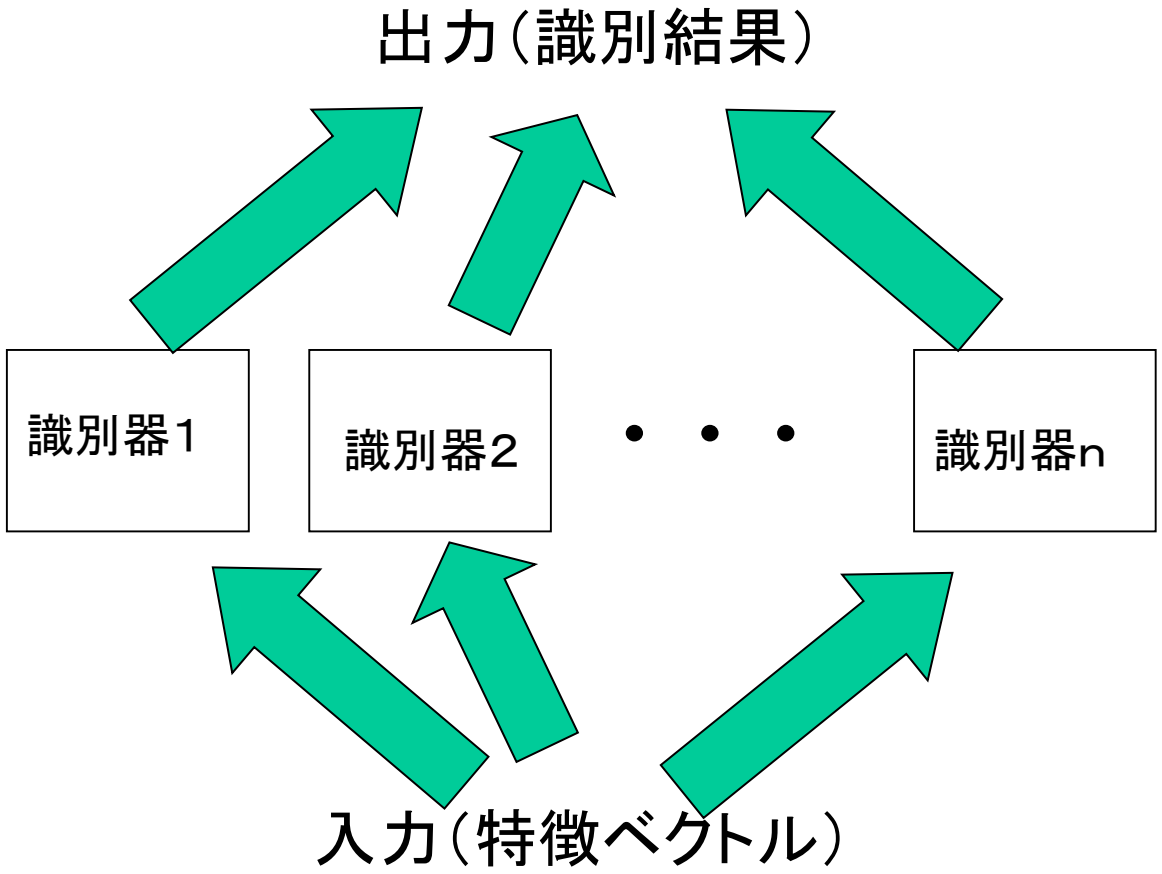
パターン認識特論

ADA Boosting

低精度の識別器を組み合わせて高精度の識別器を作る



Boostingとは同じタイプの識別器を異なる学習をさせて結合する

異なるタイプの識別器を学習をさせて結合する方法もある

ADA Boostingの利点

- 単純な計算の反復で安定な識別器を作ることができる.
- 最適化の特別なアルゴリズムを必要とせず，実装が容易.
- 人物顔検出などに用いることができ，特徴の「選択」も行える.

歴史的背景

- 1988年　Keans, Valiant：

  大量の学習データが与えられているとき、**ランダム選択よりも少しましな学習能力を有する識別器があれば**、それらを組み合わせることによって、任意の高精度識別器が構成できる
- 1989年 Schapire　最初の Boosting アルゴリズム
- 1990年 Freund　　改良型 Boosting
- 1995年 Freund and Schapire　ADA Boosting
- 1998年 Friedman et al Real AdaBoost

