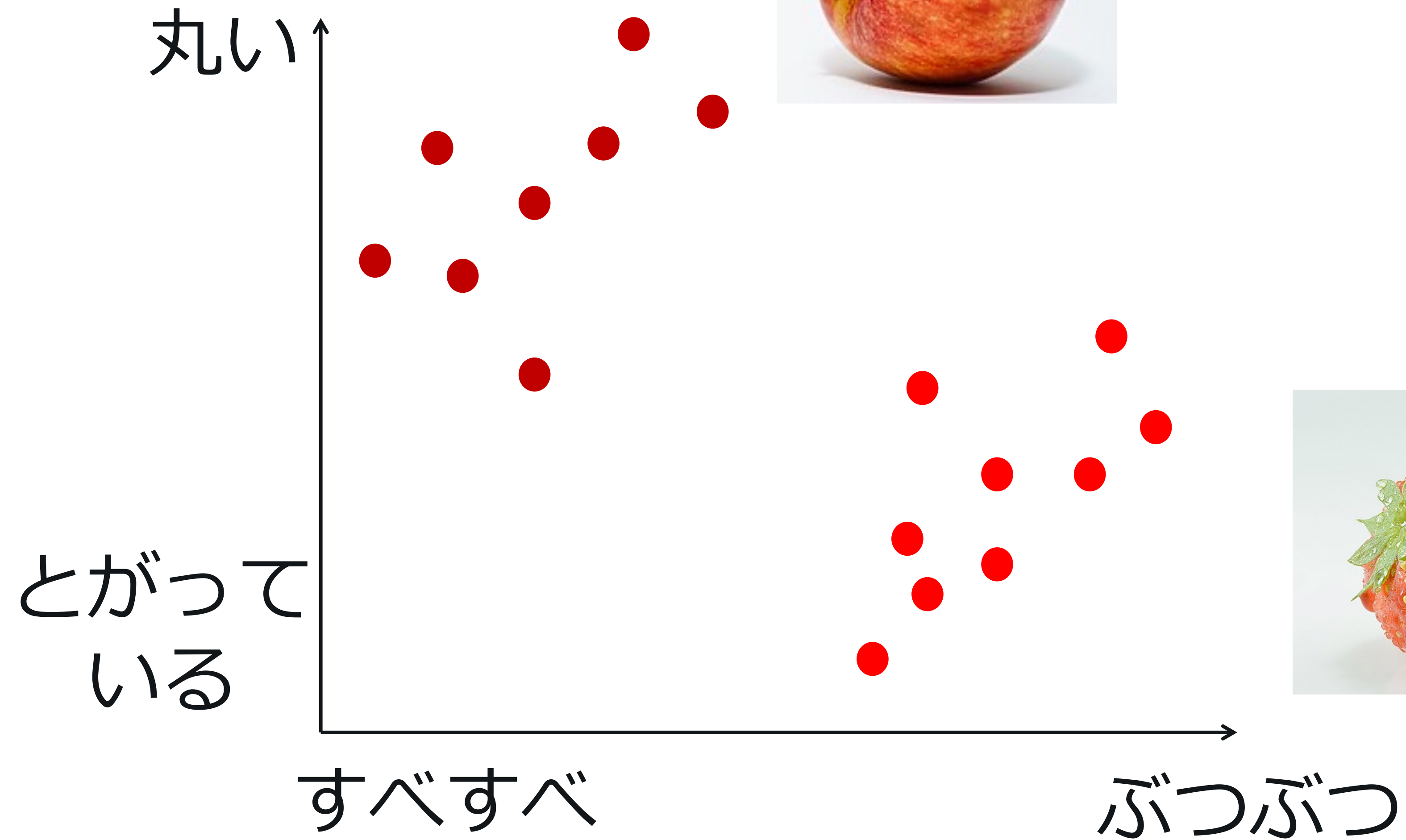
イチゴとリンゴをどうやってコンピューターは 見分けるのでしょうか？



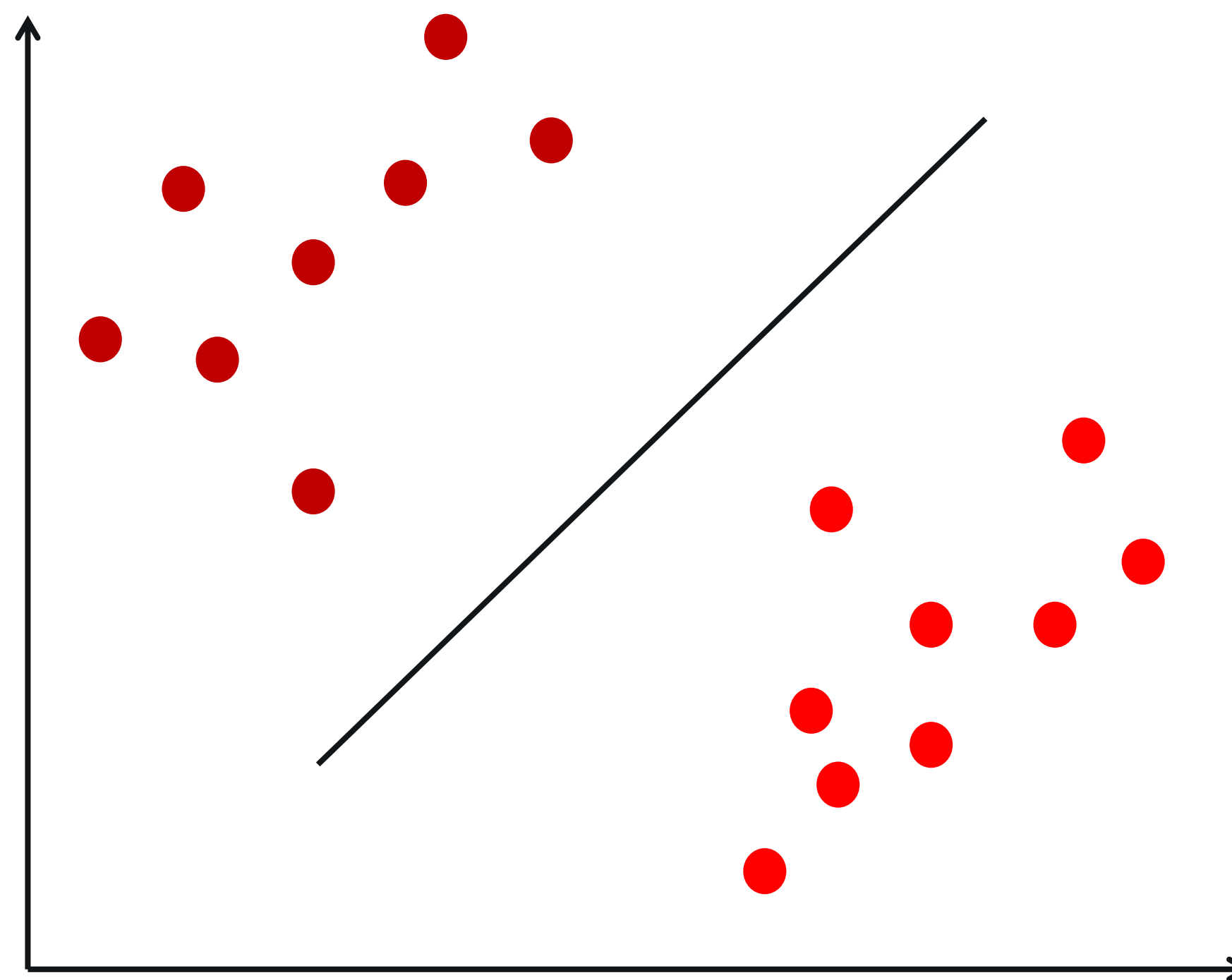
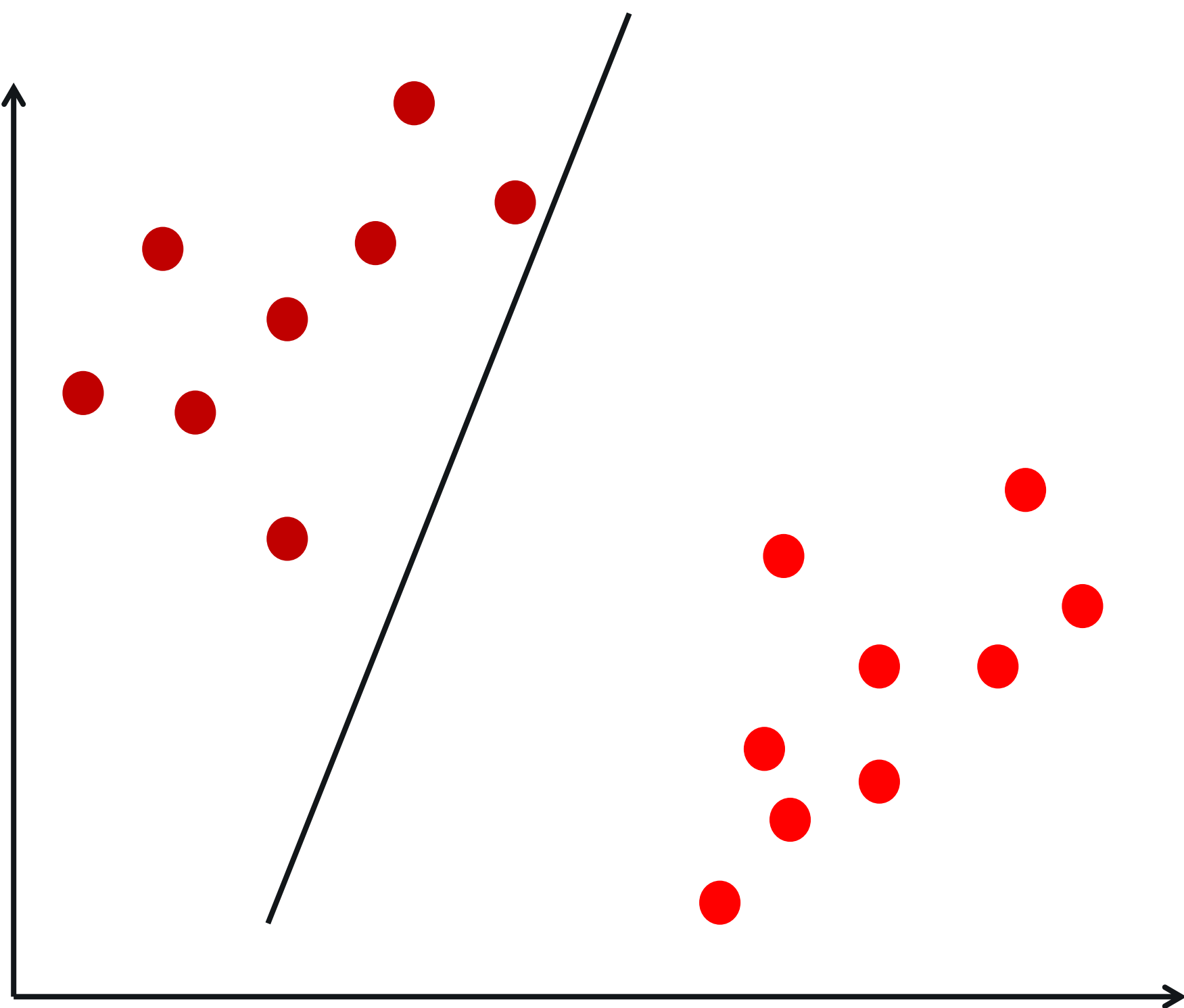
特徴をもとに判別しています。
イチゴとリンゴを区別する特徴は何でしょう？



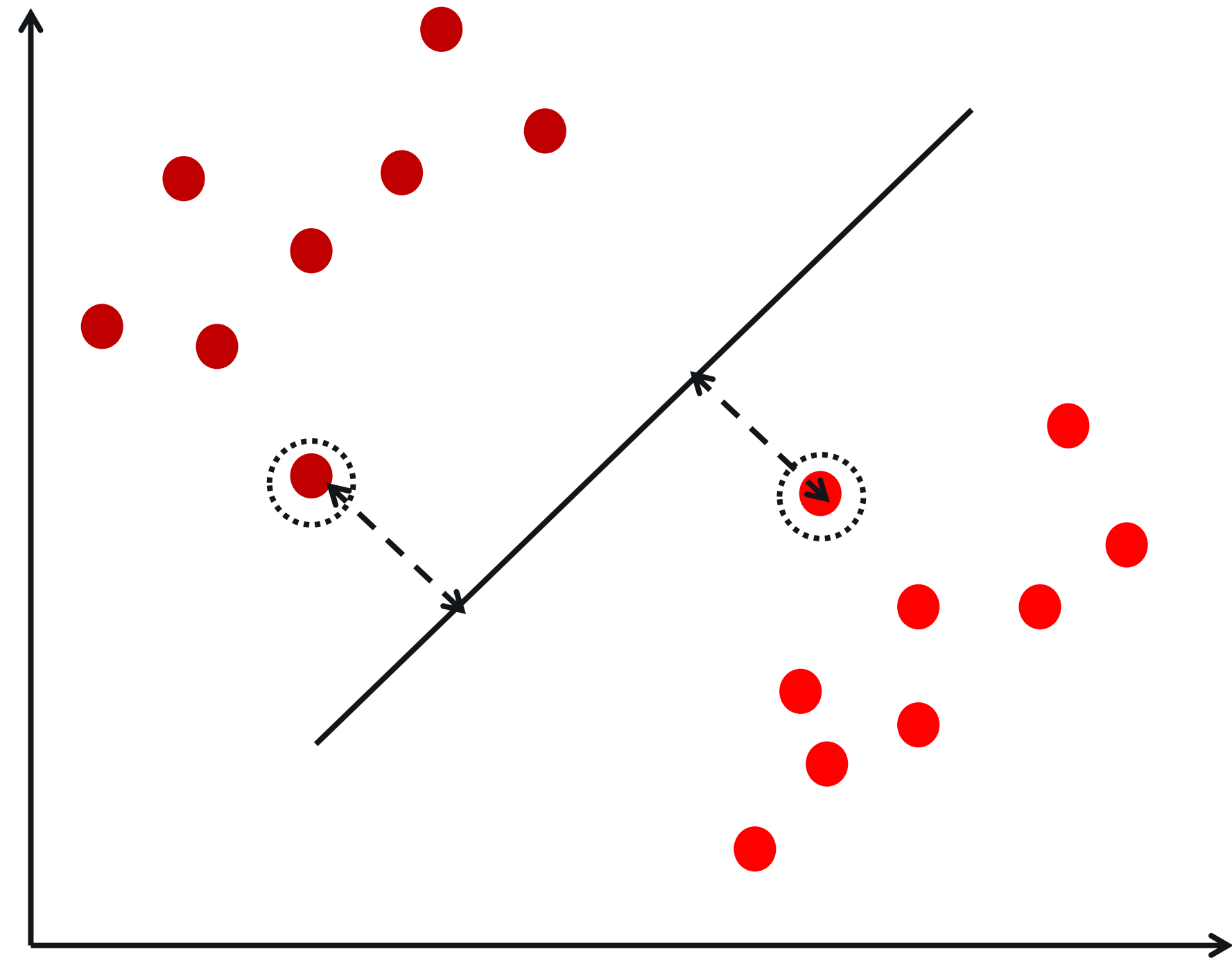
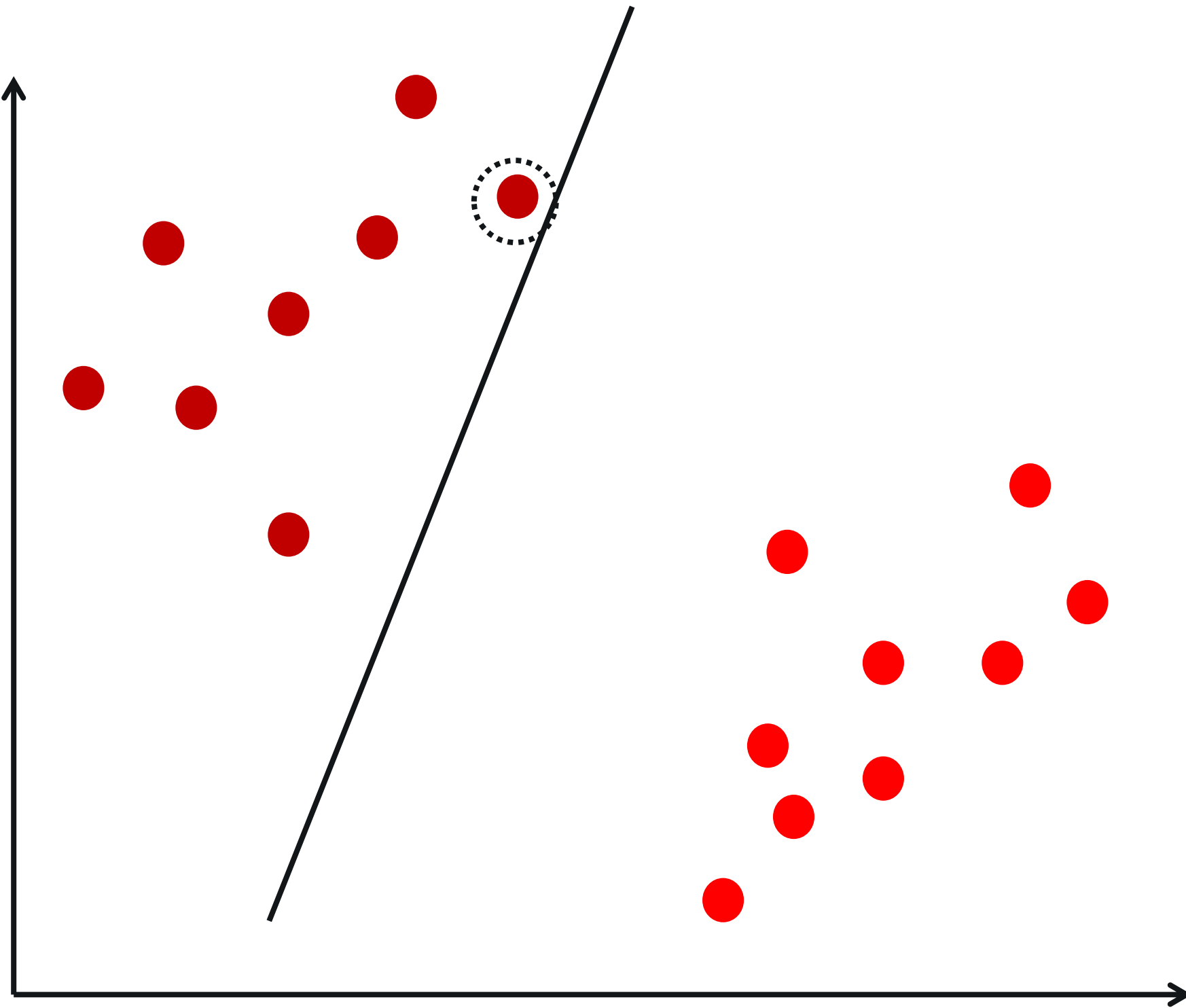
イチゴとリンゴ



