（1/31時点）

3.7 k近傍法：怠惰学習アルゴリズム

2018.2.2 Python機械学習プログラミング勉強会#4

今日の内容

- k近傍法
- kaggleタイタニックチュートリアル（k近傍法を用いた欠損値の推測など）

k近傍法

（k-nearest neighbor classifier)

（KNN）

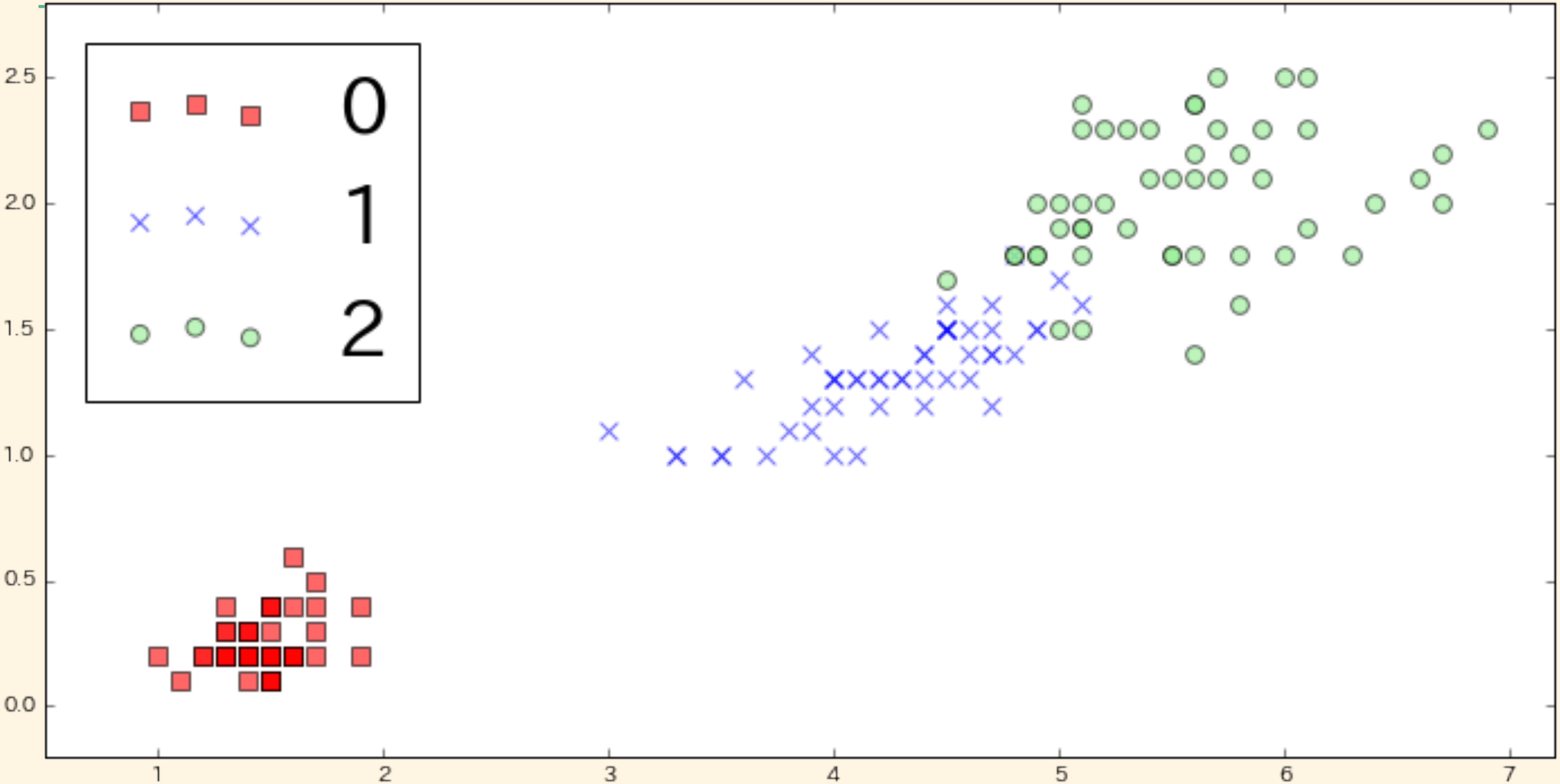
k近傍法

- k近傍法の発想は、

  「距離が近ければ、似たもの同士でしょ！」

  という発想

例 irisのいつものデータでk-means



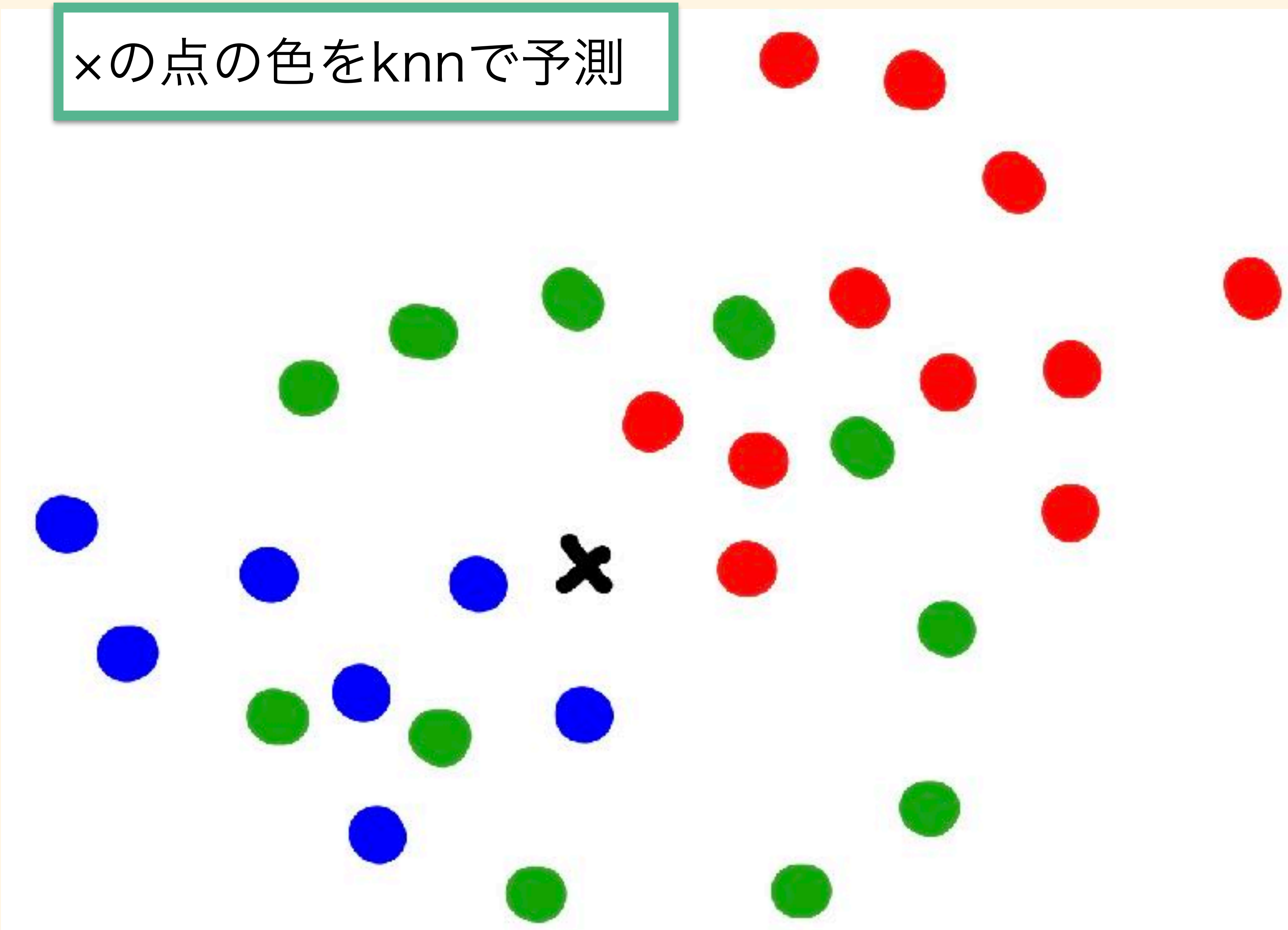
k=5のkNN → V ページ、91ページ

k-近傍法とは(90頁)

