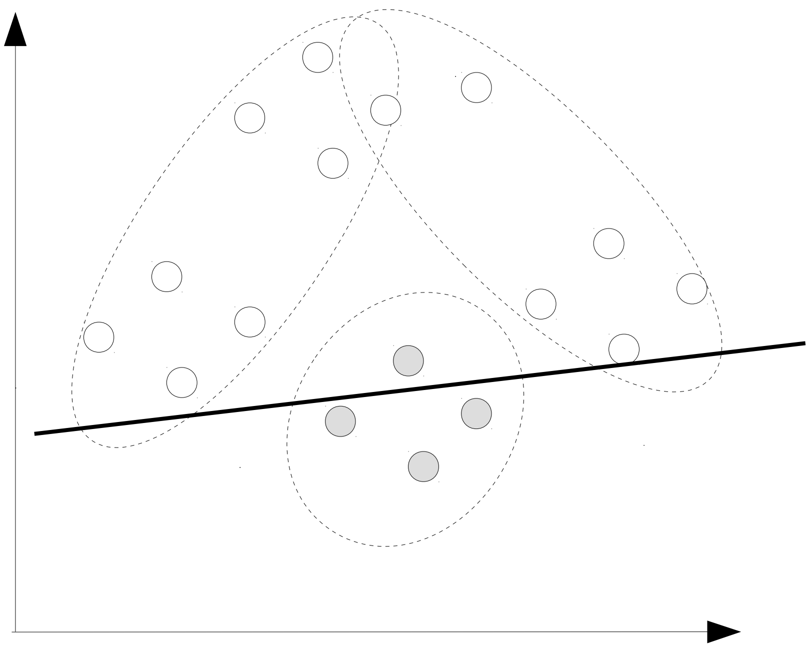
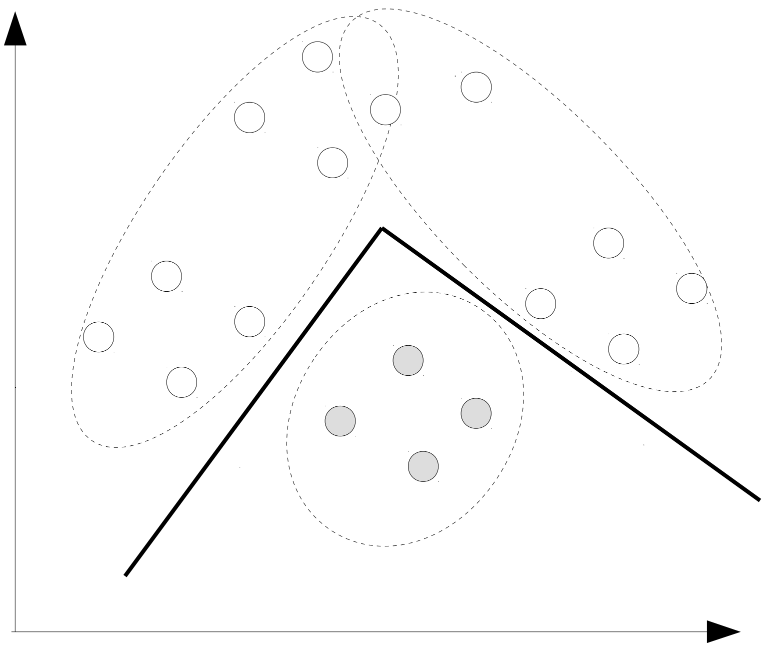
4.3　区分的線形識別関数とk-NN法