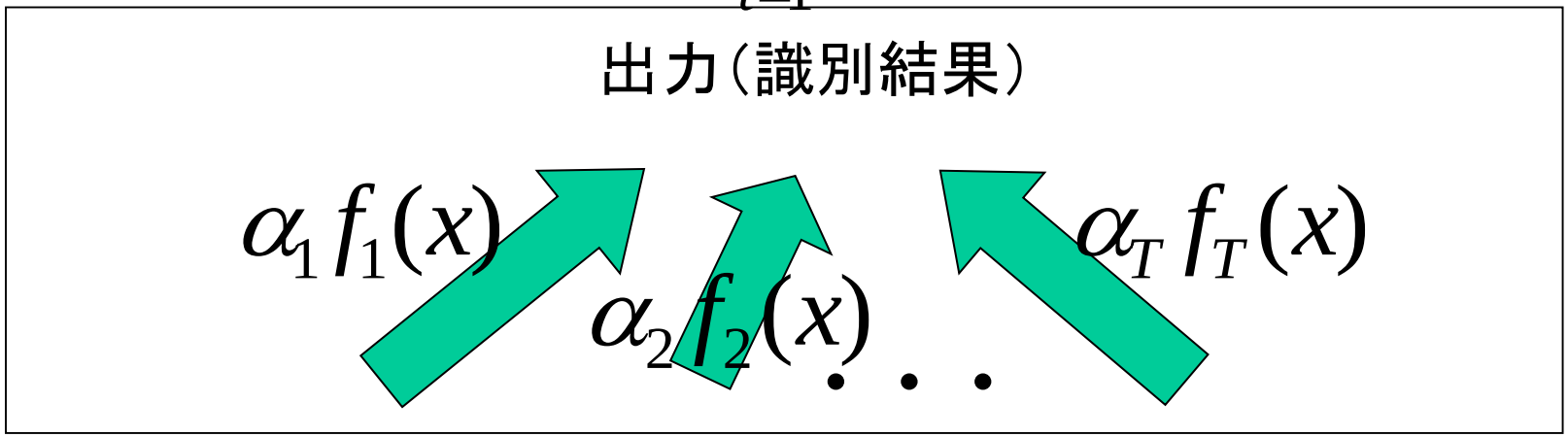
  Friedman et al LogitBoost

  Friedman et al Gentle ADABoost
- 2000年 Freund BrownBoosting

識別器出力の統合

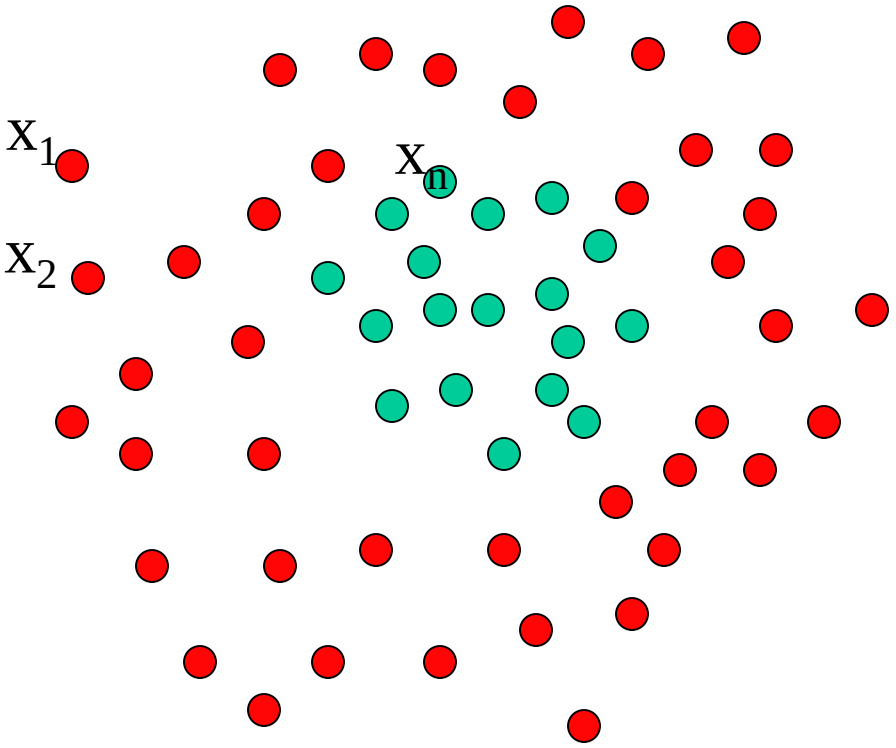
- 組み合わされた出力 $F(x)$ は、個々の識別器の出力 $f_t(x) \mapsto \{-1, +1\}$ の重み付き和として表される。

$$F(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$$



Step1:初期化

以下のデータ分布が与えられているとする.



各点は以下のクラスラベル:

$$y_n = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

と重みを持つ:

$$D_n = 1/N$$

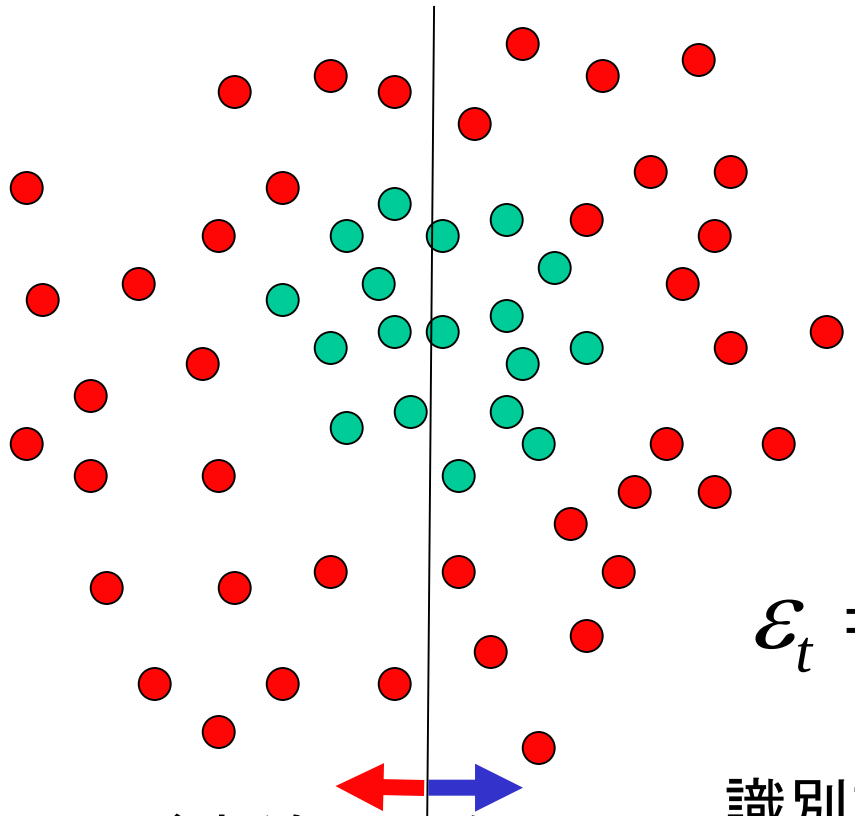
Step1:初期化

- $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$が与えられているとする。但し、$y_i \in \{-1, +1\}$ である。