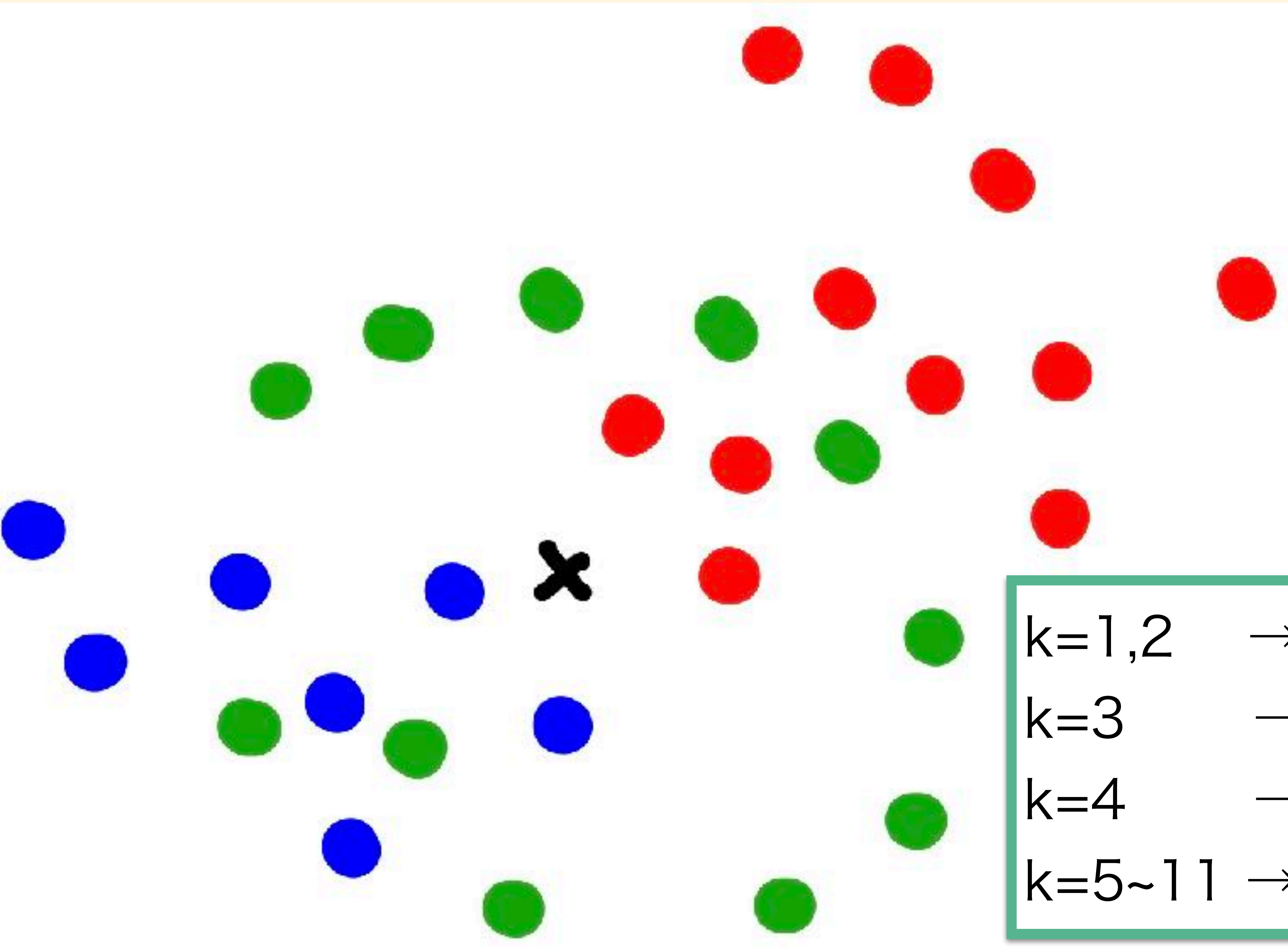
- トレーニングデータセットのサンプルの中から、選択された距離指標に基づき、分類したいデータ点に最も近いk個ののサンプルを見つけ出す。

- 新しいデータ点のクラスラベルを多数決で決める。

例 kによって結果はかわる



例 kによって結果はかわる



k=1,2 → 青
k=3 → 赤
k=4 → 青
k=5~11 → 緑

最近傍法

- $k = 1$ の時のkNNを最近傍法という。

輪読本89~92頁

- KNNは怠惰学習の代表的な例
- KNNはノンパラメトリックモデル、学習過程のコスト０（学習しない）

KNNのアルゴリズム(89頁)

- kの値と距離指標を選択
- 分類したいサンプルからk個の最近傍のデータを見つけ出す
- 多数決によりクラスラベルを割り当てる

KNNのメリデメ(90頁)

- 新しいトレーニングデータを集めるとすぐに分類器が適応
- 計算量がトレーニングデータセットのサンプル個数に比例して増加（最悪の場合）
- トレーニングサンプルを破棄しないので、記憶域が必要

目次

- 数式で、kNNの分類規則表す
- 最近傍法の精度は？
- 特徴量が多い場合使えるか（次元の呪い）
- 計算量が多い？効率的に計算する方法は？

最近傍法（はじパタ54頁）

最近傍法の識別規則

- C_i : クラス（i = 1 , ... , K）
- $\Omega = \{C_1, \ldots, C_K\}$: K個の集合のクラス
- $N(i)$: i番目のクラスの学習データ数
- $S_i = \{x_1^{(i)}, , x_{N(i)}^{(i)}\}$:学習データの集合
- $d(x, x_j^{(i)}) = \|x - x_j^{(i)}\|$: 入力データxと学習データ$x_j^{(i)}$のユークリッド距離

識別規則：

$$Class = \begin{cases} \arg\min_i \{\min_j d(x, x_j^{(i)})\} & if \min_{i,j} d(x, x_j^{(i)}) < t \\ \text{リジェクト} & if otherwise \end{cases}$$