4.3.1　平面で区切れない場合



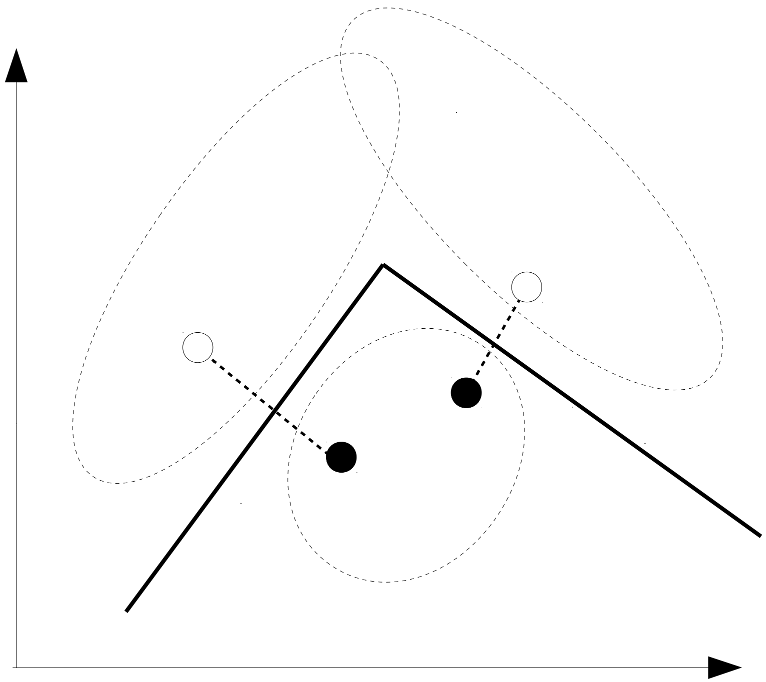
線形分離不可能なデータ



区分的線形識別関数を用いた場合

4.3.2 区分的線形識別関数の実現

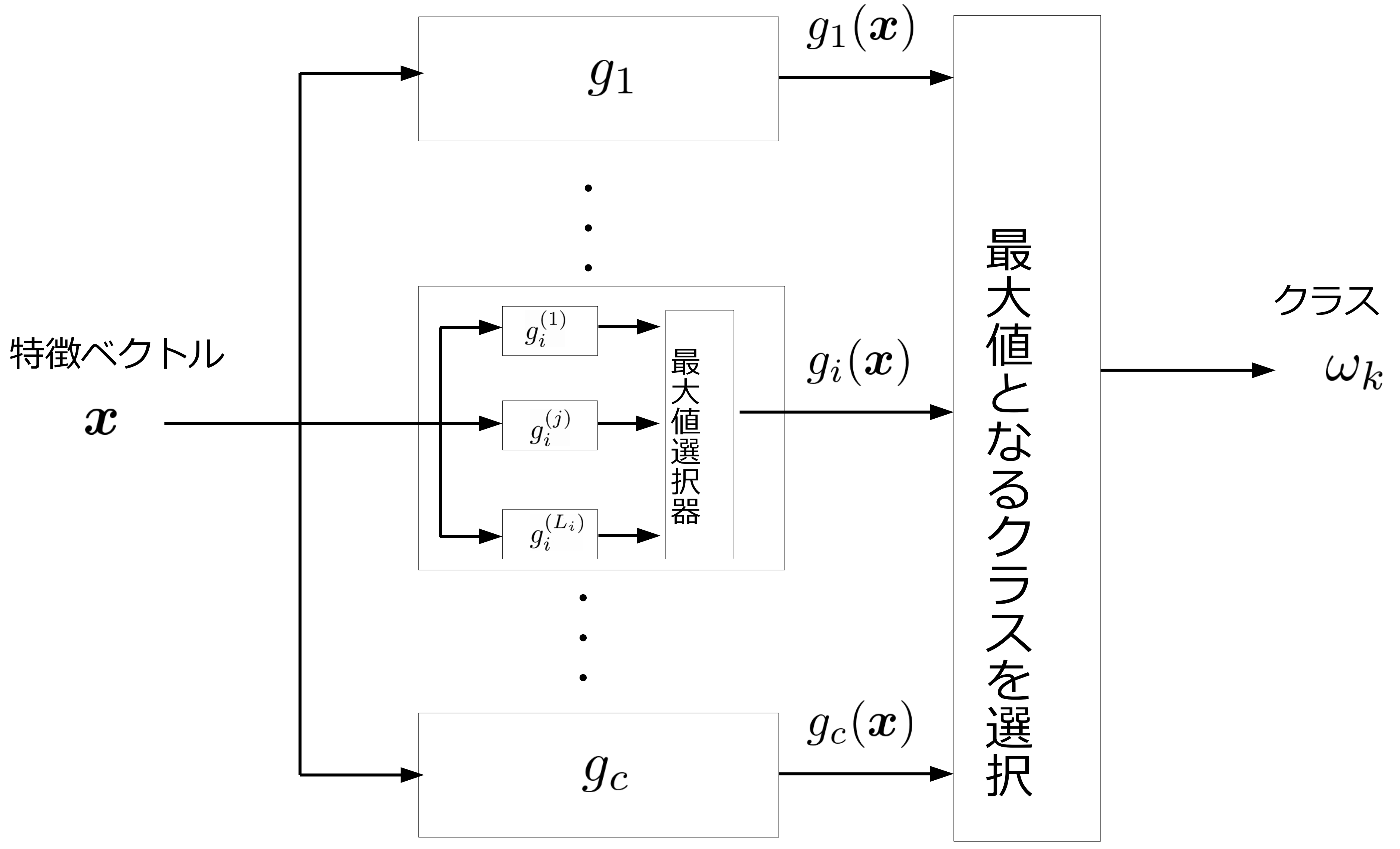
- 区分的線形識別関数の決め方
  - 各クラス複数のプロトタイプを設定



4.3.2 区分的線形識別関数の実現

- 区分的線形識別関数の定義
  - クラス ω_i の識別関数 $g_i(\boldsymbol{x})$を L_i 個の副次(線形)識別関数 $g_i^{(l)}(\boldsymbol{x})$ $(l = 1, \ldots, L_i)$ の最大値としてあらわす

4.3.2 区分的線形識別関数の実現

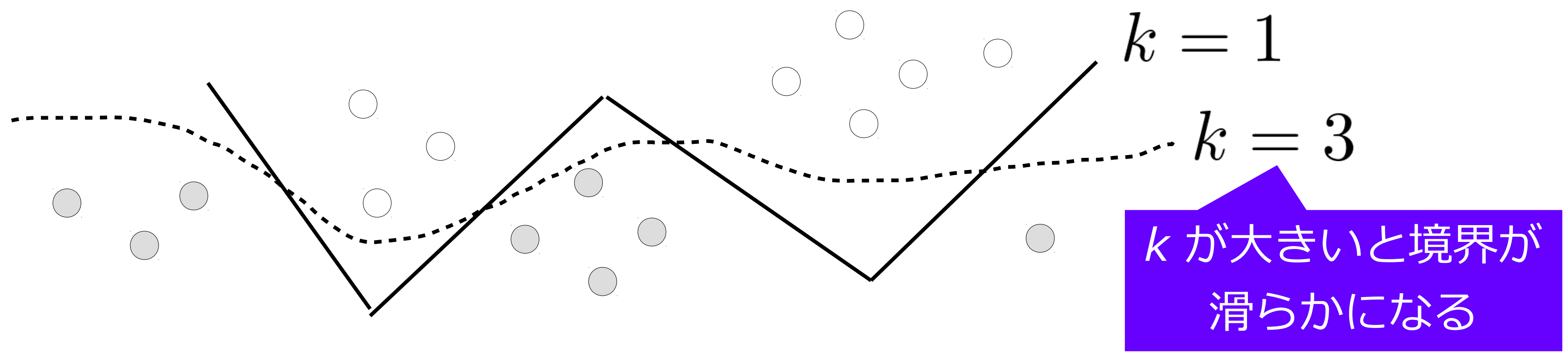


4.3.3 区分的線形識別関数の識別能力と学習

- 区分的線形識別関数の能力
  - どのような複雑な決定境界も任意の精度で近似可能
- 区分的線形識別関数の学習
  - 副次識別関数の個数 L_i と、各関数の重みの両方を学習しなければならない
- パーセプトロンの学習規則が適用できず、一般に学習は難しい

4.3.4 学習をあきらめるのも一手 - k-NN法

- k-ＮＮ法とは
  - 全ての学習データをプロトタイプとする
  - 入力に近い順からｋ個のプロトタイプのクラスを調べ、多数決を取る
  - 実験の際のベースラインとして用いられる



4.3.4 学習をあきらめるのも一手 - k-NN法

- k-NN法の特徴
  - 非線形性を示すデータにも対処できる可能性がある
- kが多いほど識別境界は滑らかになる
- k-NN法の問題点
  - 記憶容量
  - 計算時間

現在ではあまり問題にならない