- $D_i^1 = 1/N \ (i = 1 \cdots N)$とする。

  このとき、以下のStep2と3を、t=1～Tまで繰り返す。

Step2：識別器tの学習



弱識別器が直線の場合

$$I(E)=\begin{cases}0 & E=false\\ 1 & E=true\end{cases}$$

$$\varepsilon_t=\sum_{n=1}^{N}D_n^t I(y_n\neq f_t(\boldsymbol{x}_n))$$

識別誤差を最小化するように学習

Step3：重みの更新

重み付きエラー

$$I(E)=\begin{cases}0 & E=false\\ 1 & E=true\end{cases}$$

$$\varepsilon_t=\sum_{n=1}^{N} D_n^t I(y_n \neq f_t(\boldsymbol{x}_n))$$

識別器tの重み

$$\alpha_t=\frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

間違わないほど大きな値になる

データの重み

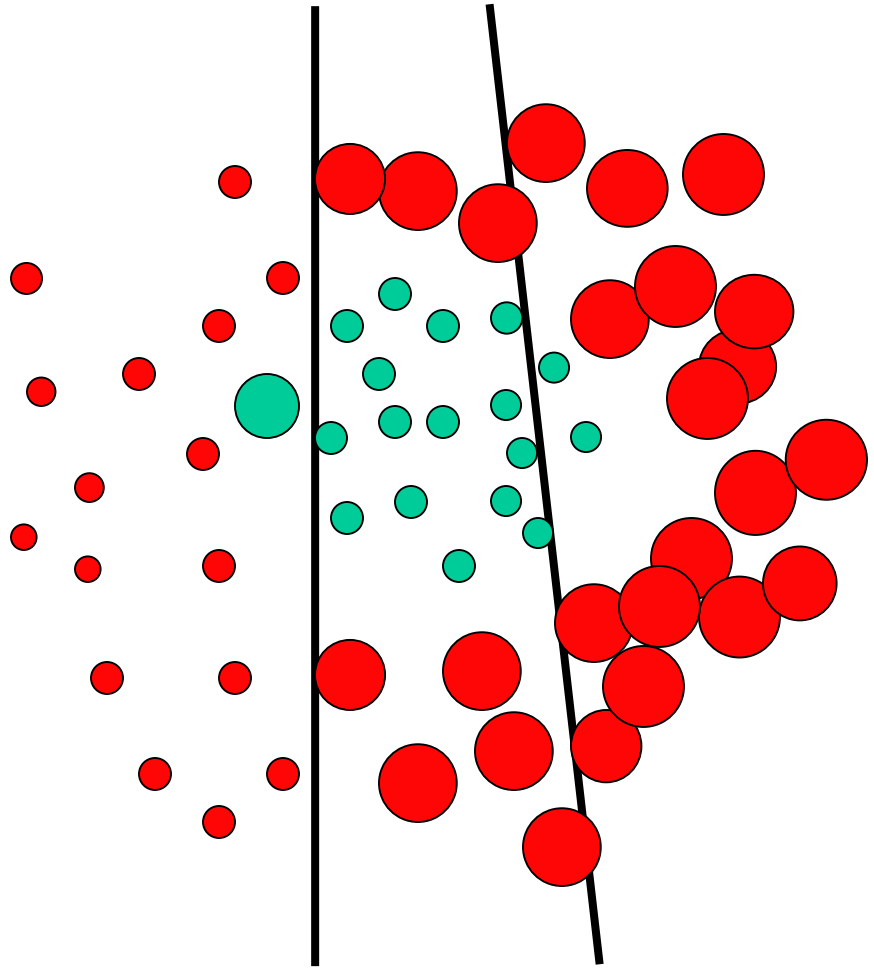
$$\sum_{n=1}^{N} D_n^t=1$$

$$D_n^{t+1}=D_n^t\frac{\exp\{-\alpha_t y_n f_t(\boldsymbol{x}_n)\}}{Z_t}$$

正規化項

間違うほど大きな値になる