最近傍法（はじパタ54頁）

最近傍法の識別規則

- C_i : クラス（i = 1 , ... , K）
- $\Omega = \{C_1, \ldots, C_K\}$: K個の集合のクラス
- $N(i)$: i番目のクラスの学習データ数
- $S_i = \{x_1^{(i)},, x_{N(i)}^{(i)}\}$:学習データの集合
- $d(x, x_j^{(i)}) = \|x - x_j^{(i)}\|$: 入力データxと学習データ$x_j^{(i)}$のユークリッド距離

識別規則：

各クラスから最小のサンプルをとってきて

$$Class = \begin{cases} \arg\min_i \{\min_j d(x, x_j^{(i)})\} & if\ \min_{i,j} d(x, x_j^{(i)}) < t \\ \text{リジェクト} & if otherwise \end{cases}$$

最近傍法（はじパタ54頁）

最近傍法の識別規則

- C_i : クラス（i = 1 , ... , K）
- $\Omega = \{C_1, \ldots, C_K\}$: K個の集合のクラス
- $N(i)$: i番目のクラスの学習データ数
- $S_i = \{x_1^{(i)},, x_{N(i)}^{(i)}\}$:学習データの集合
- $d(x, x_j^{(i)}) = \|x - x_j^{(i)}\|$: 入力データxと学習データ$x_j^{(i)}$のユークリッド距離

識別規則：

その中で、最小のi(クラス)を返す

$$Class = \begin{cases} \arg\min_i \{\min_j d(x, x_j^{(i)})\} & if \min_{i,j} d(x, x_j^{(i)}) < t \\ \text{リジェクト} & if otherwise \end{cases}$$

最近傍法（はじパタ54頁）

最近傍法の識別規則

- C_i : クラス（i = 1 , ... , K）
- $\Omega = \{C_1, \ldots, C_K\}$: K個の集合のクラス
- $N(i)$: i番目のクラスの学習データ数
- $S_i = \{x_1^{(i)},, x_{N(i)}^{(i)}\}$:学習データの集合
- $d(x, x_j^{(i)}) = \|x - x_j^{(i)}\|$: 入力データxと学習データ$x_j^{(i)}$のユークリッド距離

識別規則：

$$Class = \begin{cases} \arg\min_i \{\min_j d(x, x_j^{(i)})\} & \mathit{if} \min_{i,j} d(x, x_j^{(i)}) < t \\ \text{リジェクト} & \mathit{if otherwise} \end{cases}$$

距離がt以上であれば判断をさける

kNN法（はじパタ59頁）

- $\mathcal{T}_N = \{x_1, \ldots, x_N\}$: 鋳型の集合
- $\Omega = \{C_1, \ldots, C_K\}$: 鋳型が所属するクラスの集合
- $w_i \in \Omega$: i番目の鋳型が所属するクラス
- $k(x) = \{x_{1i}, \ldots, x_{ik}\}$:入力xに最も近いk個の鋳型の集合
- k_j : クラスjに属する学習データの数、$k = k_1 + \ldots + k_K$

識別規則：

$$識別クラス = \begin{cases} j & if \;\; k_j = max\{k_1, \ldots, k_K\} \\ リジェクト & if \{k_i, \ldots, k_j\} = max\{k_1, \ldots, k_K\} \end{cases}$$

kNN法（はじパタ59頁）

- $\mathcal{T}_N = \{x_1, \ldots, x_N\}$: 鋳型の集合
- $\Omega = \{C_1, \ldots, C_K\}$: 鋳型が所属するクラスの集合
- $w_i \in \Omega$: i番目の鋳型が所属するクラス
- $k(x) = \{x_{1i}, \ldots, x_{ik}\}$:入力xに最も近いk個の鋳型の集合
- k_j : クラスjに属する学習データの数、$k = k_1 + \ldots + k_K$

識別規則：

$$\text{識別クラス} = \begin{cases} j & \text{if} \;\; k_j = max\{k_1, \ldots, k_K\} \\ \text{リジェクト} & \text{if} \{k_i, \ldots, k_j\} = max\{k_1, \ldots, k_K\} \end{cases}$$

1.入力xに近いk個の鋳型を取ってきて、k個の鋳型のクラスの多数決で、入力xのクラスを決める
2.多数決が同数の場合は、判断を保留する

kNN法（はじパタ59頁）

- $\mathcal{T}_N = \{x_1, \ldots, x_N\}$: 鋳型の集合

scikit-learnの場合は、多数決が同数の場合(以下の2行目の場合)
1. サンプルまでの距離がより近いものが優先される
2. 1.で決まらない場合は、最初に現れるクラスラベルが選択される

識別規則：

$$
識別クラス = \begin{cases} j & if\ \ k_j = max\{k_1, \ldots, k_K\} \\ リジェクト & if\{k_i, \ldots, k_j\} = max\{k_1, \ldots, k_K\} \end{cases}
$$