どちらの方がよく分類できているでしょうか？



右図の方が境界線と最も近いデータ点との距離が長い

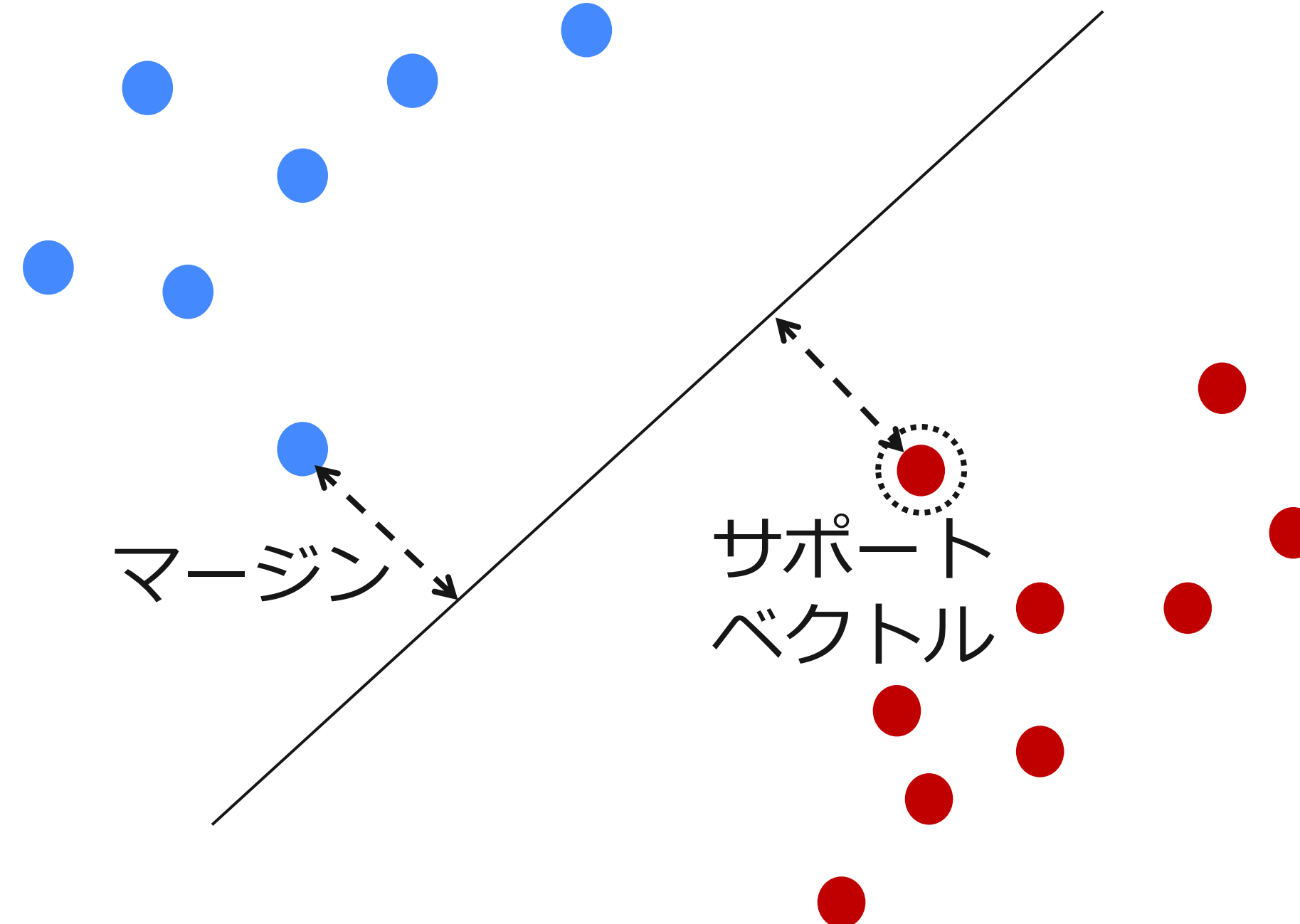


より安定した分け方

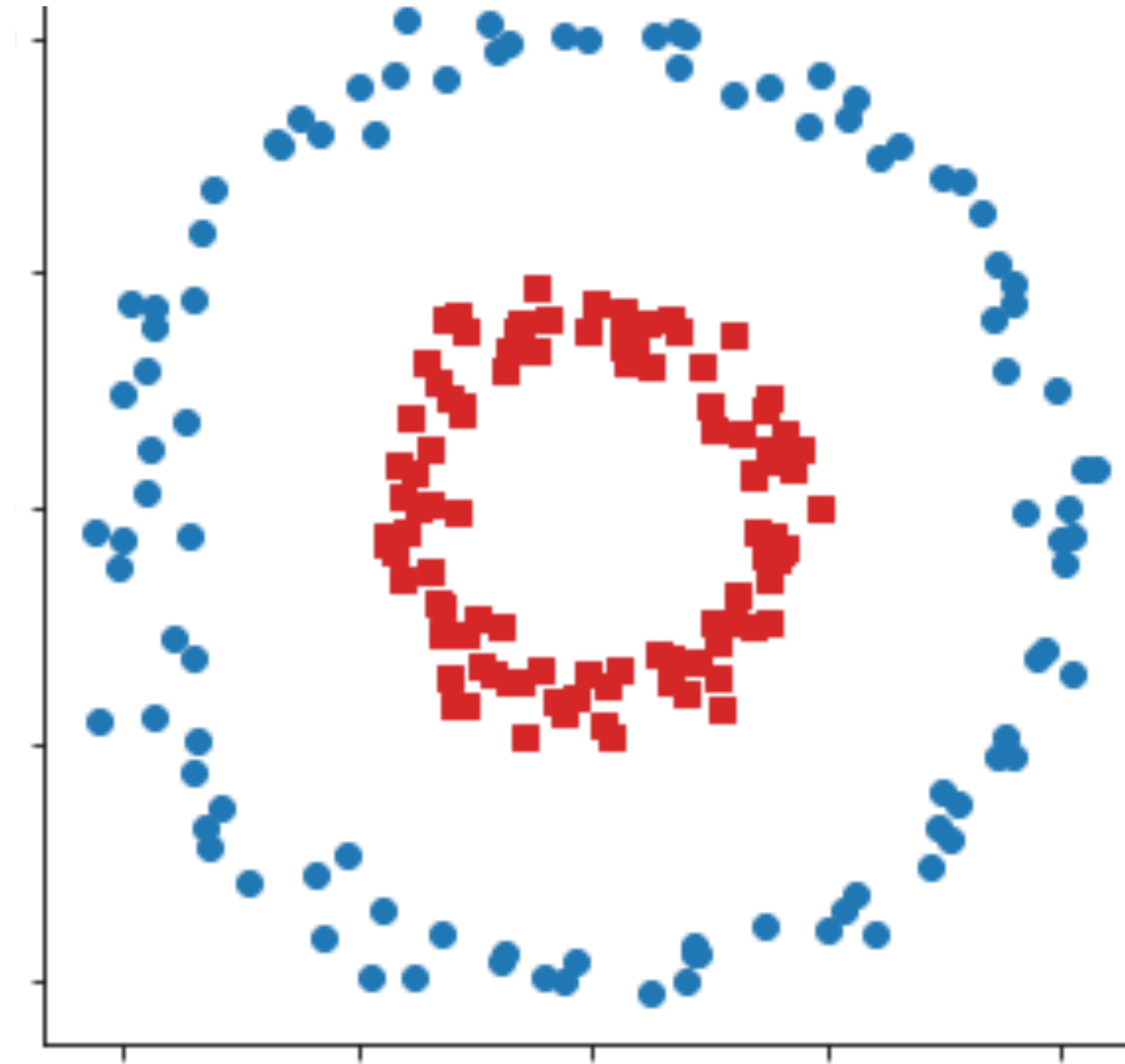
SVM(サポートベクターマシン) とは

データを2つのグループに分ける手法(2値分類)

- グループ間の境界面を定める分析手法
- マージン（境界線と最近接データ点との距離）をできるだけ大きく取るように最適化

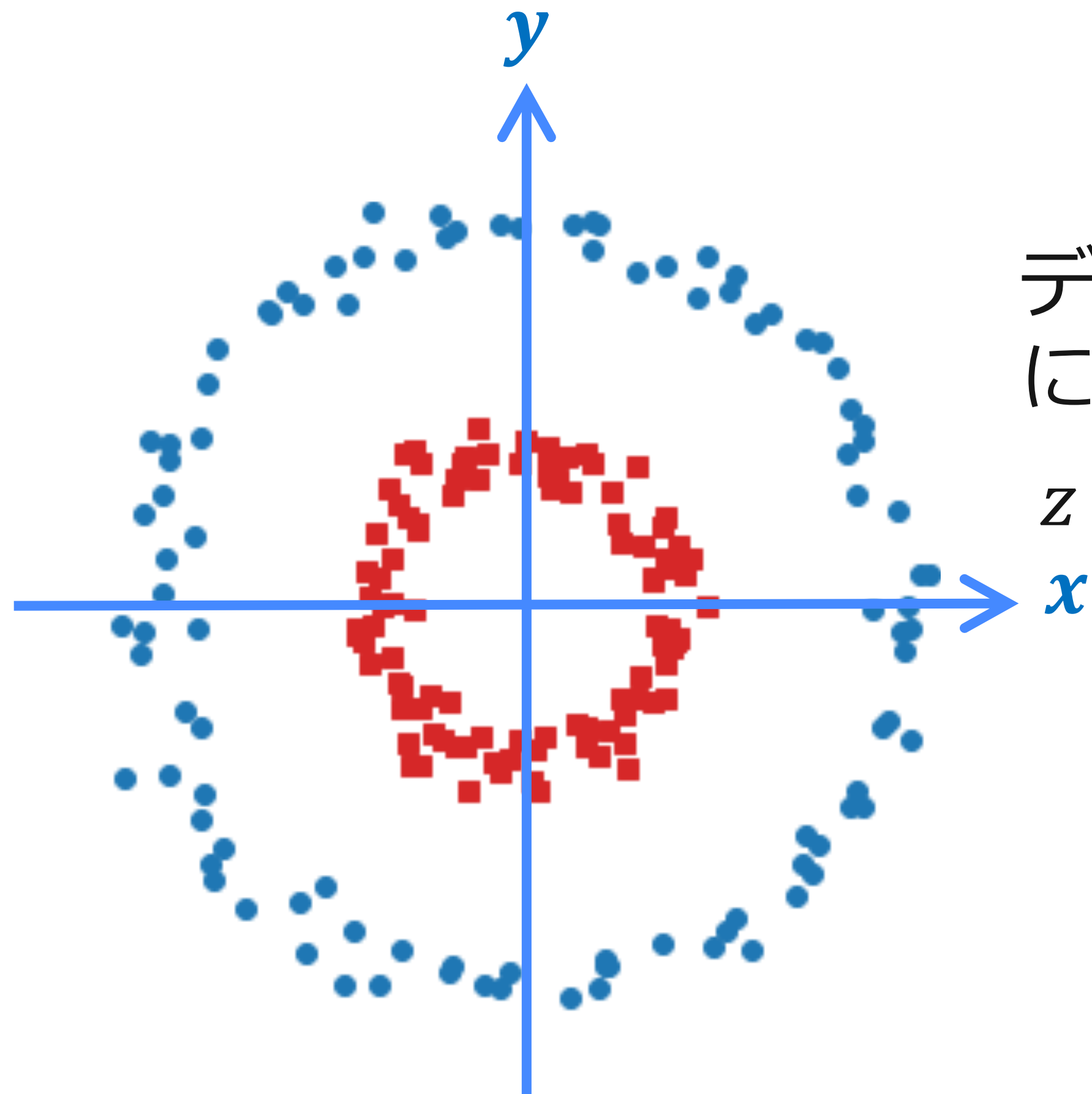


直線で分けられないデータの場合



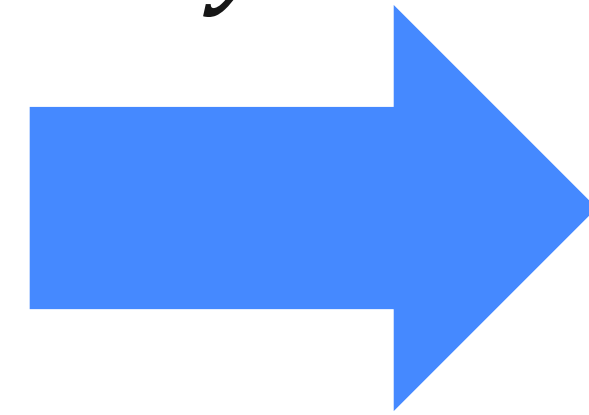
このようなデータセットは、
2グループに分けられることは
明らかですが、
境界線が直線にはなりません。
(線形に分離できないといいます)