Step3：重みの更新（イメージ図）

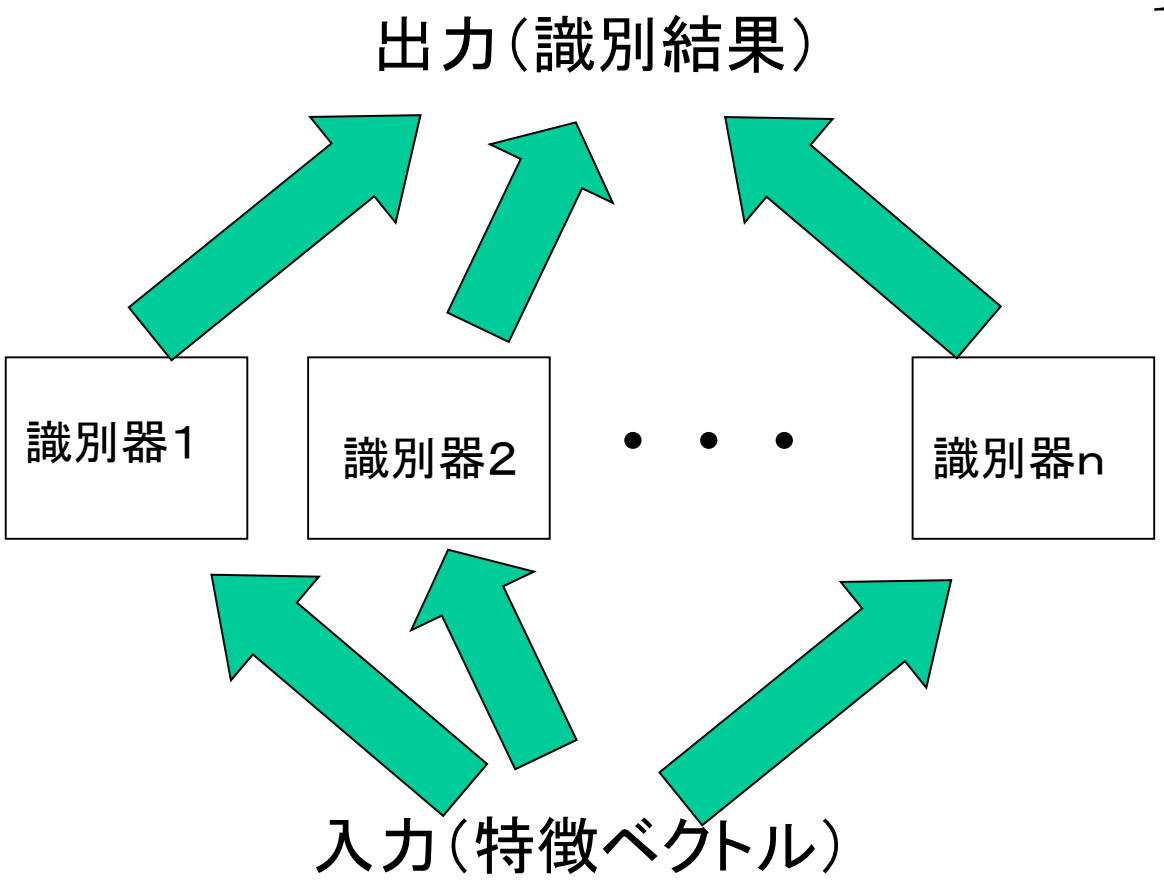


誤識別を起こしたデータの重みを増す

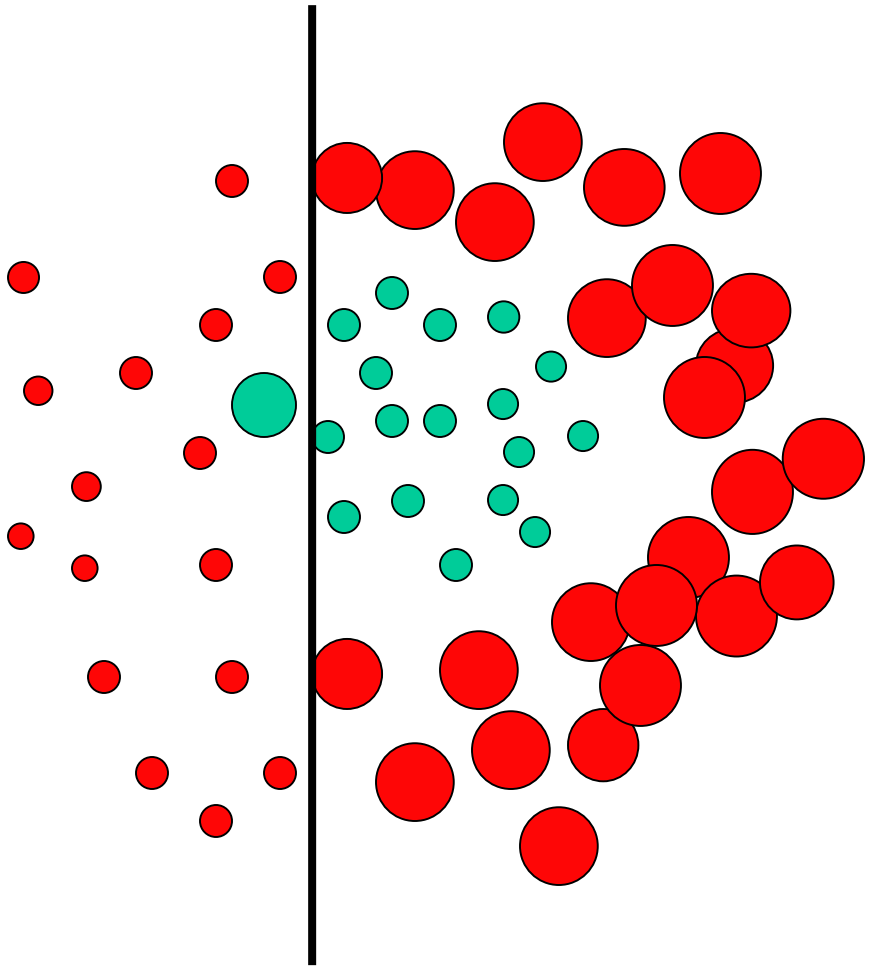
学習終了後に得られる識別器

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

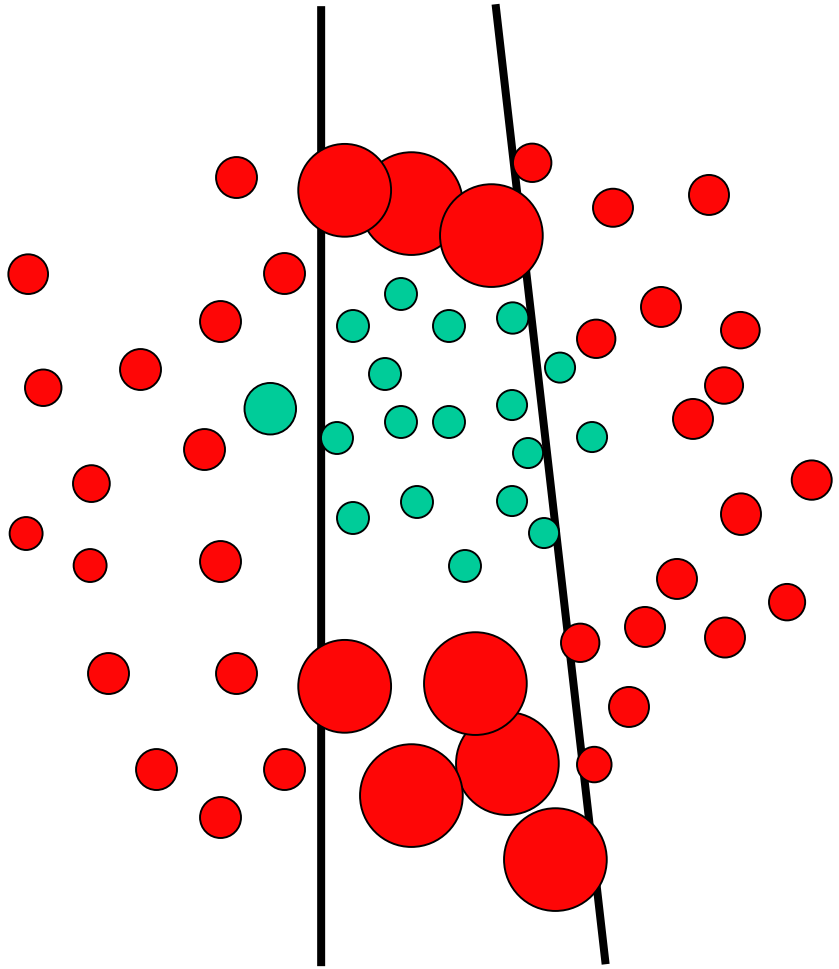