1.入力xに近いk個の鋳型を取ってきて、k個の鋳型のクラスの多数決で、入力xのクラスを決める
2.多数決が同数の場合は、判断を保留する

目次

- 数式で、kNNの分類規則表す
- 最近傍法の精度は？
- 特徴量が多い場合使えるか（次元の呪い）
- 計算量が多い？効率的に計算する方法は？

kNN法とベイズ誤り率(はじパタ62頁)

$\hat{\varepsilon}^*$ ：ベイズ誤り率（後述）

ε_{kNN} : kNNの誤り率

x_{1NN} ：入力xの最近傍鋳型

$\mathcal{T}_N$ ：N個の鋳型の集合

漸近仮定：$\lim_{N\to\infty} \mathcal{T}_N \Rightarrow d(x, x_{1NN}) \to 0$が成り立てば

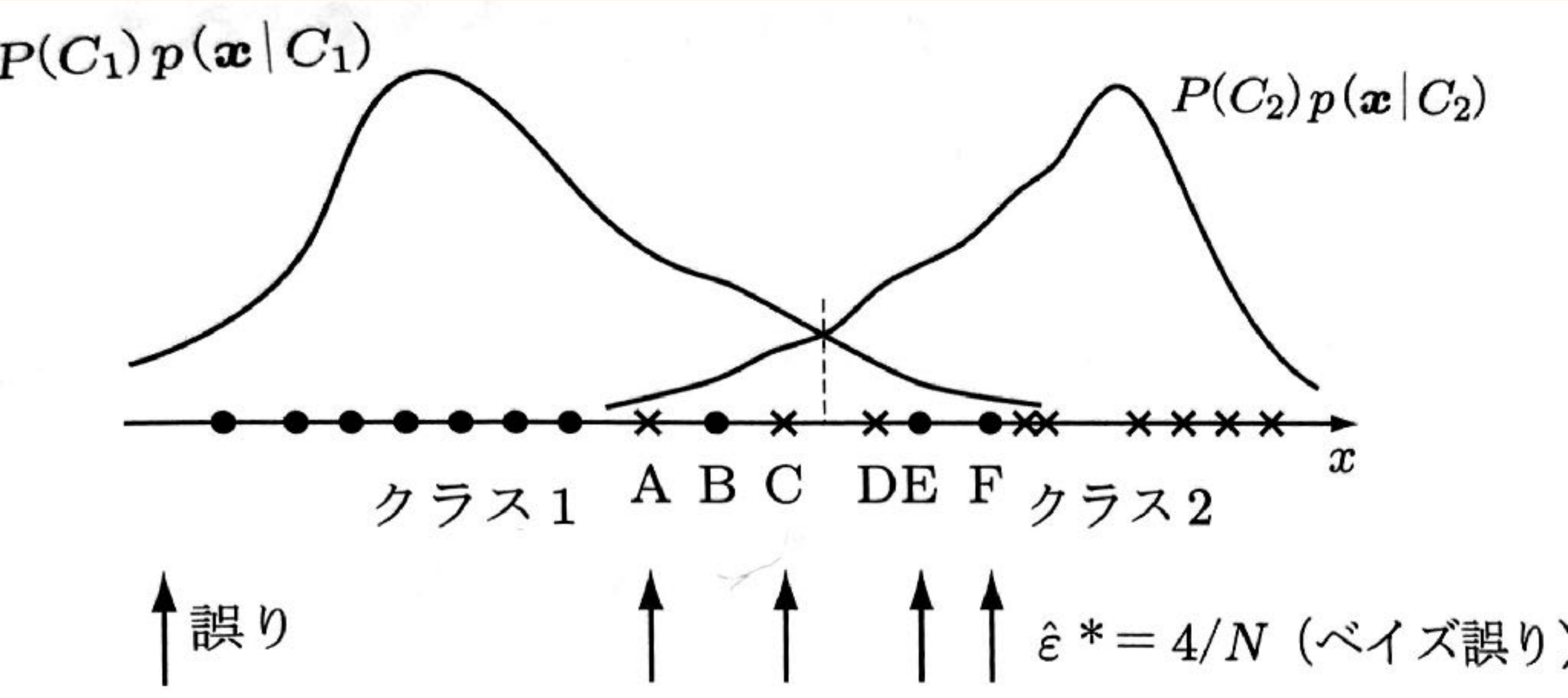
$$\frac{1}{2}\varepsilon^* \leq \varepsilon_{2NN} \leq \varepsilon_{4NN} \leq \cdots \leq \varepsilon^* \leq \cdots \leq \varepsilon_{3NN} \leq \varepsilon_{1NN} \leq 2\varepsilon^*$$

が成立

ベイズ誤り率(はじパタ21~26頁)