データマッピングで分類

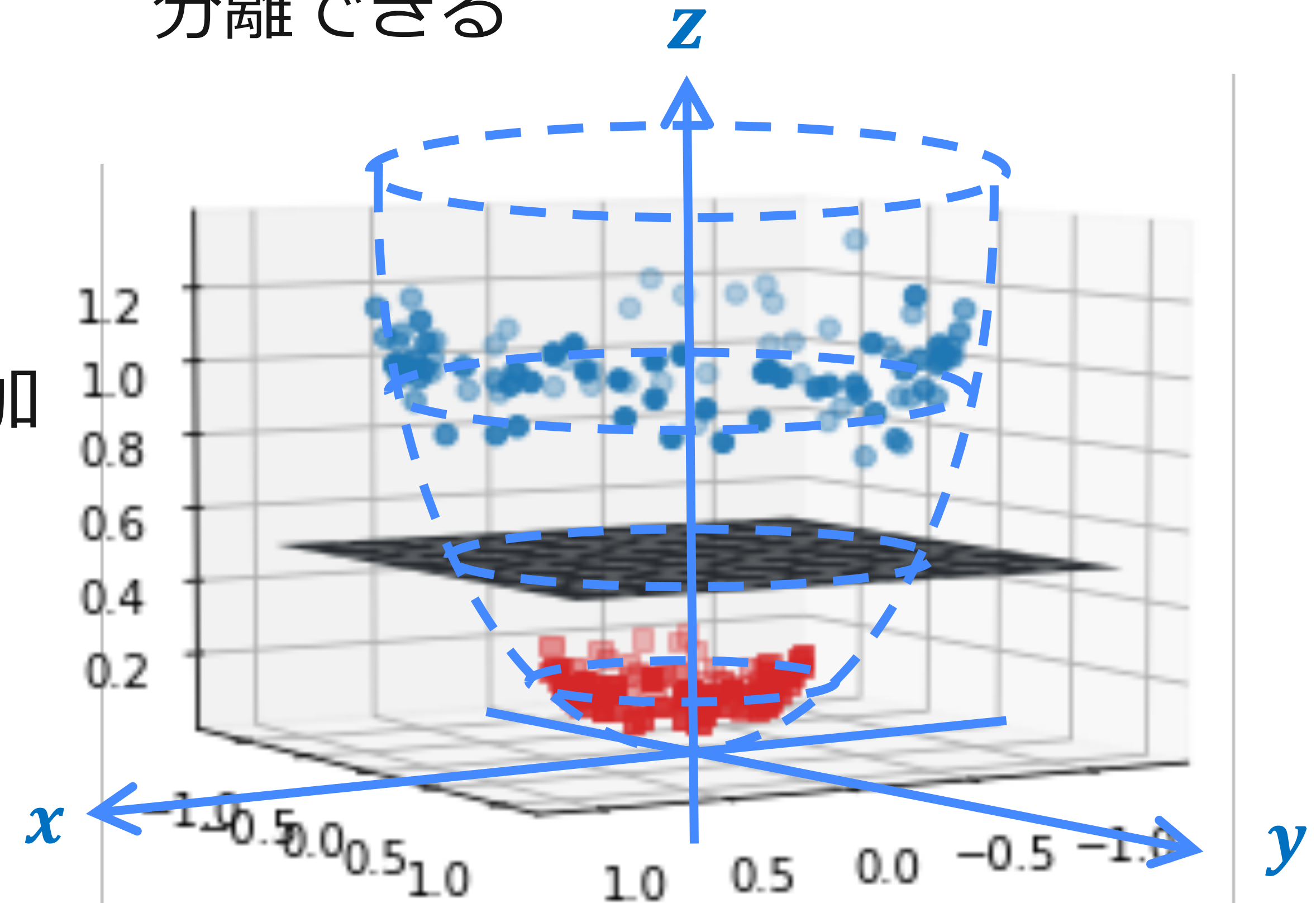


データを3次元
に変換

$$z = x^2 + y^2 \text{を追加}$$



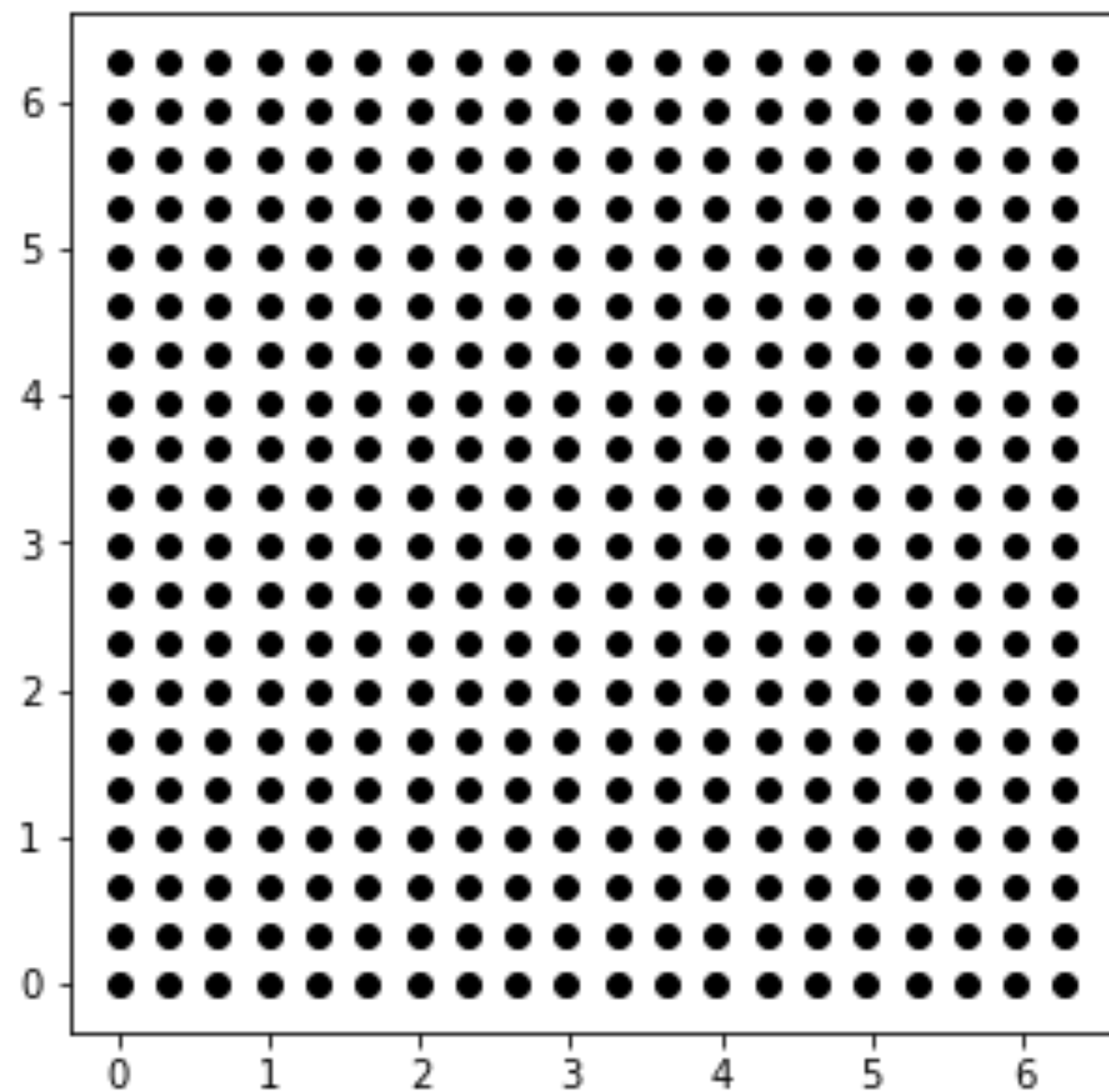
3次元では、データは平面で
分離できる



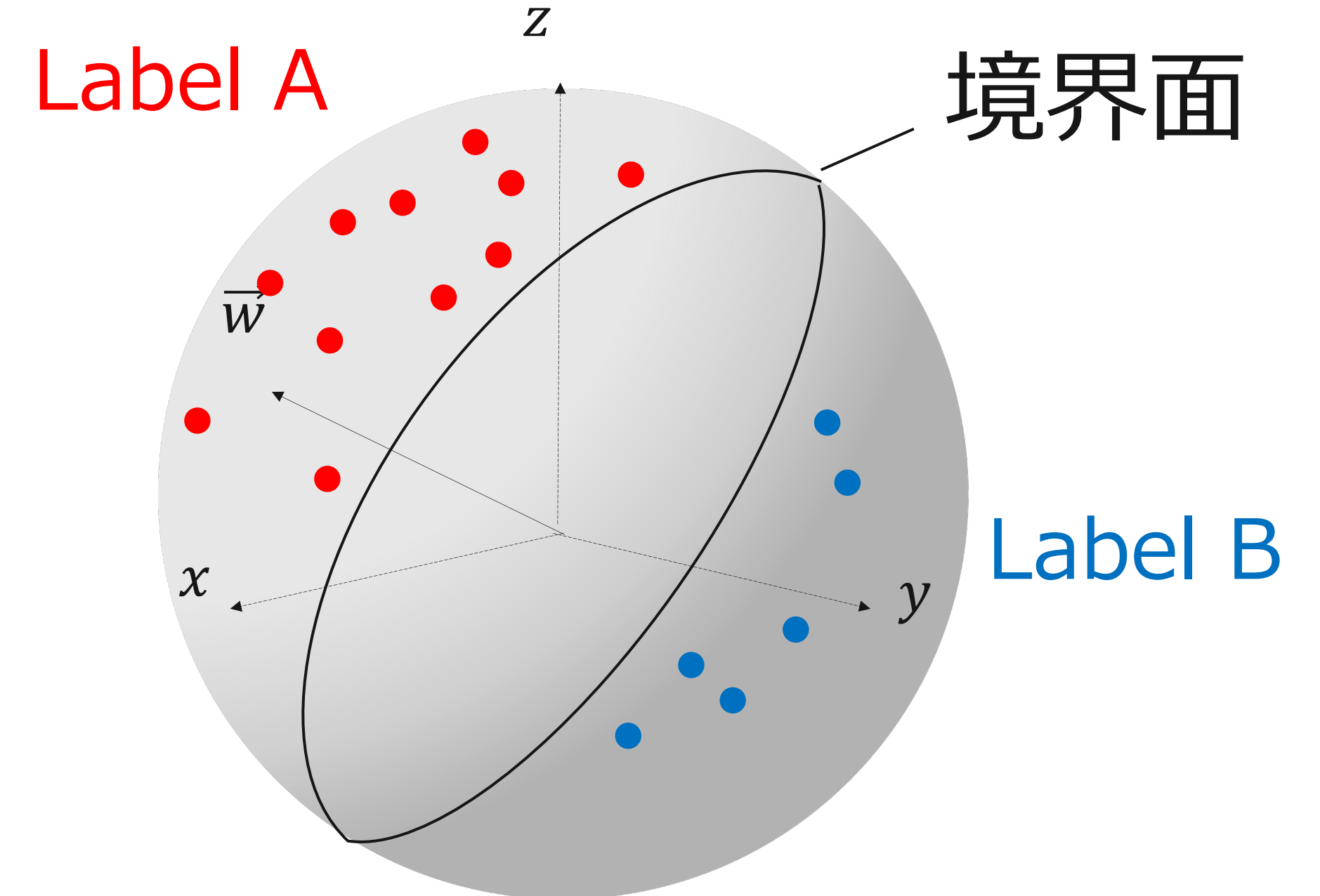
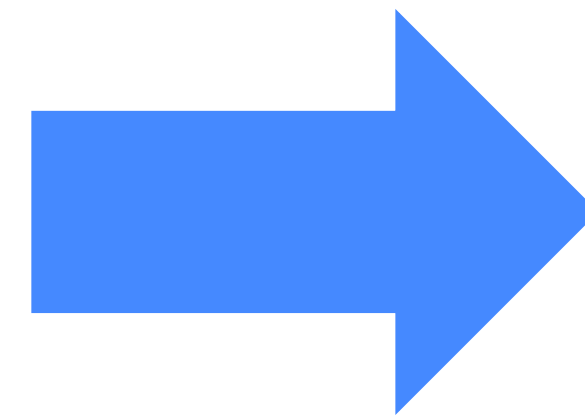
特徴量を高次元化(特徴量マッピング)することで、
平な(線形な)境界面で切り分けることができます。

量子SVM(サポートベクターマシン)

特徴量を量子空間に特徴量マッピングすることで、線形な境界面で切り分けます。



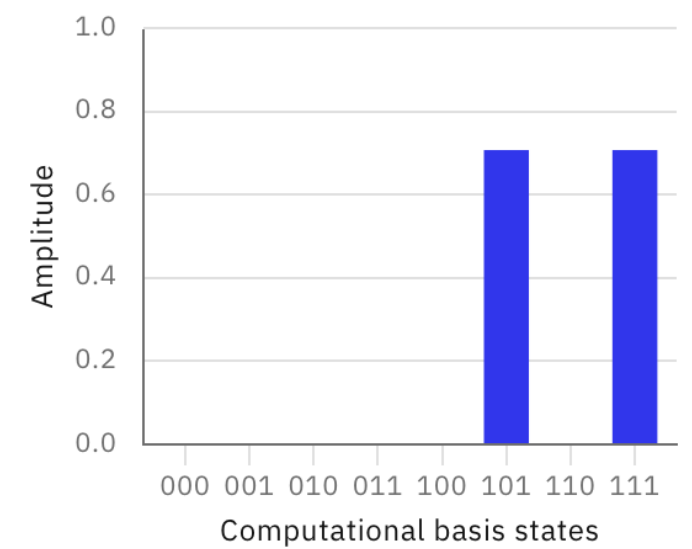
量子状態の球に
マッピング



データを量子機械学習のために符号化する手法（代表的なもの）

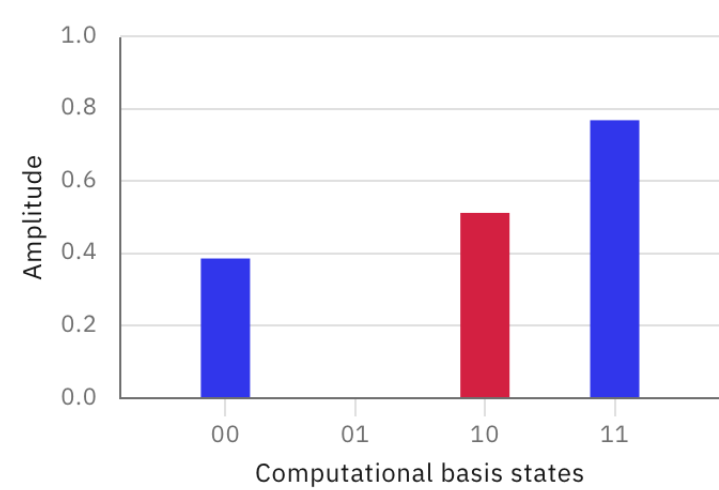
1. 計算基底符号化

例) データセット $X = \{x_1 = 101, x_2 = 111\}$ \rightarrow 量子状態 $|x\rangle = \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle)$



2. 振幅符号化

例) $X = \{x_1 = (1.5, 0), x_2 = (-2, 3)\}$ \rightarrow $|x\rangle = \frac{1}{\sqrt{15.25}}(1.5|00\rangle - 2|10\rangle + 3|11\rangle)$



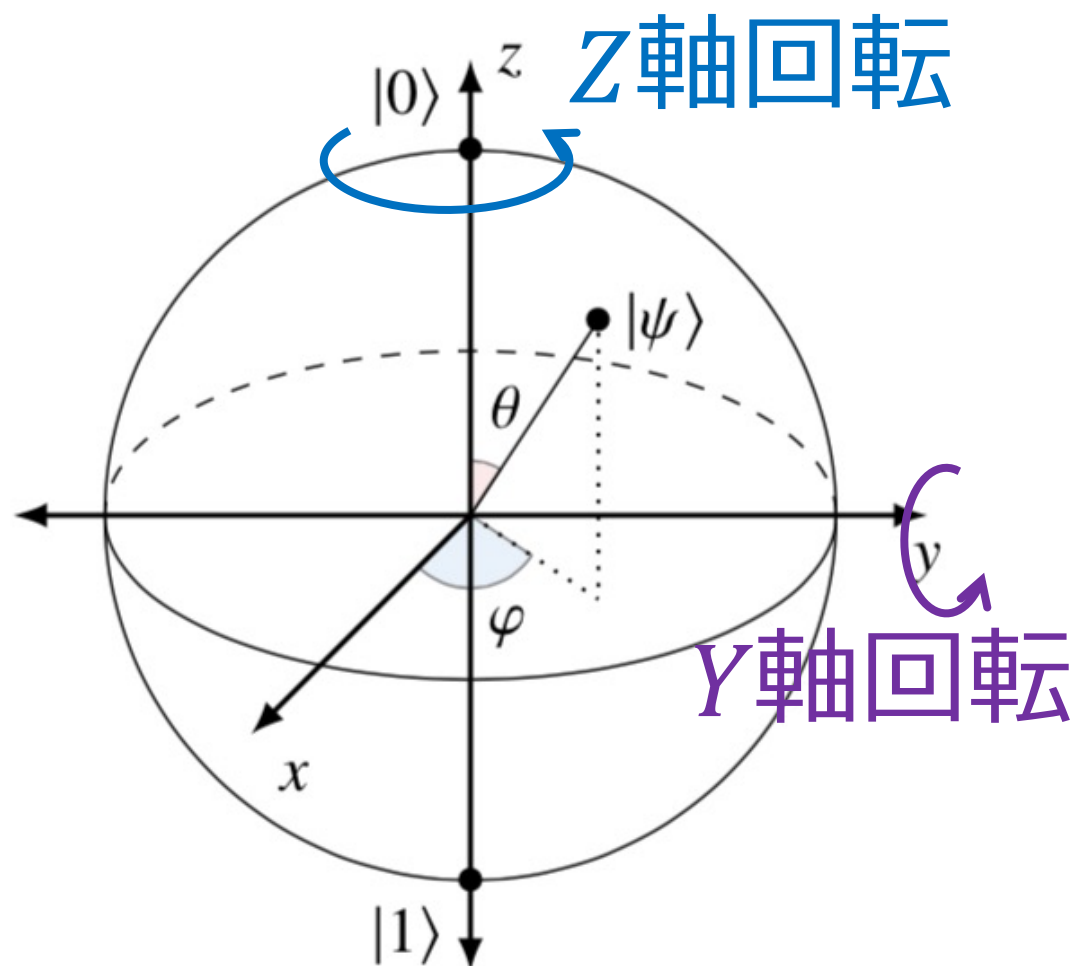
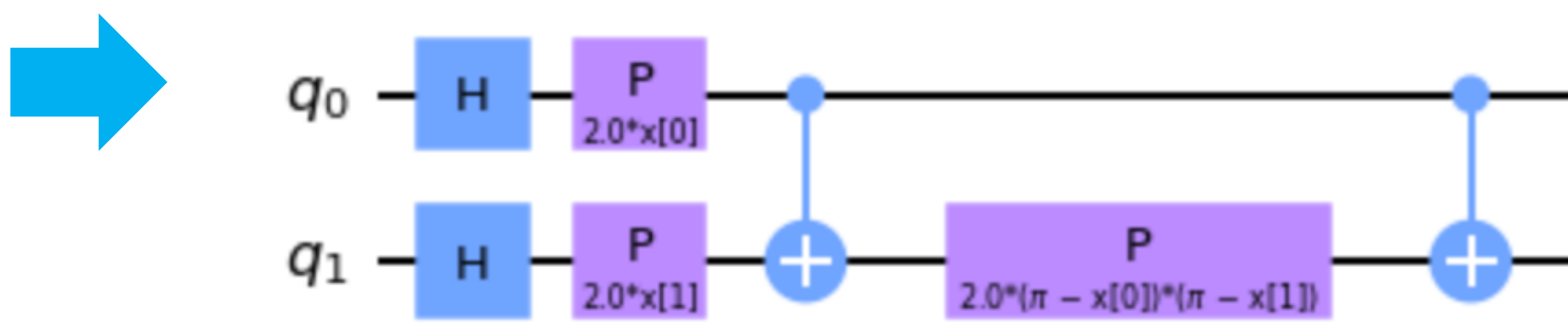
3. 角度符号化

例) データポイント $x = (x_1, x_2)$ \rightarrow $S_x = RY(x_1) \otimes RY(x_2)$



4. 角度符号化の応用

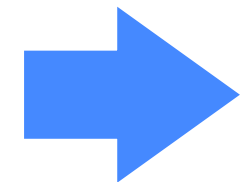
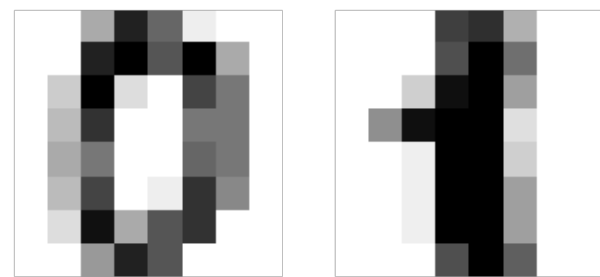
例) $x = (x_1, x_2)$