- 事後確率が、最も大きなクラスに観測データを分類する（ベイズの識別規則）

- ベイズの識別規則は誤り率が最小となる



- 事後確率が大きい方に分類
- A,C,E,Fは誤り

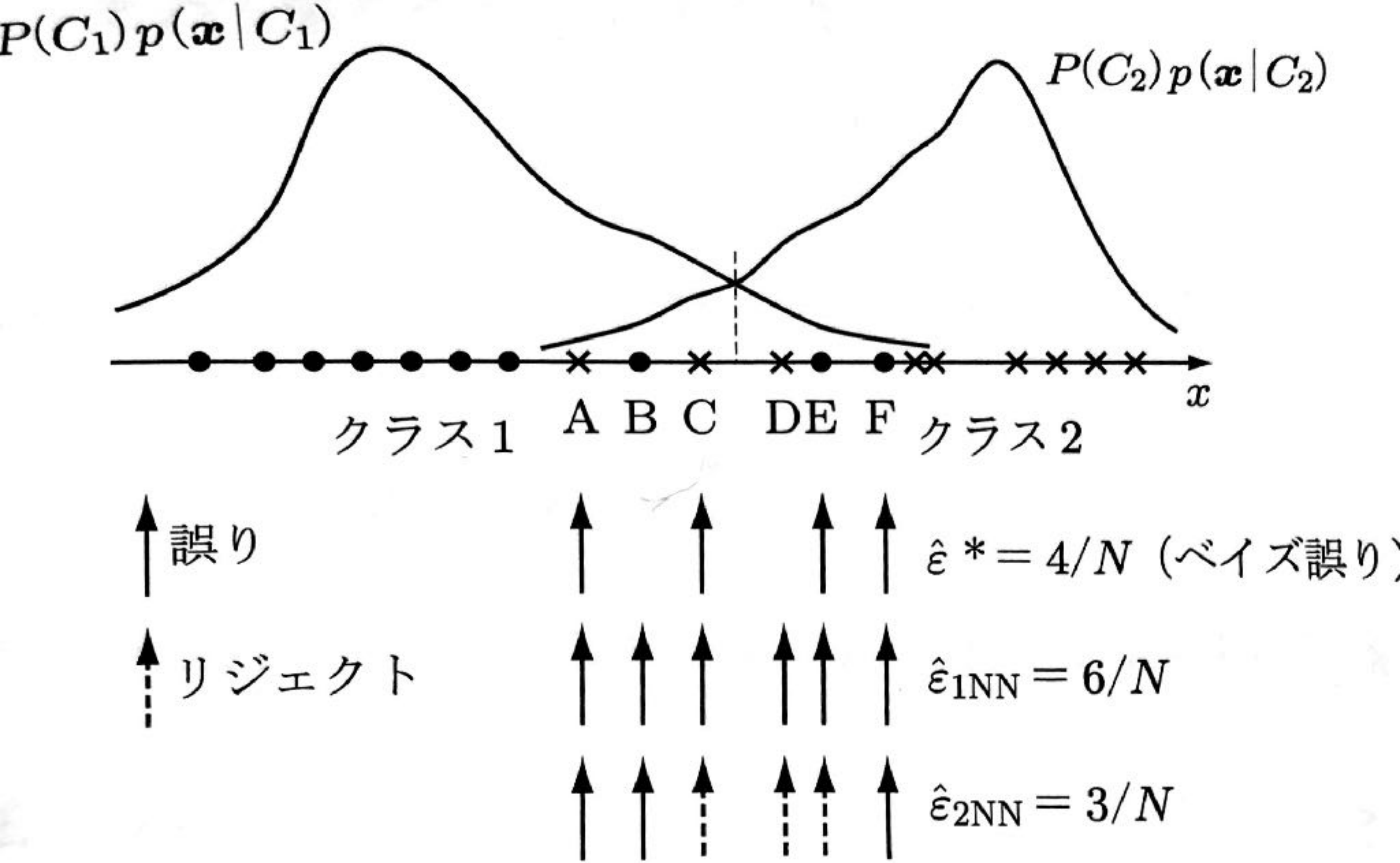


図 5.10　kNN 法による誤りの発生機構

kNN法とベイズ誤り率(はじパタ62頁)

$\hat{\varepsilon}^*$ ：ベイズ誤り率（後述）

ε_{kNN} : kNNの誤り率

x_{1NN} ：入力xの最近傍鋳型

$\mathcal{T}_N$ ：N個の鋳型の集合

漸近仮定：$\lim_{N\to\infty} \mathcal{T}_N \Rightarrow d(x, x_{1NN}) \to 0$が成り立てば

$$\frac{1}{2}\varepsilon^* \le \varepsilon_{2NN} \le \varepsilon_{4NN} \le \cdots \le \varepsilon^* \le \cdots \le \varepsilon_{3NN} \le \varepsilon_{1NN} \le 2\varepsilon^*$$