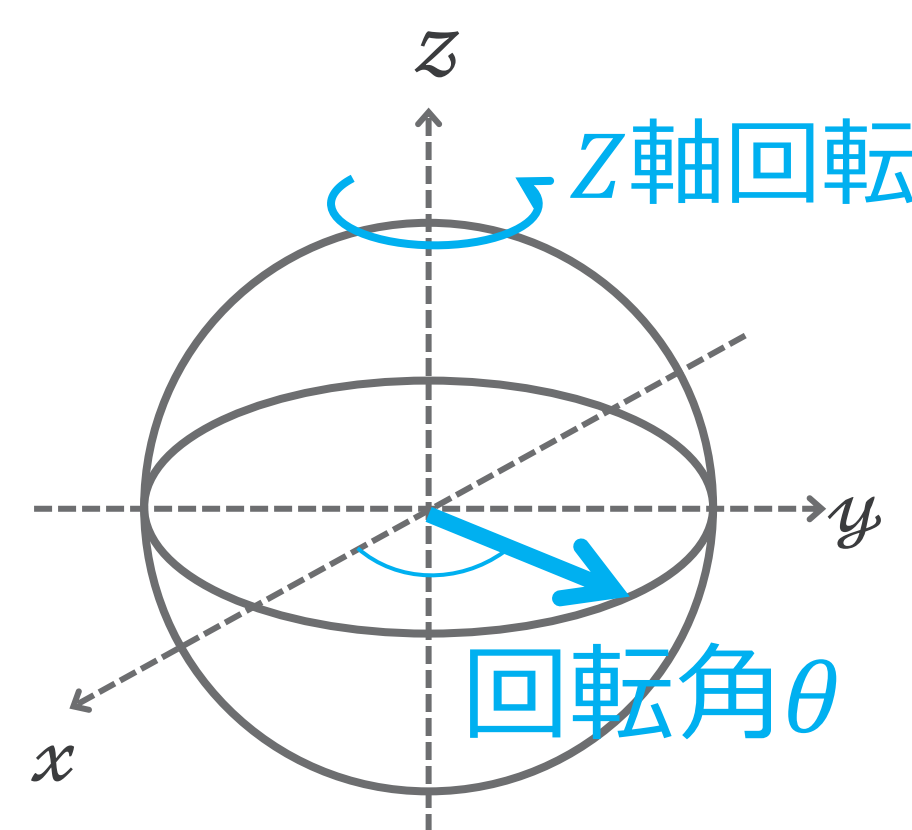
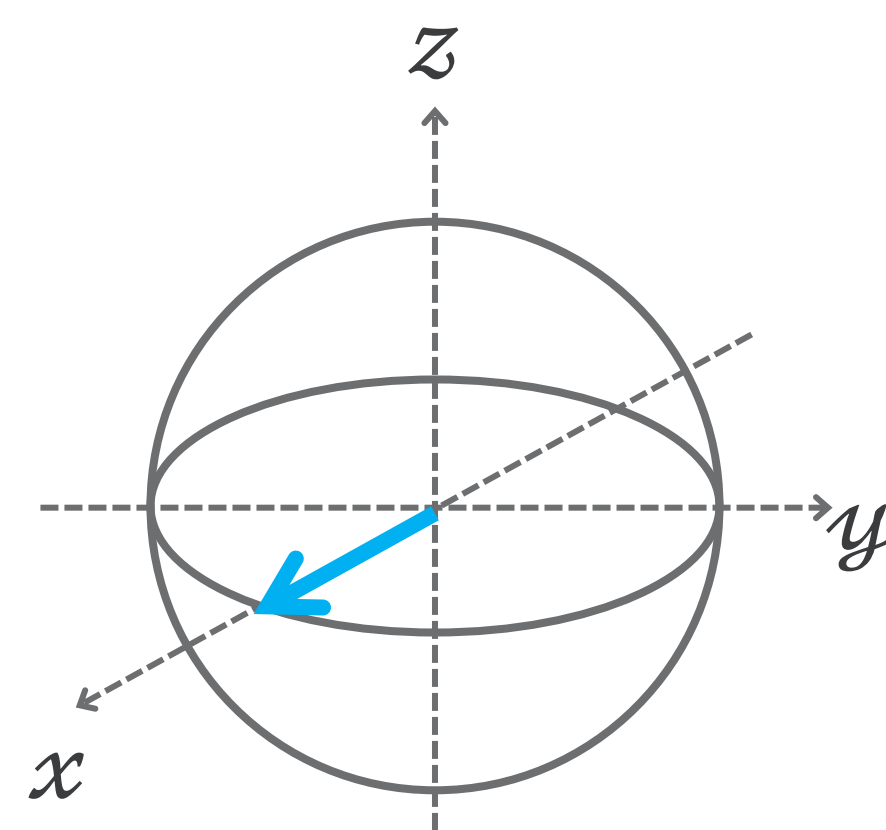
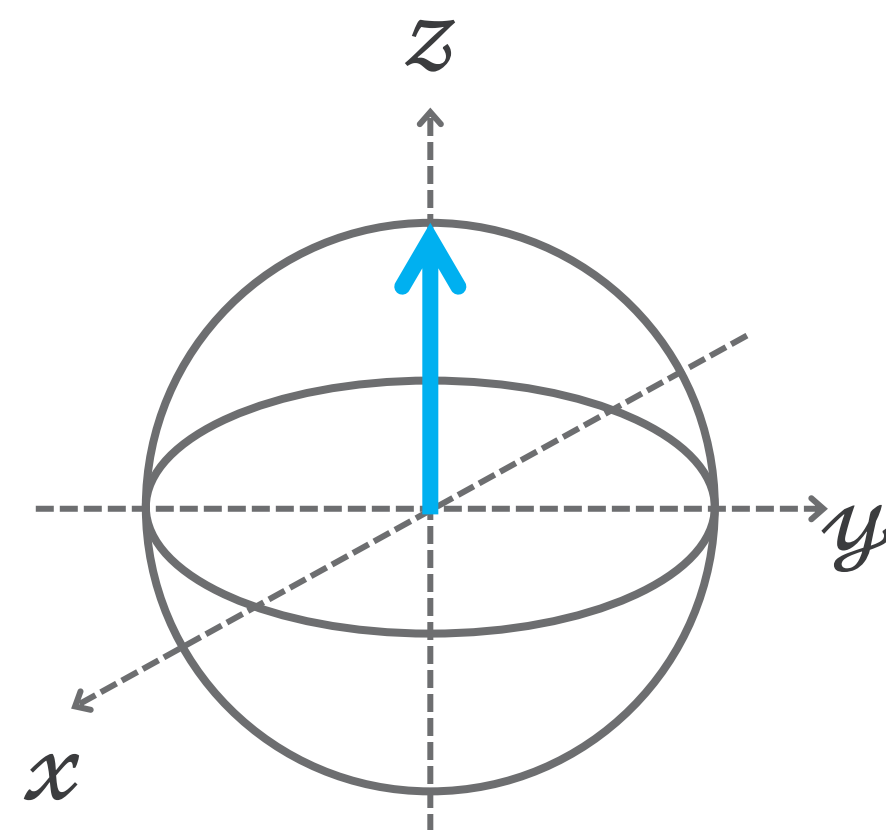
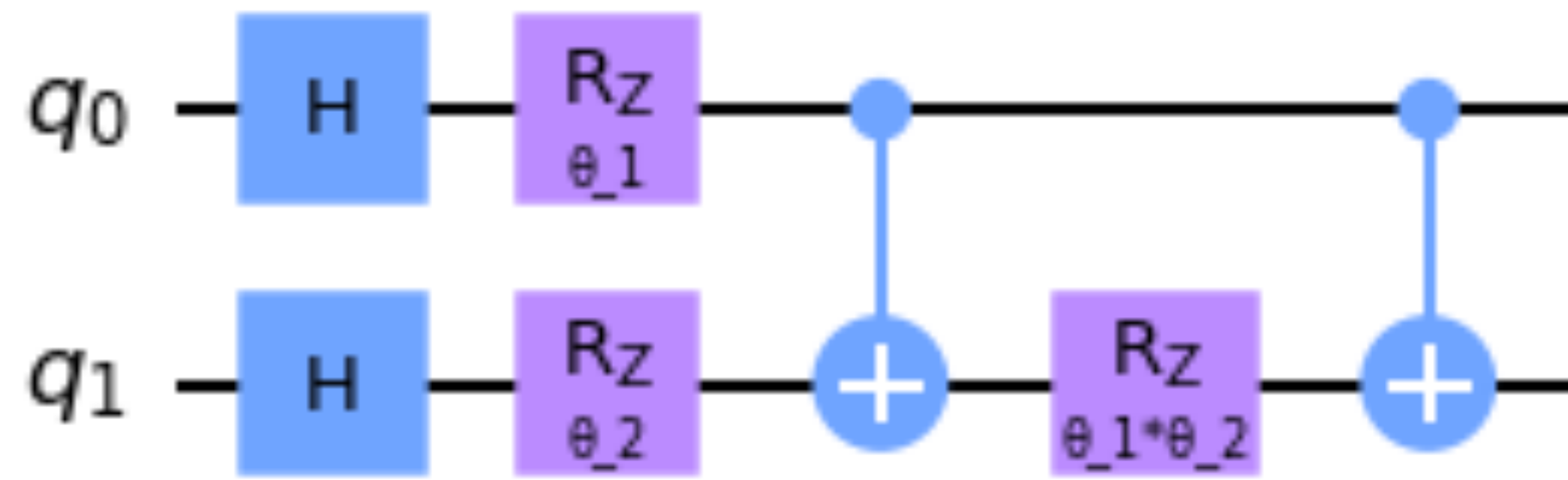
量子カーネルSVM

データを量子データにエンコード（符号化）する際に、
パラメーター（量子ゲートの回転角 θ ）を使った、角度符号化の応用の
量子特徴量マップ (Feature Map)を使って、回転角 θ の部分にデータを入れます。

入力データ



量子特徴量マップ

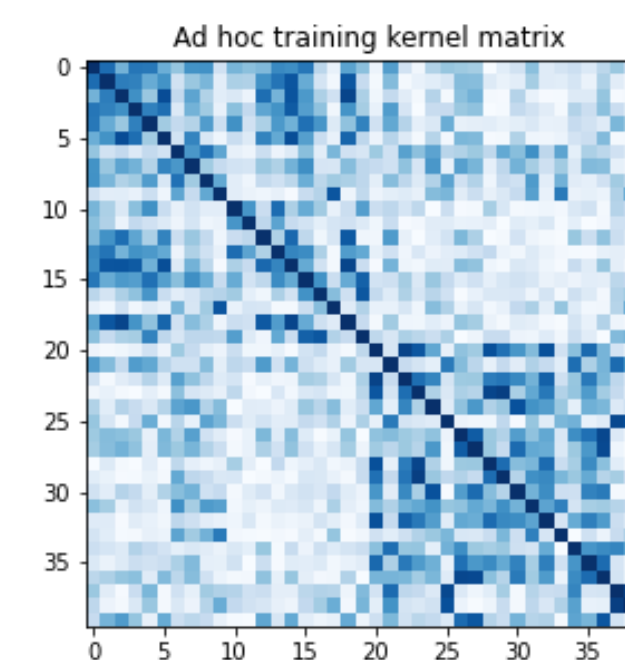
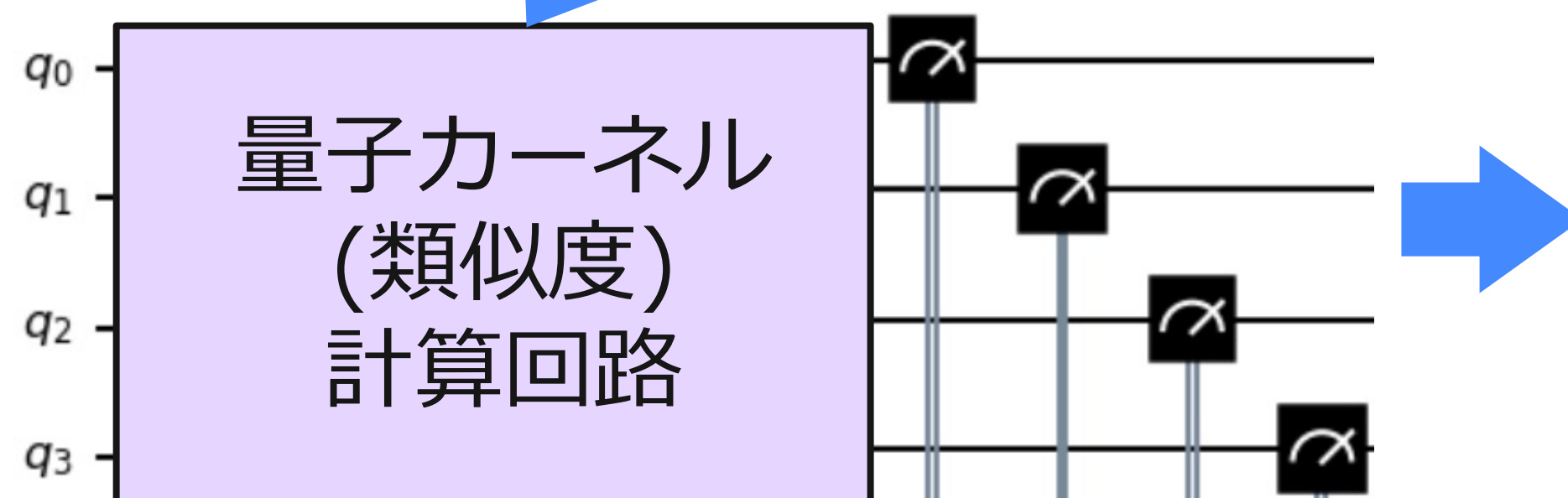
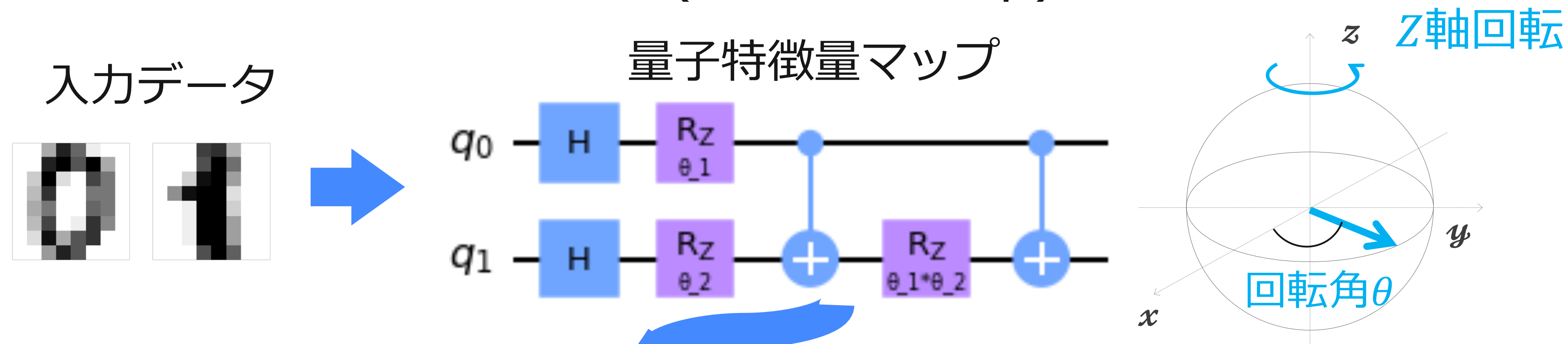