が成立

目次

- 数式で、kNNの分類規則表す
- 最近傍法の精度は？
- 特徴量が多い場合使えるか（次元の呪い）
- 計算量が多い？効率的に計算する方法は？

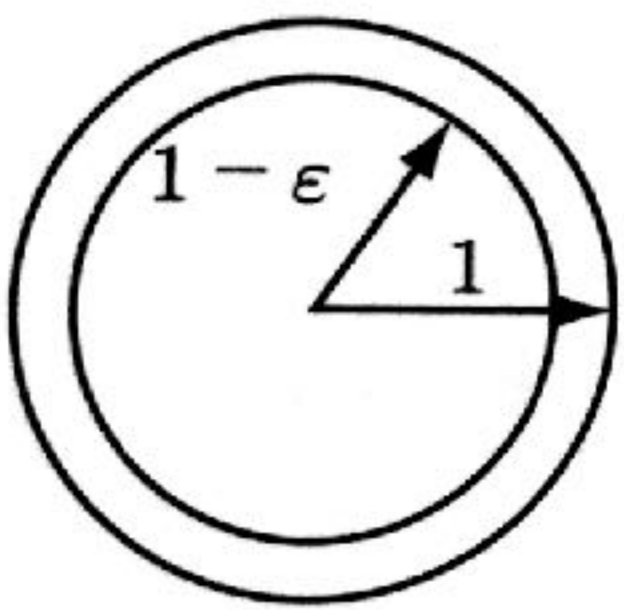
特徴量が多い場合使えるか

- 漸近仮定は成り立つか
- 半径１のd次元超球の体積と厚さε の殻の内部の部分の体積比は

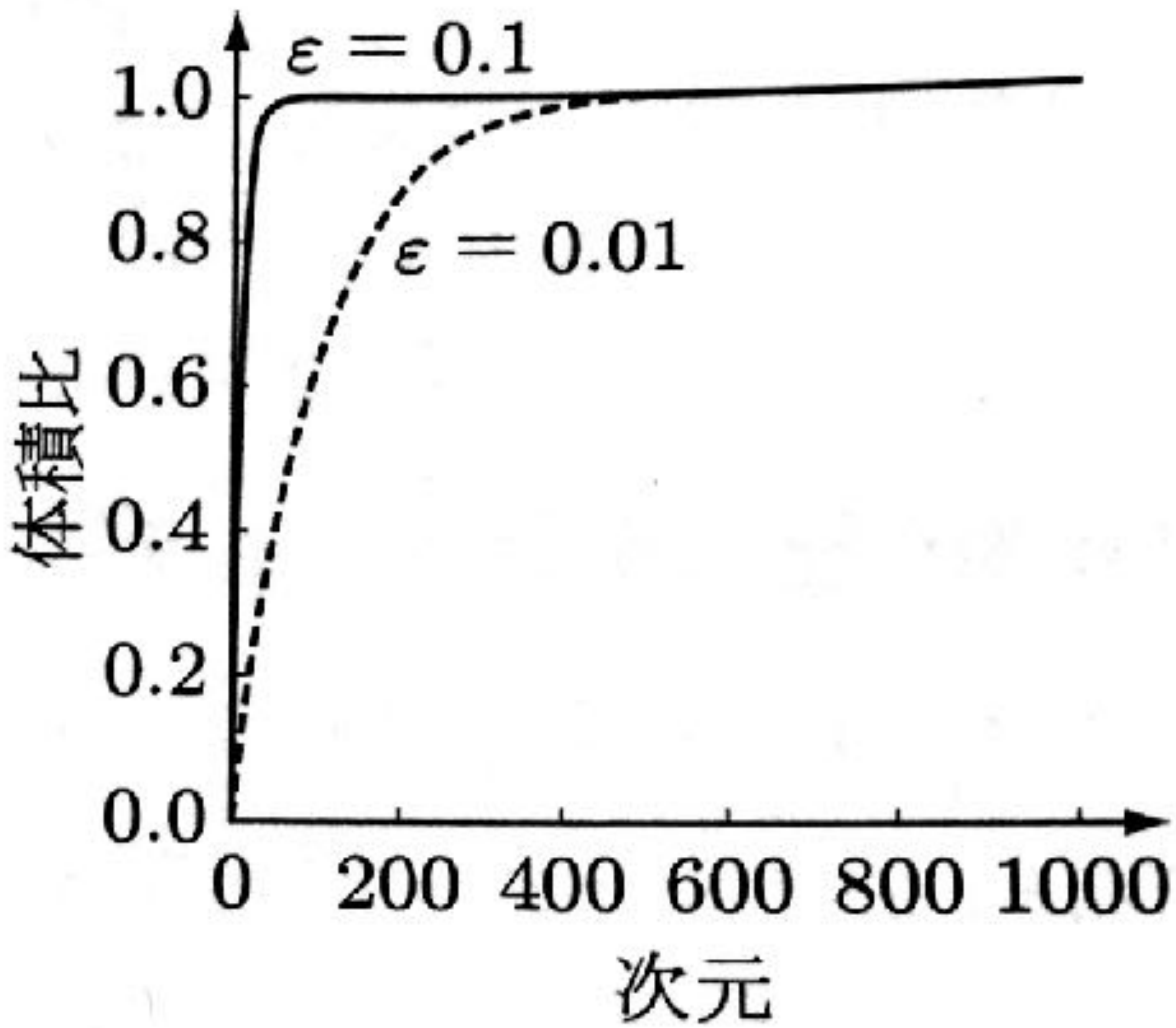
$$\frac{殻部分の体積}{超球の体積} = 1 - (1 - \varepsilon)^d$$

と書ける

$$\frac{\text{殻部分の体積}}{\text{超球の体積}} = 1 - (1 - \varepsilon)^d$$



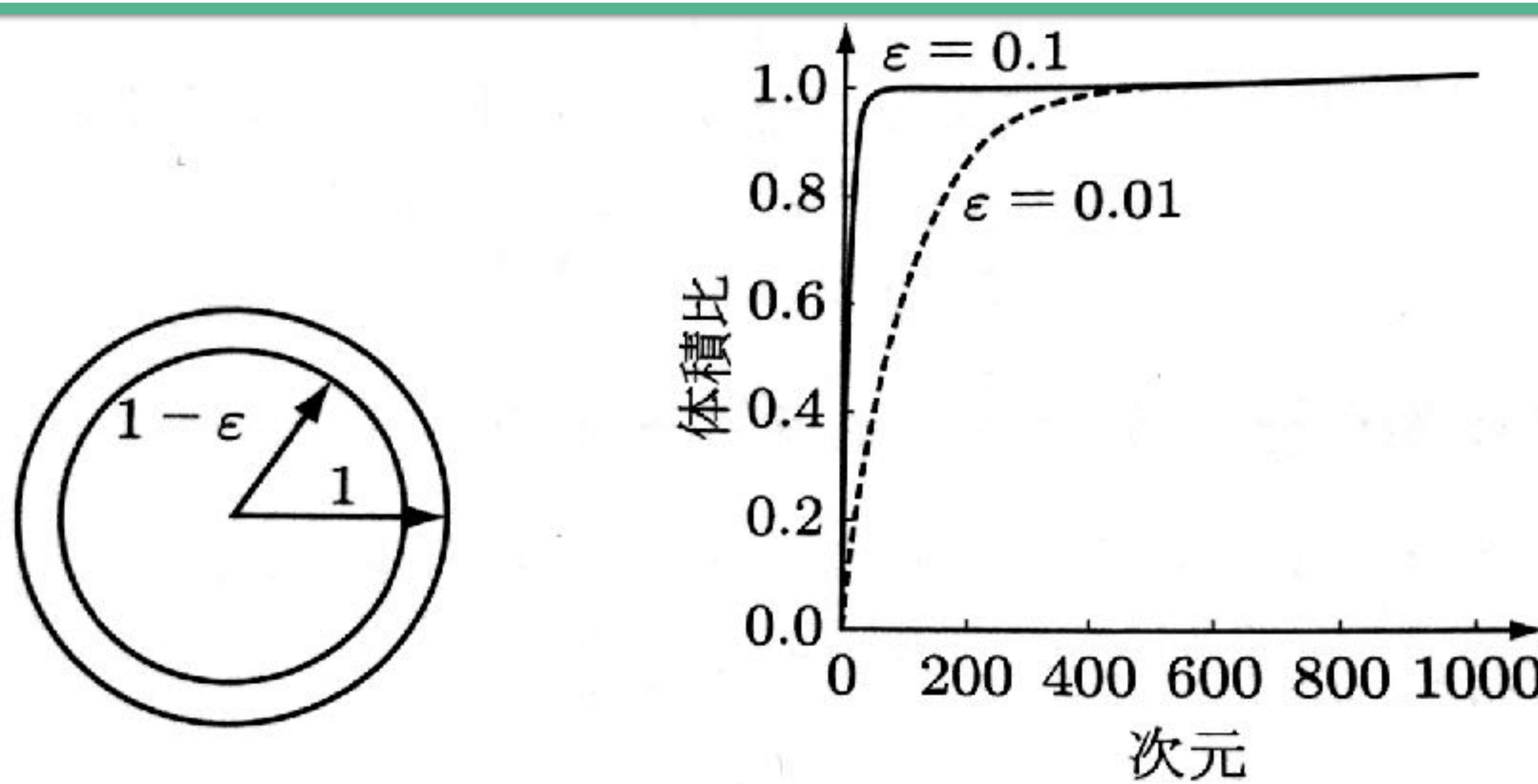
（a）単位超球の殻



（b）殻の部分の体積比

図 5.11　単位超球と殻の部分の体積比

- 次元が大きいと、殻の部分の体積で、ほぼ全てが占められる
- サンプルが、一様に分布していると仮定すると、サンプルはほぼ殻の部分に分布
- 漸近仮定は不成立



（a）単位超球の殻　（b）殻の部分の体積比

図 5.11　単位超球と殻の部分の体積比