エンタングルメント

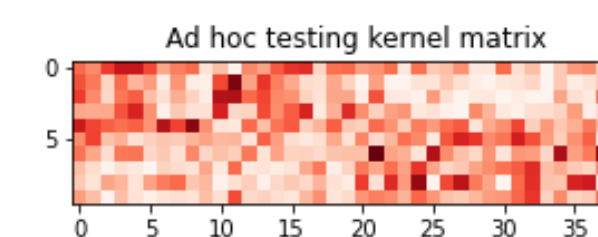


量子カーネルSVM

データを量子特徴量マップ(Feature Map)でエンコードした後、



カーネル
(類似度)計算

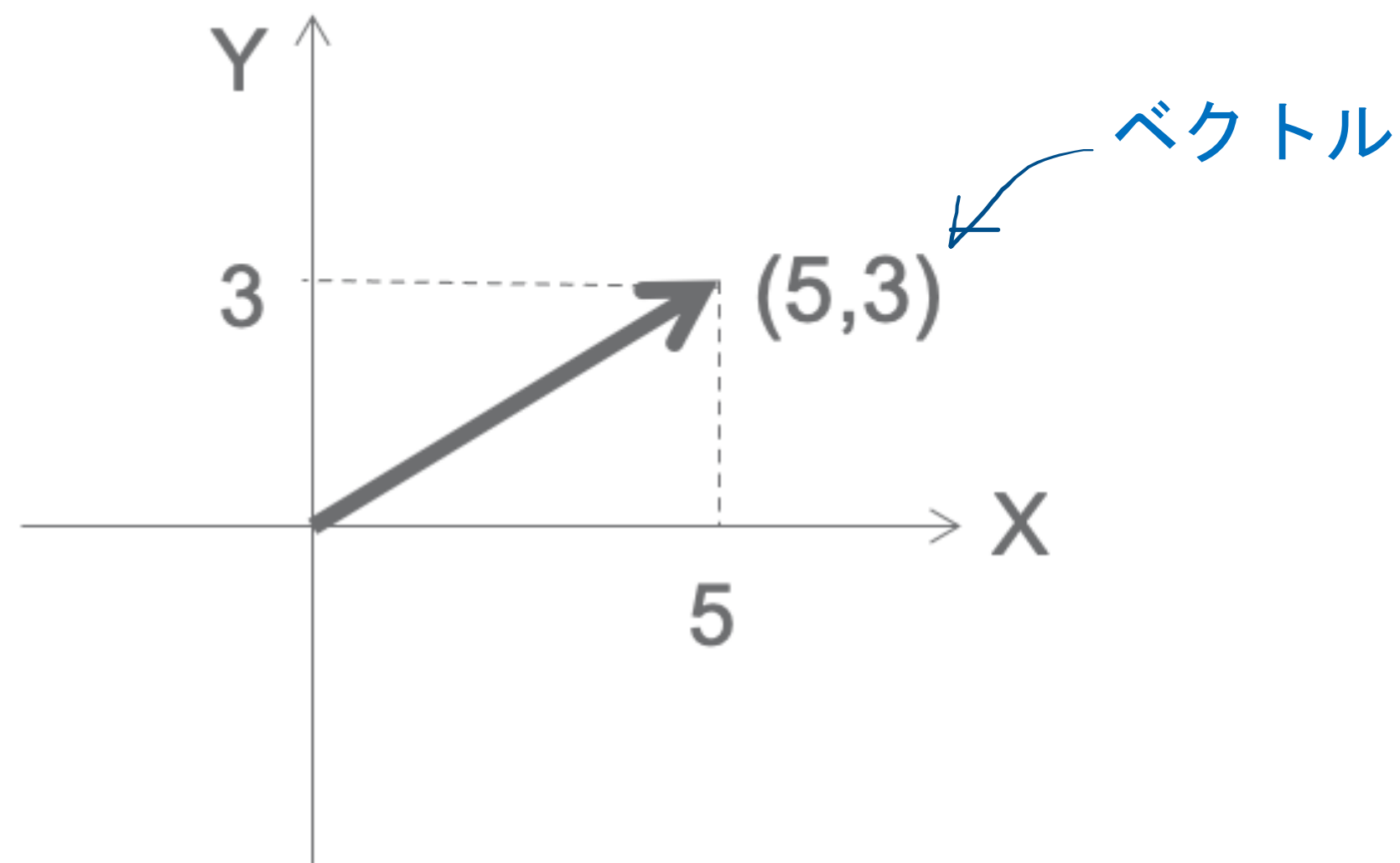


古典SVM
分類器

量子回路で量子カーネル(類似度)の計算を行い、
量子カーネルを使って、古典SVM計算(線形な境界面で分ける2値分類)で学習・分類を行います。

ベクトルとは

「大きさ」と「向き」を持った量です。



ベクトルは、数が横や縦に一行に並んだ形をしています。

横ベクトルの例

$$(5 \quad 3)$$

縦ベクトルの例

$$\begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

ベクトルを拡張して、数を長方形の形に並べたものが行列です。

行列の例

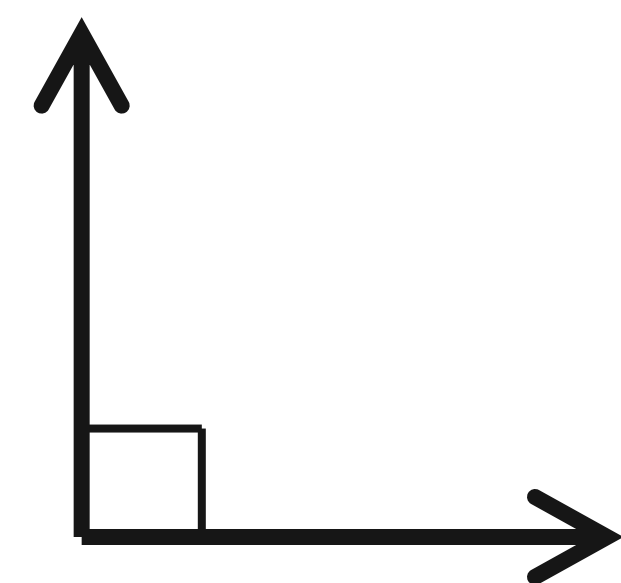
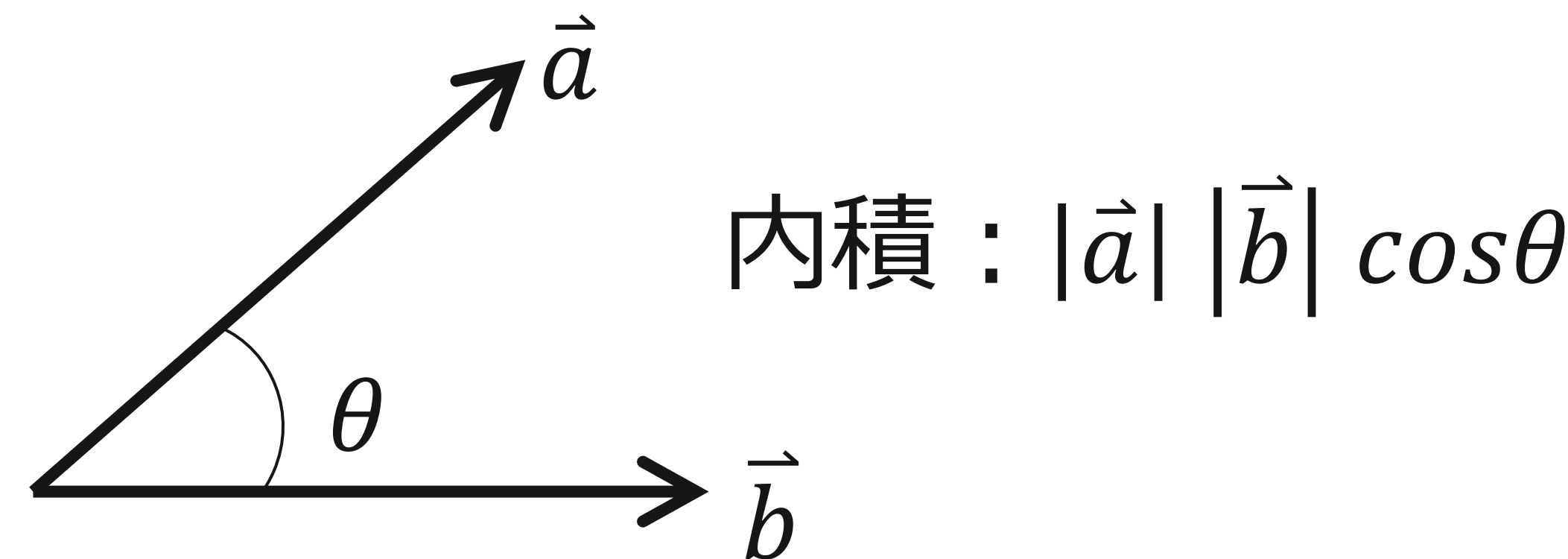
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

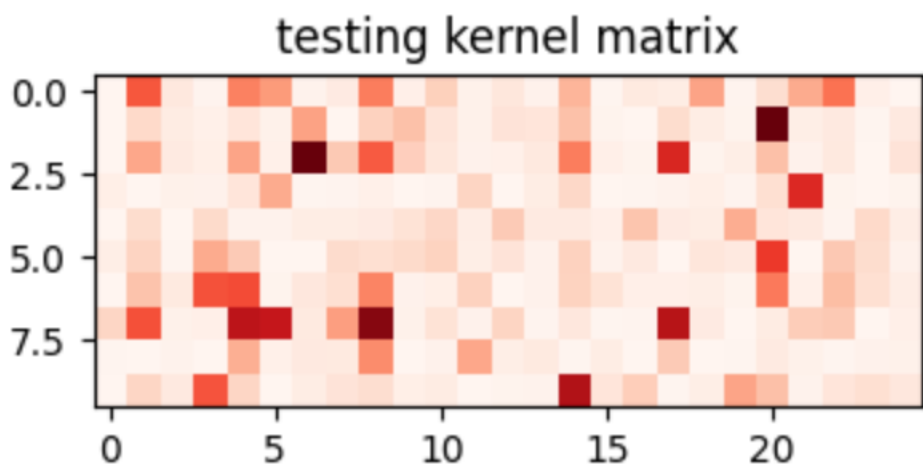
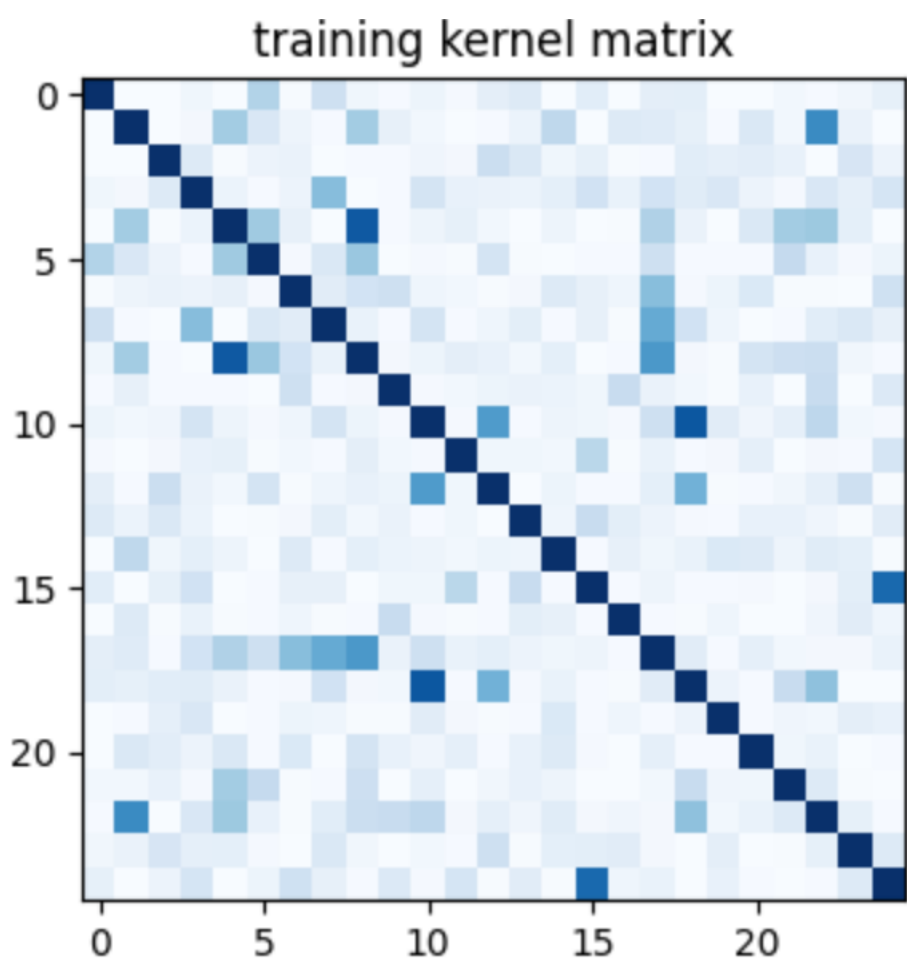
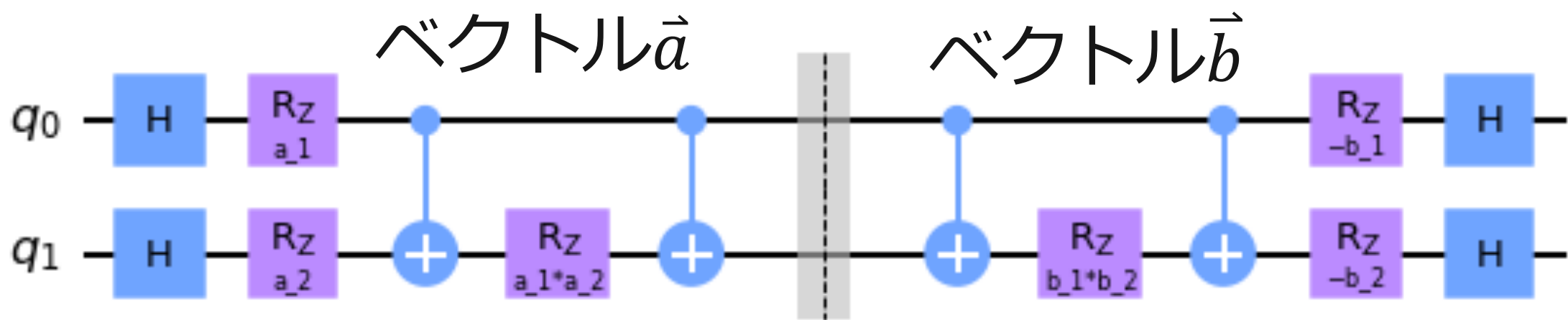
カーネル(類似度)計算は、ベクトルの内積の発展形



直行した
ベクトルの
内積は0

自分自身との
内積は1

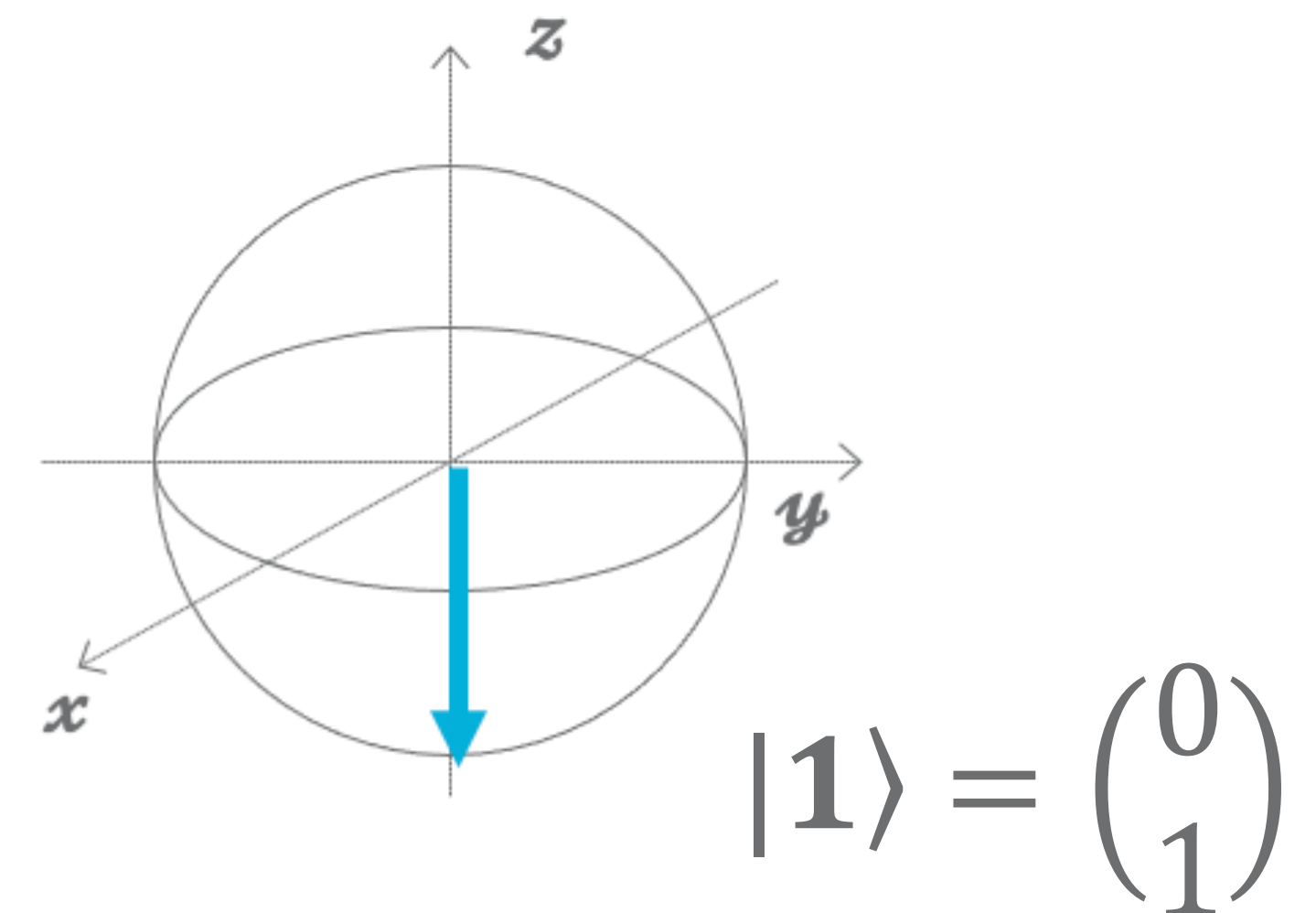
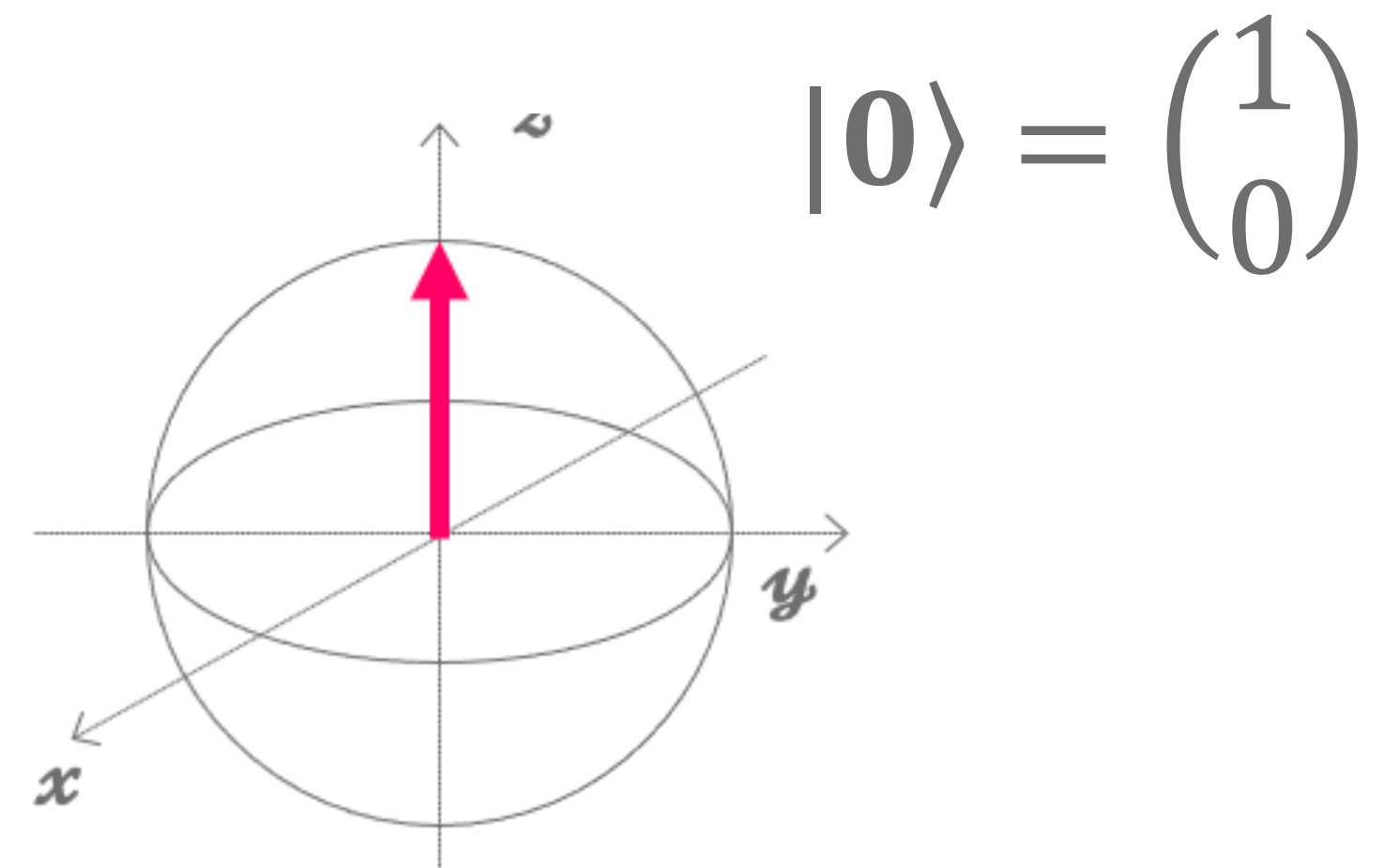
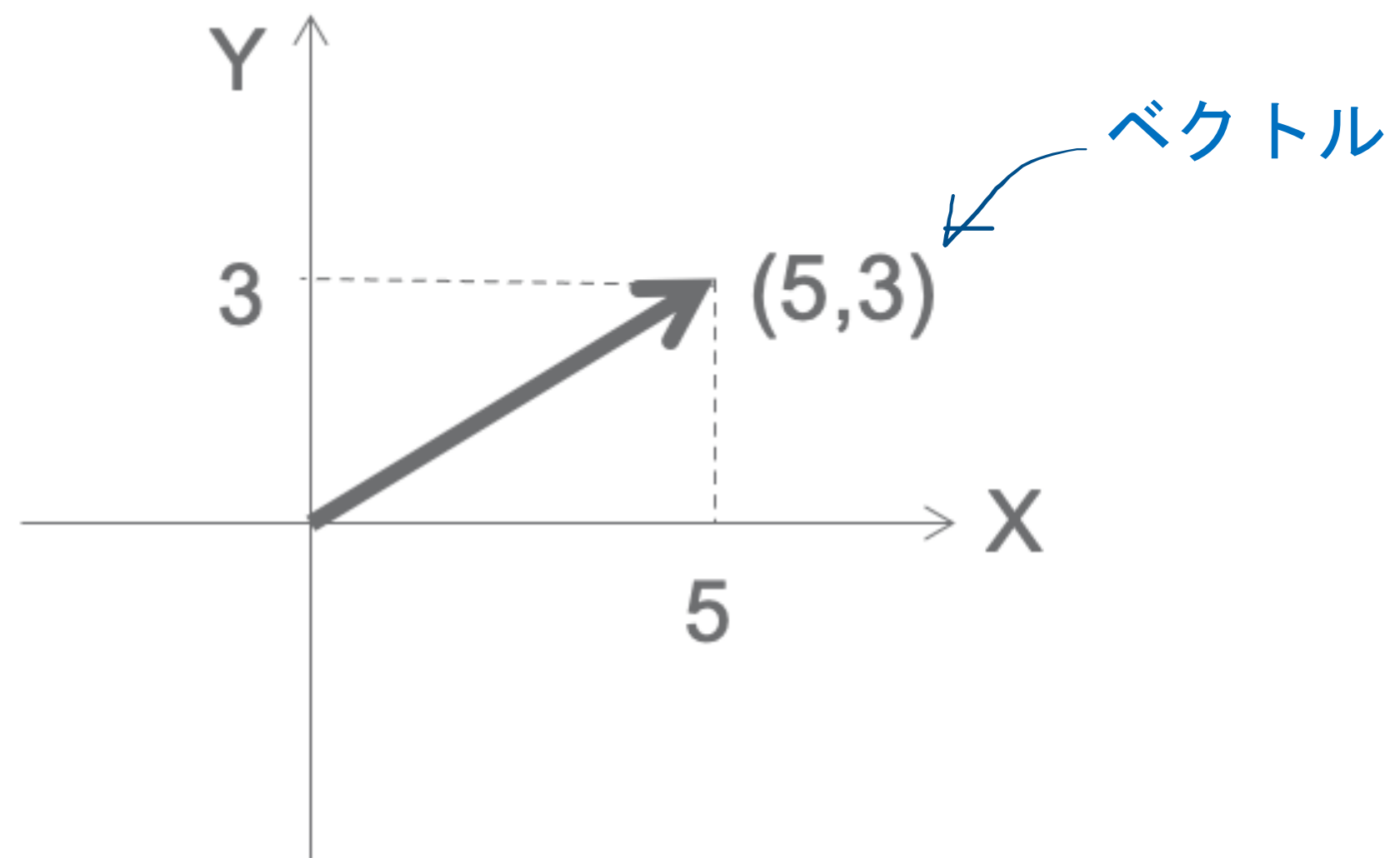
カーネル(類似度)行列



全く違う特徴：0(白)
同じ特徴：1(濃紺/赤)


ベクトルとは

「大きさ」と「向き」を持った量です。

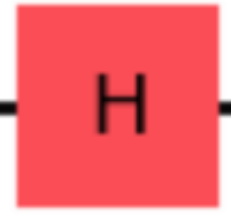


量子計算は、ベクトルと行列の計算

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

q —  —

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

q —  —

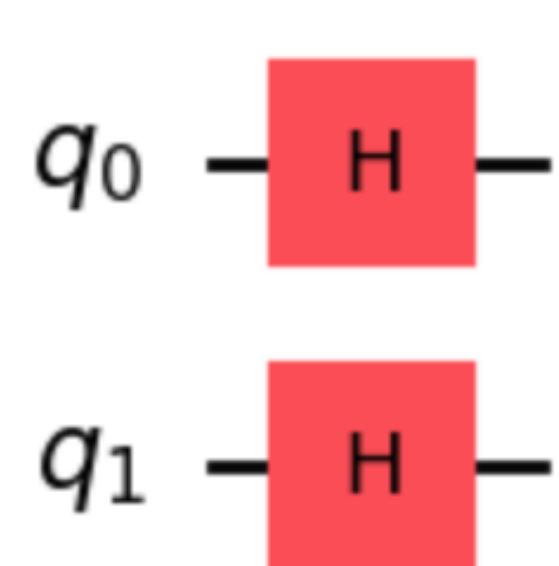
$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

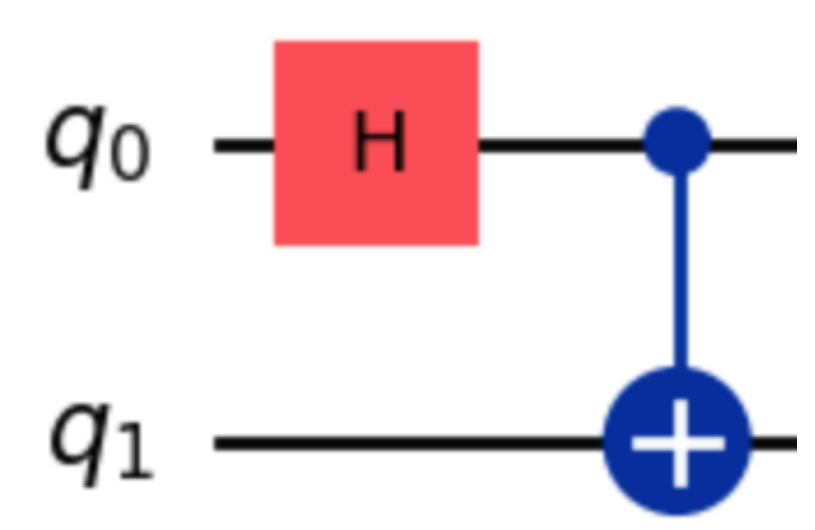
行列と縦ベクトルの積

— 黄色の行と青色の列の成分を1つずつかけて、全てたし合わせて、ベクトルの一つの成分（緑色）となる。

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} a_{11}v_1 + \cdots + a_{1n}v_n \\ \vdots \\ a_{m1}v_1 + \cdots + a_{mn}v_n \end{pmatrix}$$

高度な内容) 量子計算は、ベクトルと行列の計算


$$\begin{aligned} & H|0\rangle \otimes H|0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ &= \frac{1}{2} \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) \end{aligned}$$


$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

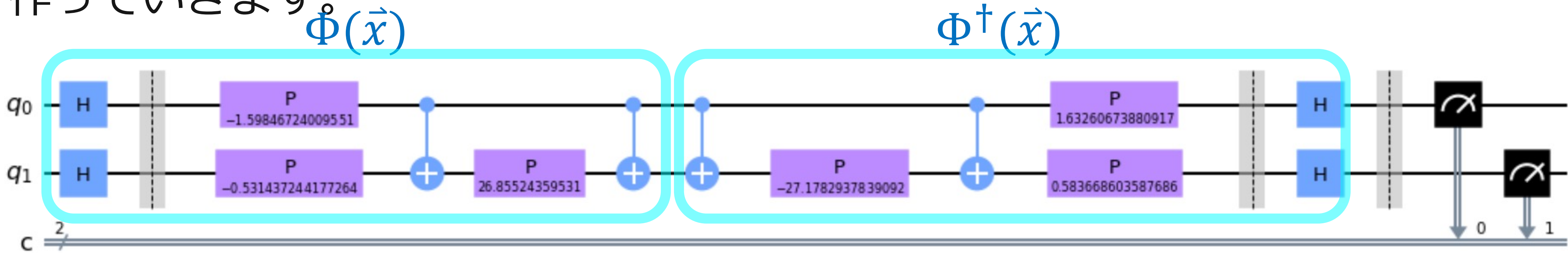
ポイント：
量子状態はベクトル！

ベクトルとベクトルの**テンソル積**：左側のベクトルの成分に右側のベクトルをかける。

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \otimes \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \\ \vdots \\ v_n \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \end{pmatrix}$$

量子カーネル

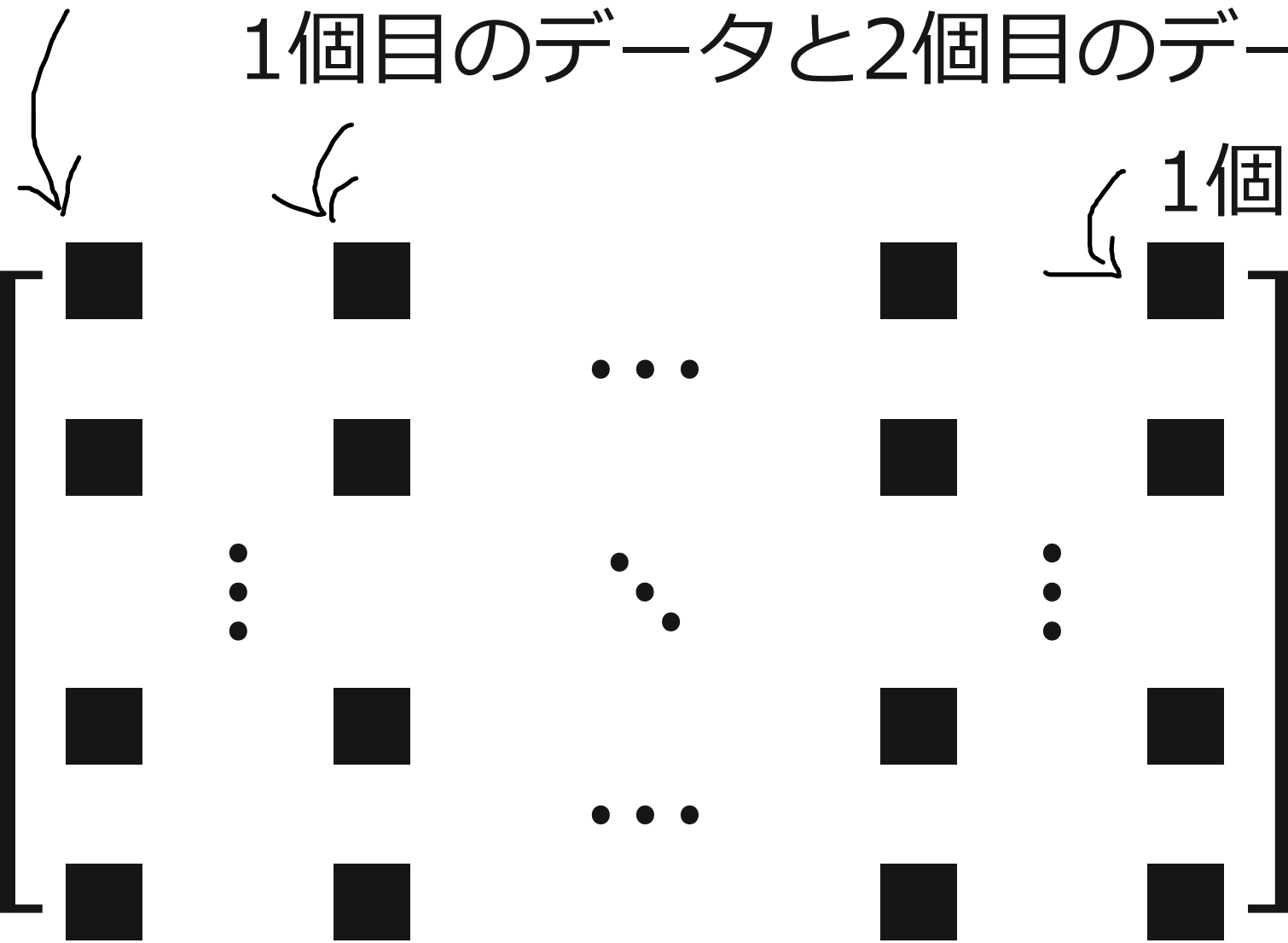
各データ対に対して内積（量子カーネル $\langle \Phi(\vec{x}) | \Phi(\vec{x}) \rangle$ ）を計算、測定してカーネル行列を作っていきます。



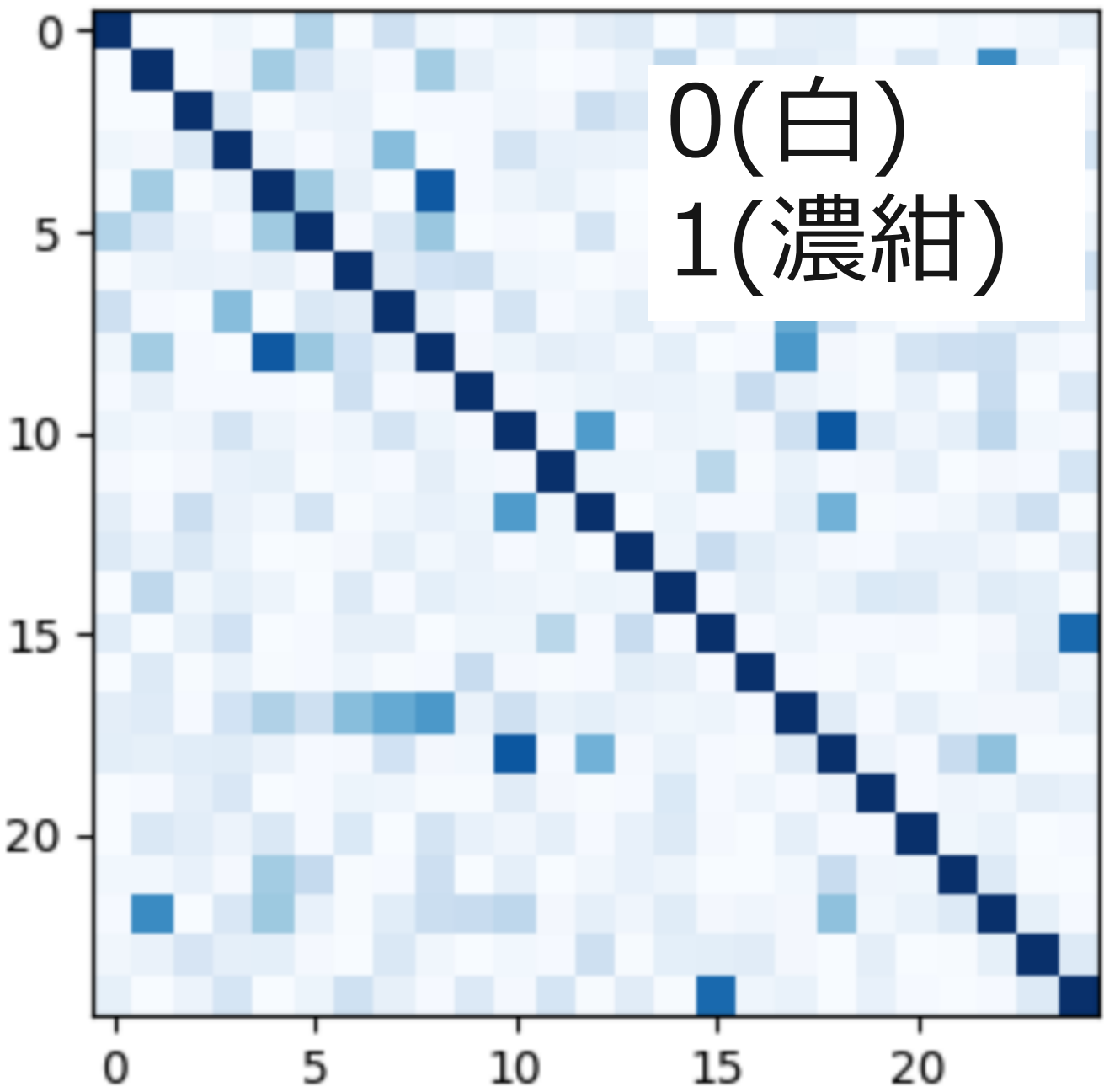
1個目のデータと1個目のデータの内積

1個目のデータと2個目のデータの内積

1個目のデータとn個目のデータの内積

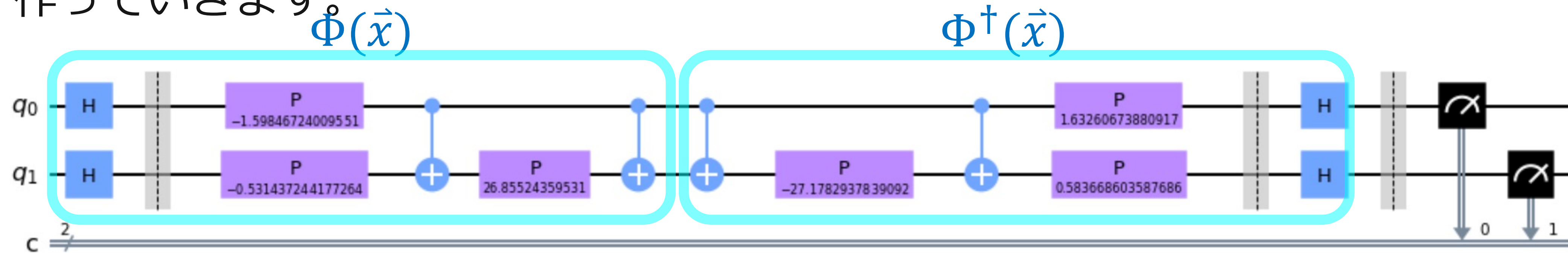


n個目のデータとn個目のデータの内積



量子カーネル

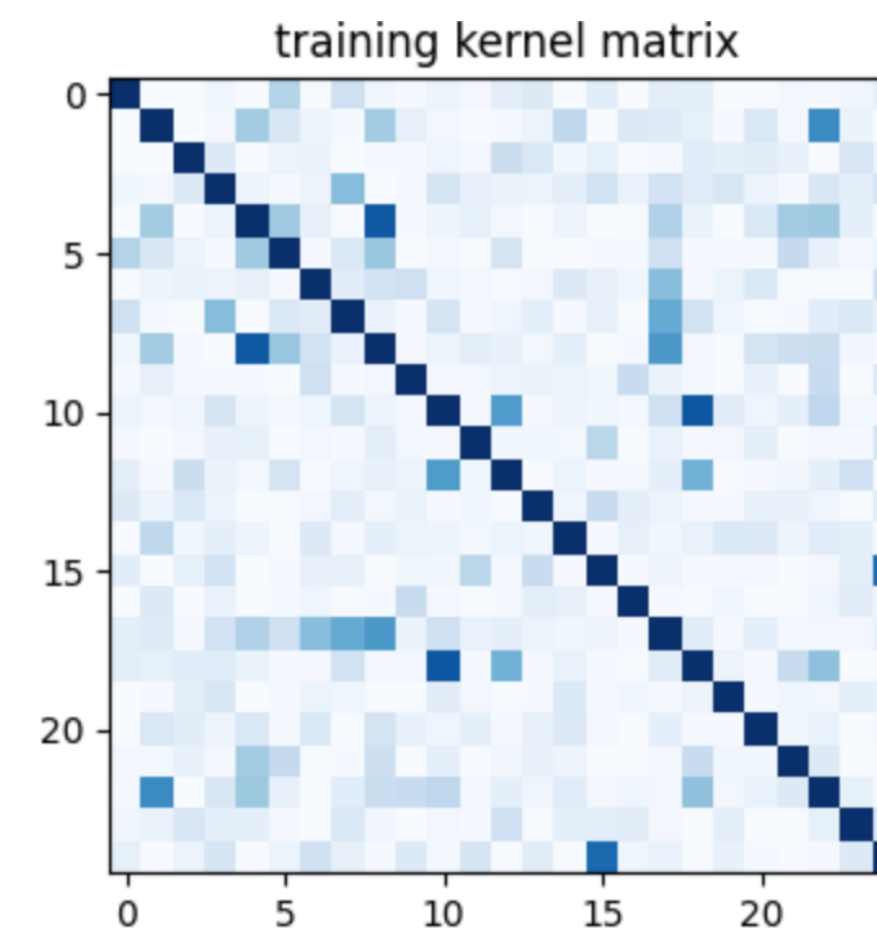
各データ対に対して内積（量子カーネル $\langle \Phi(\vec{x}) | \Phi(\vec{x}) \rangle$ ）を計算、測定してカーネル行列を作っていきます。



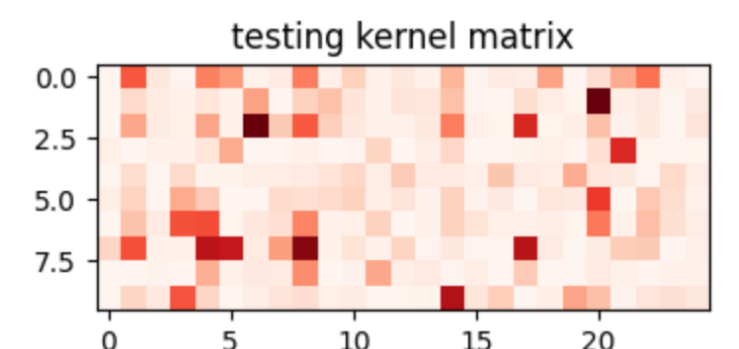
今回は、25個の学習データと10個のテストデータに対して、以下を計算します：

- 学習データ同士（例：25x25の行列）
- 学習データとテストデータ(例：25x10の行列)

25x25の行列

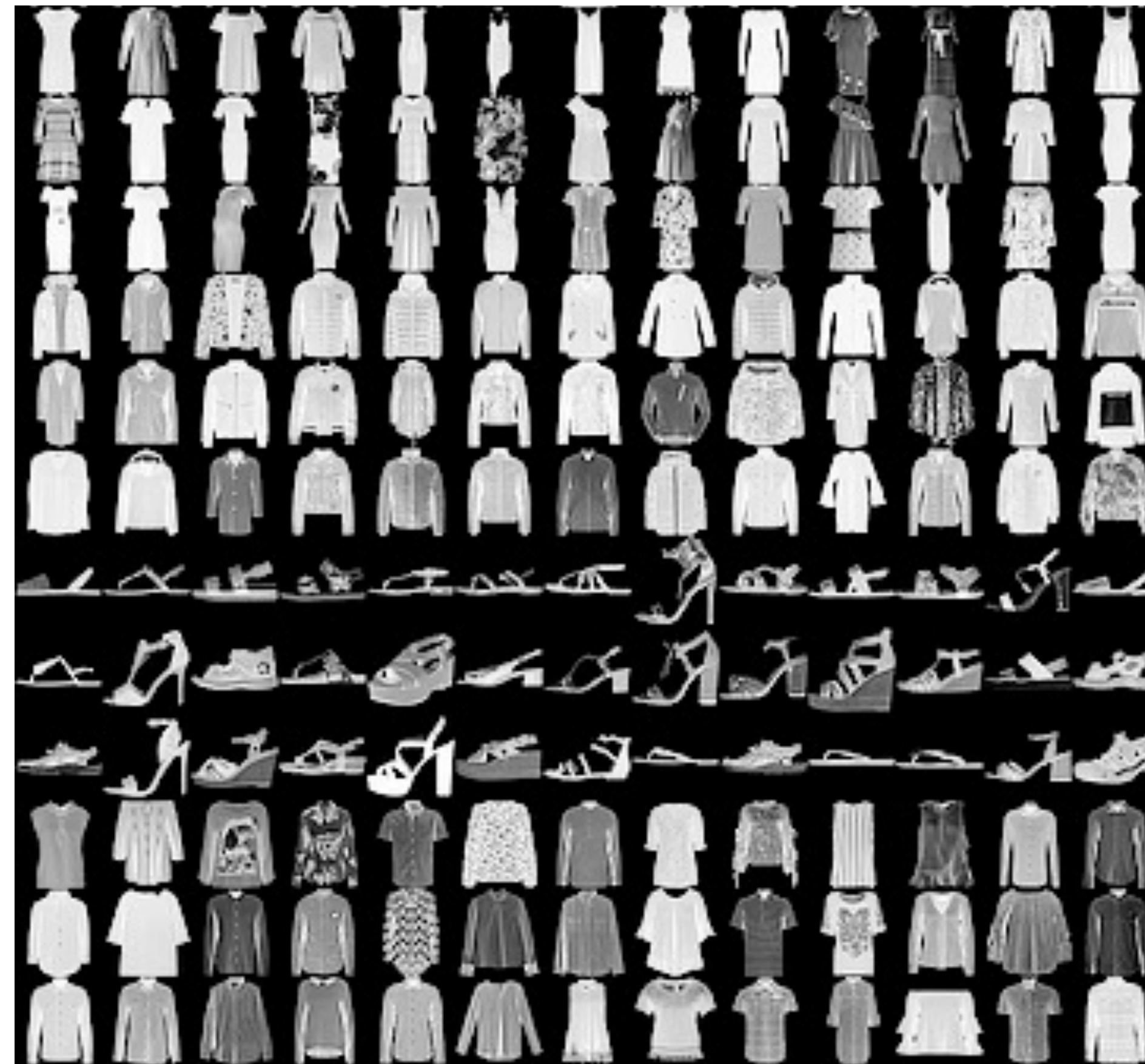
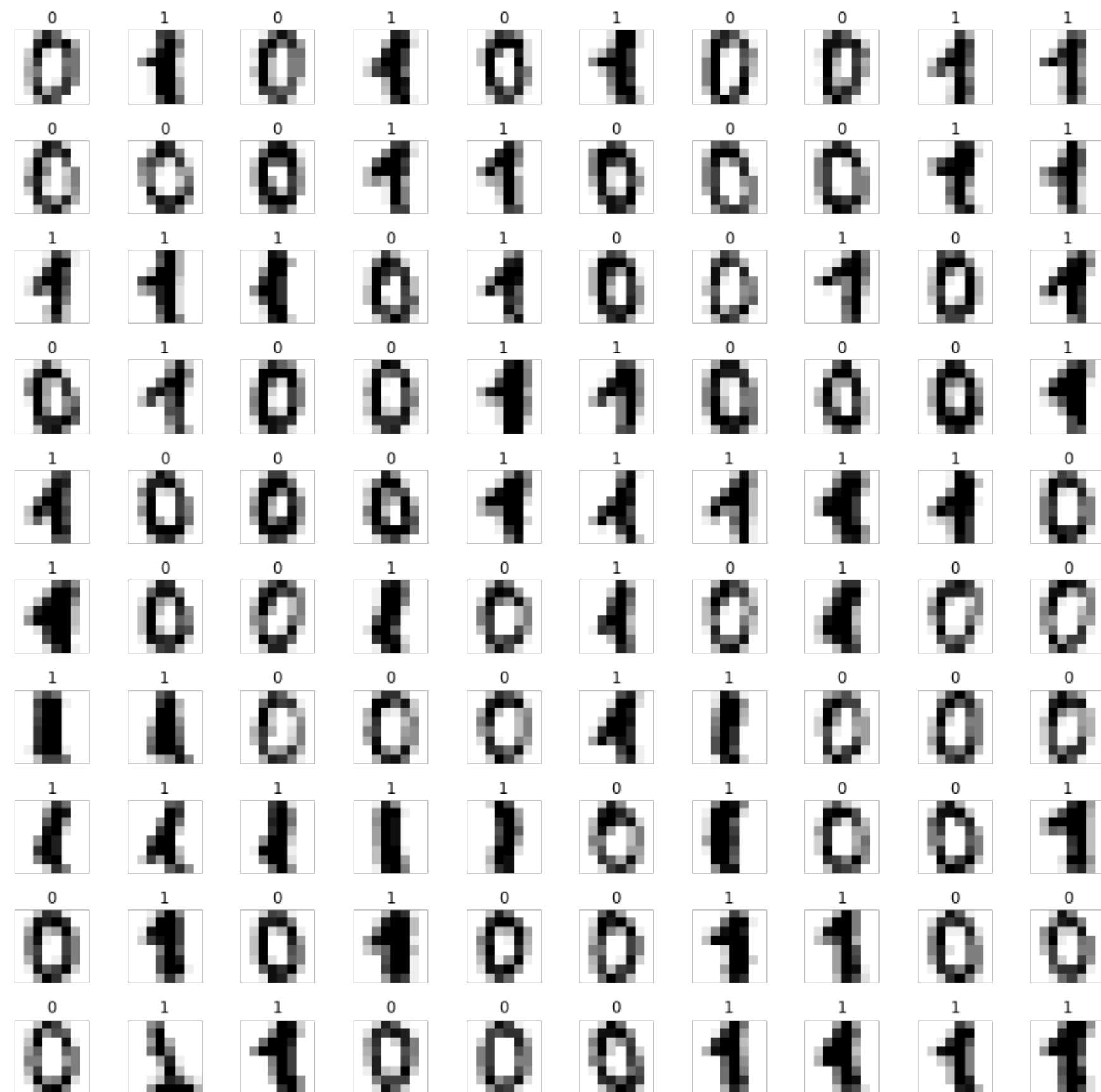


25x10の行列



演習

手書き文字(数字)データで量子カーネルを使った機械学習を学んだ後、洋服の画像について、学習分類を行ってみます。



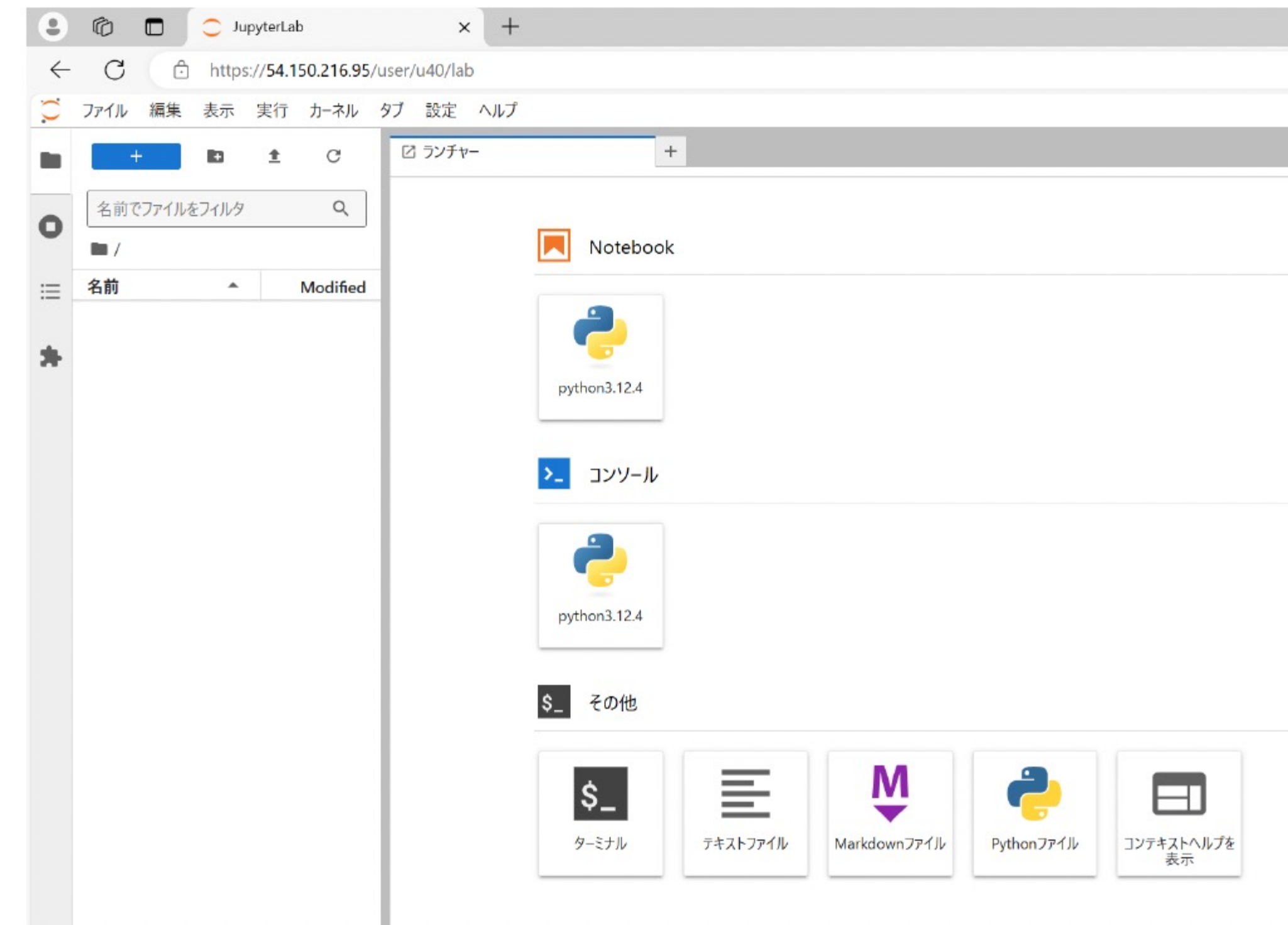
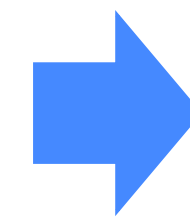
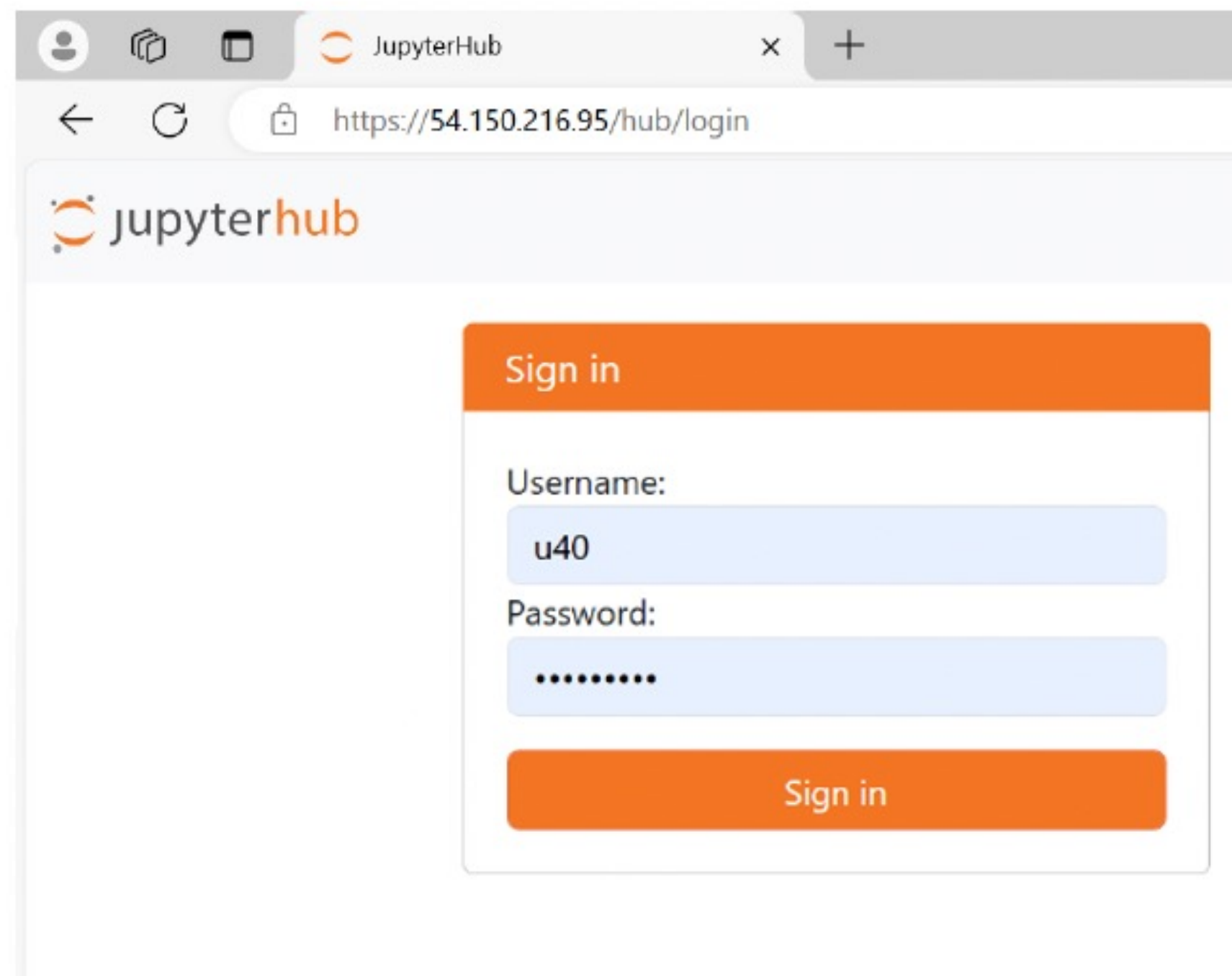
JupyterHubでの実行

(1) Webブラウザ（ Edge、Safari、Chrome、Firefoxなど）で
<https://54.150.216.95>にログイン。



(2) ユーザ名とパスワード（メールで配布）を
入力して、「Sign in」をクリック。

(3) この画面になったら成功です！



Kawasaki Campが終わった後、Qiskitを実行する場合

(1) Google Colabratory (<https://colab.research.google.com/>) を使う。
毎回、以下のコマンドを最初に実行する必要があります。

```
!pip install qiskit qiskit[visualization] qiskit-ibm-runtime qiskit-aer
```

```
!pip install qiskit-algorithms qiskit-nature scikit-learn
```

```
!pip install --prefer-binary pyscf
```

参照ブログ : <https://qiita.com/kifumi/private/51a5d2a420e6318f78fb>



(2) qBraid (<https://www.qbraid.com>) を使う。
実機で実行する場合は、以下のコマンドを実行する必要があります。

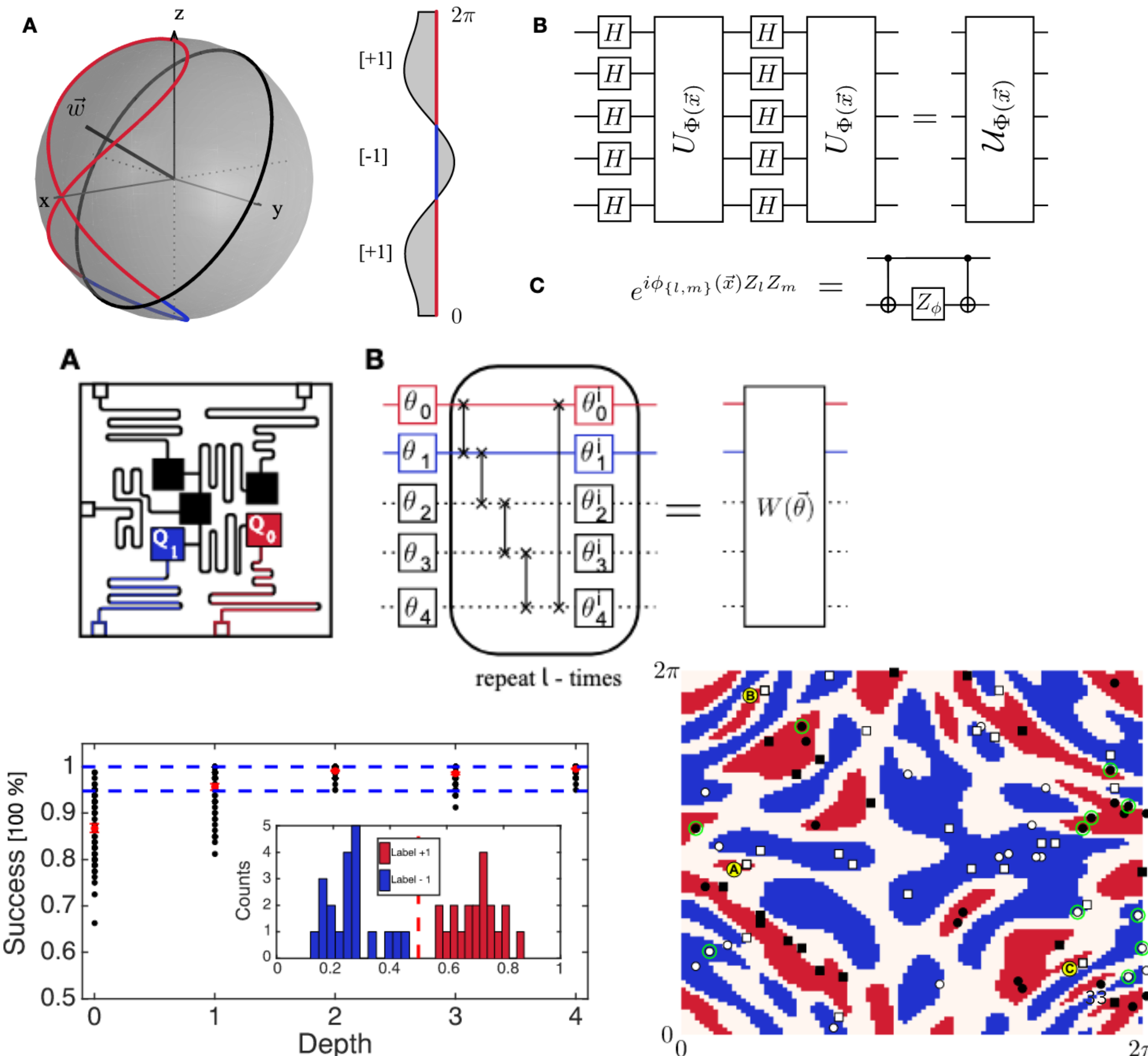
```
%pip uninstall --yes simplejson
```




Letter | Published: 13 March 2019

Supervised learning with quantum-enhanced feature spaces

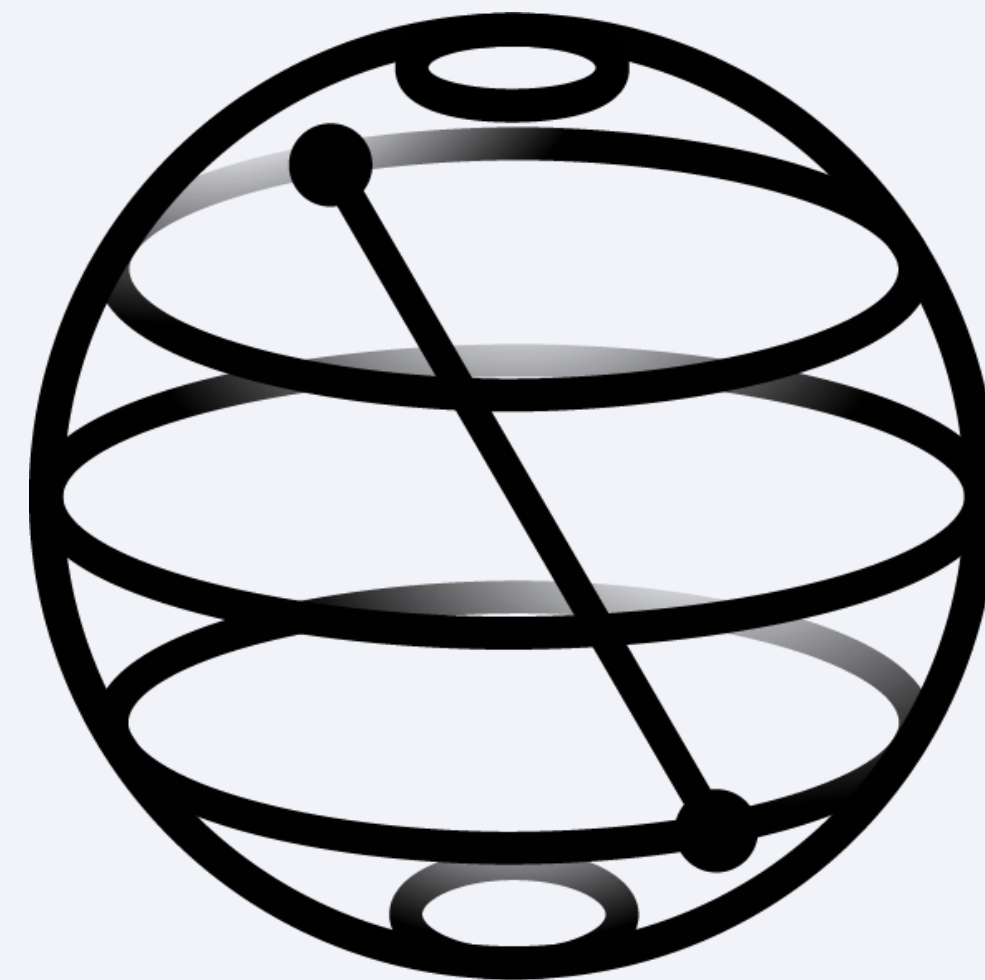
Vojtěch Havlíček, Antonio D. Córcoles , Kristan Temme , Aram W. Harrow, Abhinav Kandala, Jerry M. Chow & Jay M. Gambetta






量子カーネルSVMの参考文献

量子カーネルSVM：最も普及している量子カーネルアルゴリズム。

- Yunchao Liu, Srinivasan Arunachalam and Kristan Temme, *A rigorous and robust quantum speed-up in supervised machine learning* (2020), [arXiv:2010.02174](https://arxiv.org/abs/2010.02174).
量子カーネルSVMが特定の入力データクラスに対して従来の方法よりも高速化することを証明。
- Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven and Jarrod R. McClean, *Power of data in quantum machine learning* (2020), [arXiv:2011.01938](https://arxiv.org/abs/2011.01938).
量子カーネルSVMを使用して、量子機械学習アルゴリズムのデータの計算能力を定量化し、量子モデルが従来のモデルを上回ることができる条件を提示。
- Lloyd, Seth, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran, *Quantum embeddings for machine learning* (2020), [arXiv preprint arXiv:2001.03622](https://arxiv.org/abs/2001.03622)
量子カーネルアラインメント(最適化できるパラメーターを持つ構造)を可能にする量子メトリック学習と呼ばれる手法を提示。



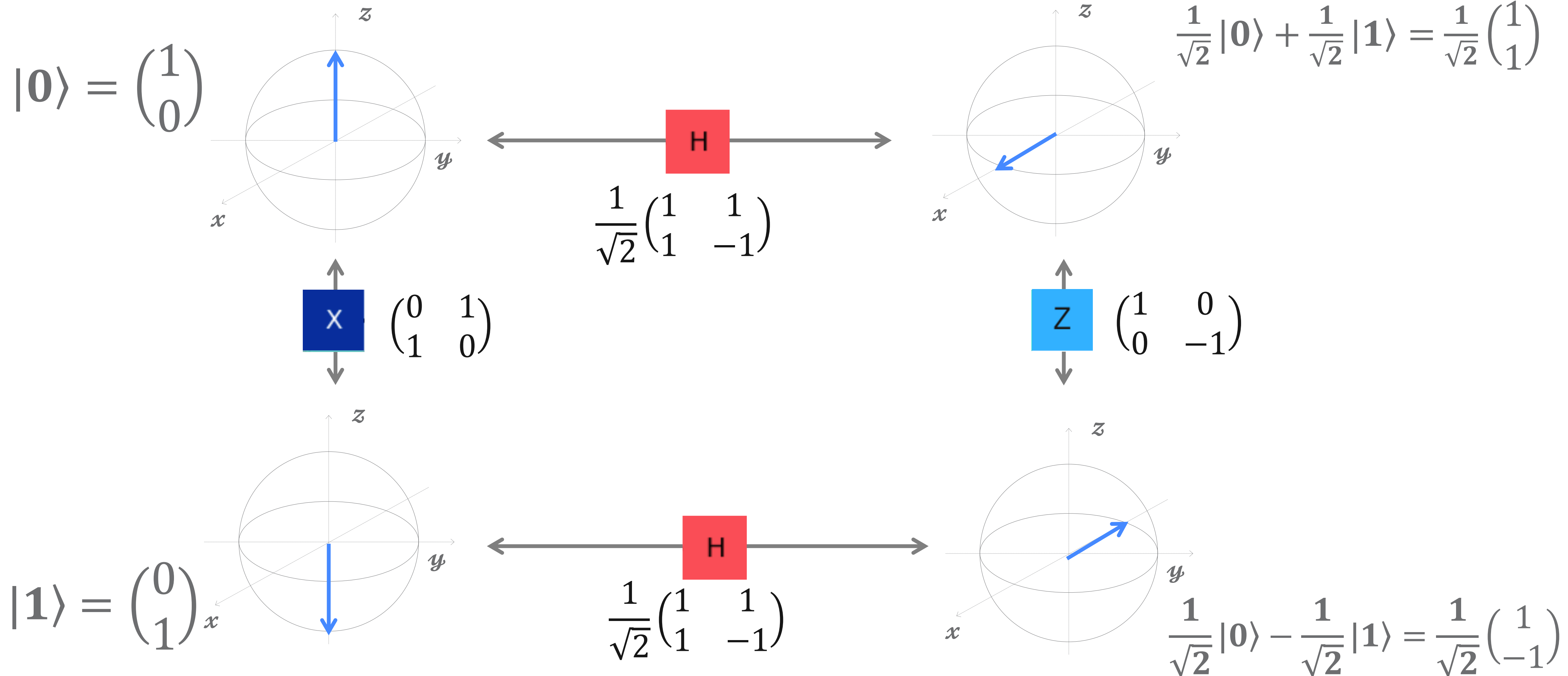
量子ビットとベクトル

	量子	ベクトル・行列表示
ビット	0と1の重ね合わせ $\alpha \times 0\rangle + \beta \times 1\rangle$	$ 0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, 1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $ \psi\rangle = \alpha 0\rangle + \beta 1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ α, β : 確率振幅 $ \alpha ^2 + \beta ^2 = 1$
論理ゲート	X, H, CNOT など   	$\begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{21} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix}$ <div>量子ゲート 入力ビット 出力ビット</div>





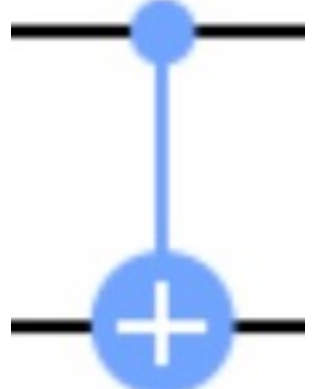
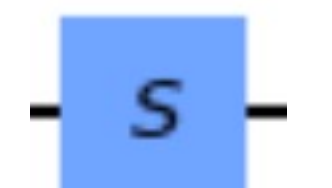
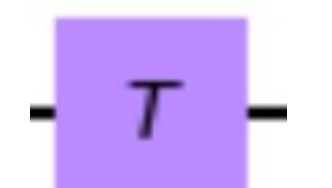
例) Xゲートの場合

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

量子計算は、行列の計算



代表的な量子ゲート一覧

	行列表現	Qiskitのコード	回路表示	
Xゲート	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	x		ビット反転
Yゲート	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	y		位相・ビット反転
Zゲート	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	z		位相反転
Hゲート	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	h		重ね合わせを作る
CNOTゲート	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	cx		量子もつれを作る
Sゲート	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	s		pi/2位相シフト
Tゲート	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	t		pi/4位相シフト