$f(x)$



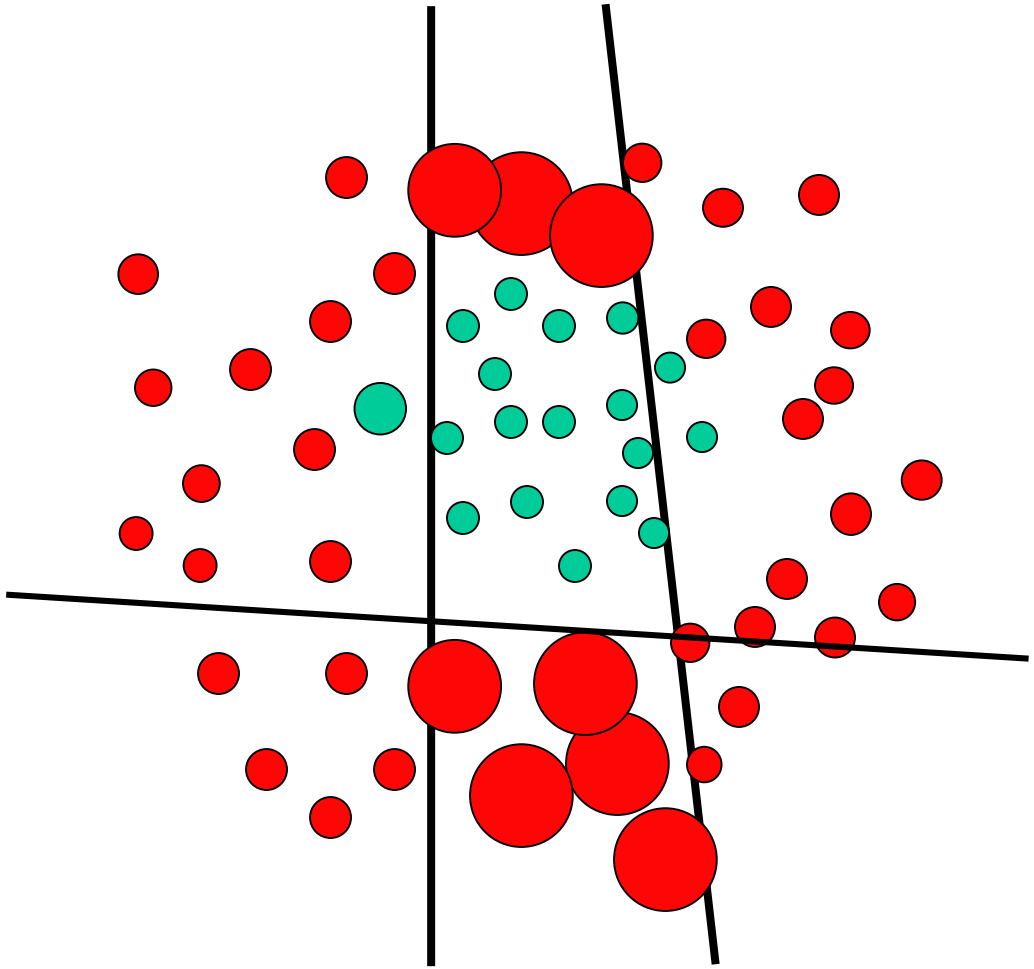
続き



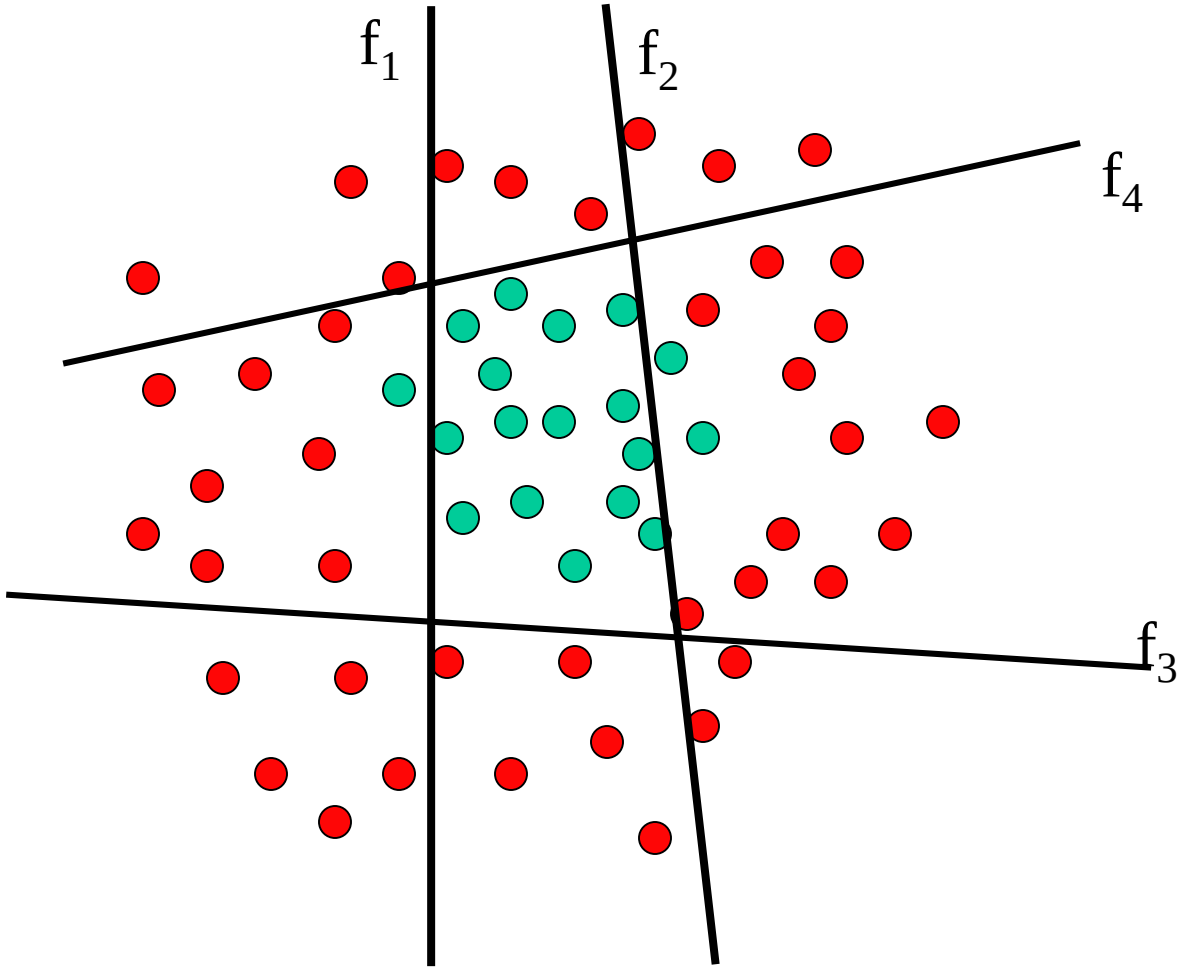
続き



続き

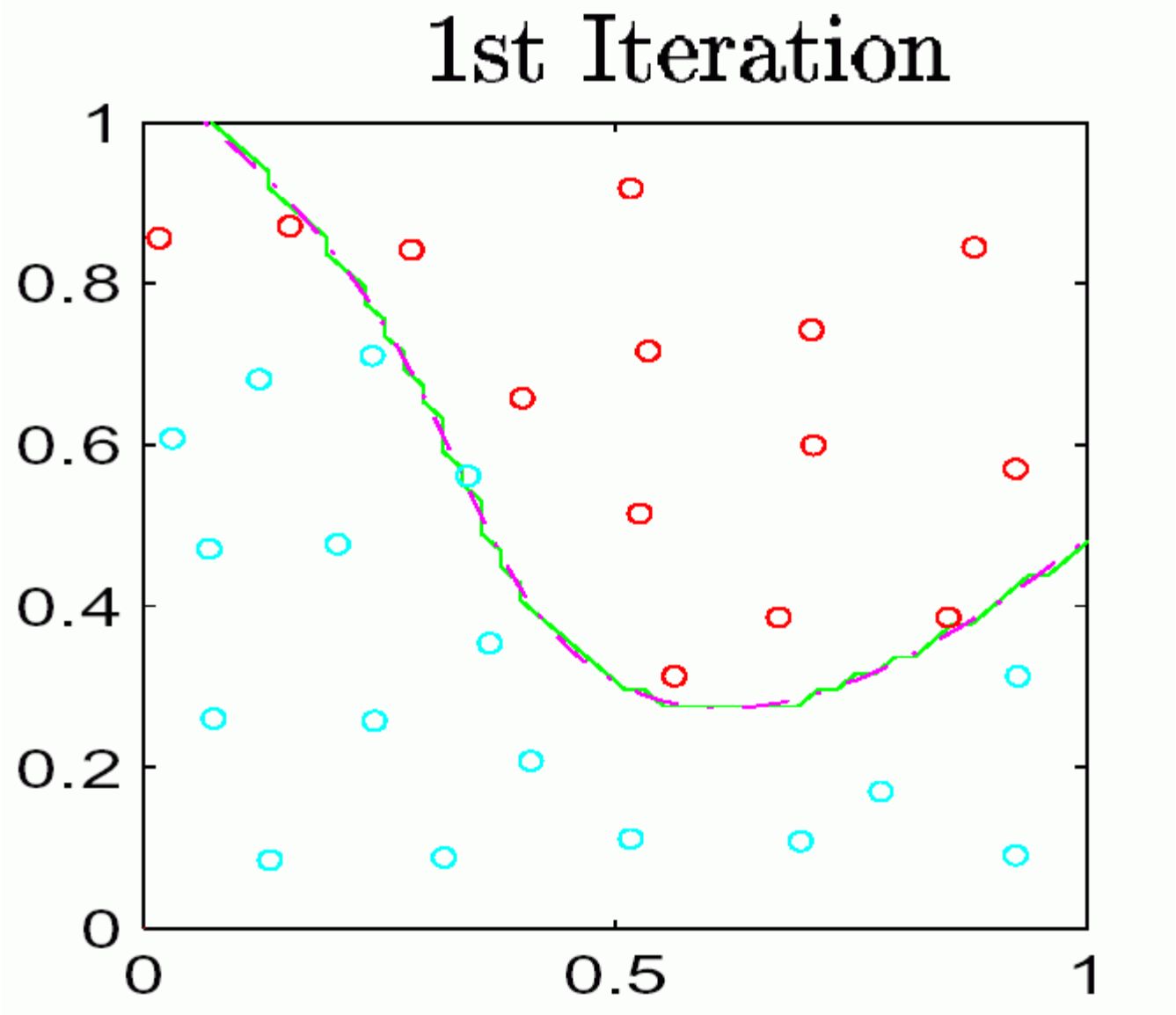


最終結果

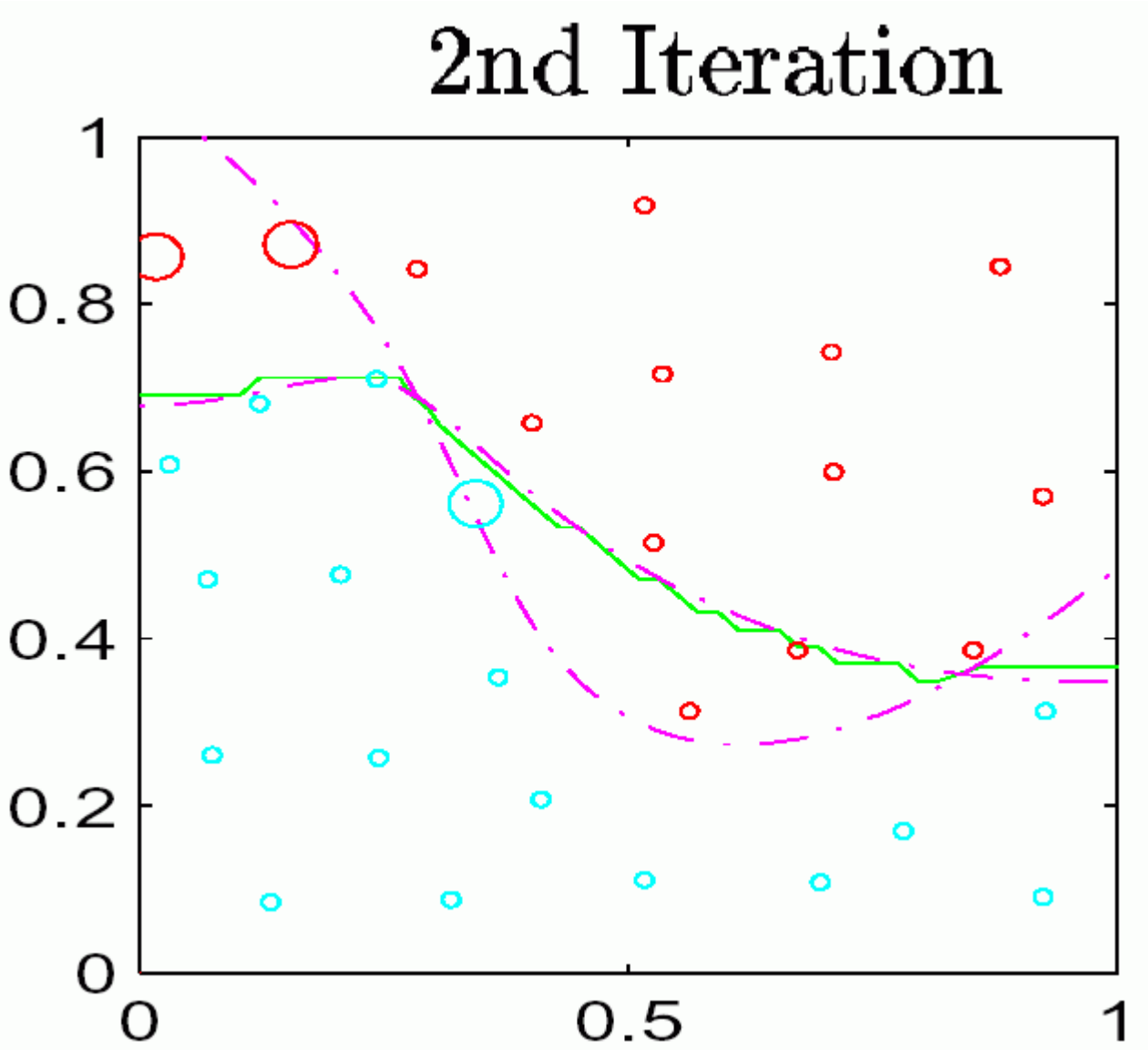


線形識別面4枚から非線形識別面を作ることができる

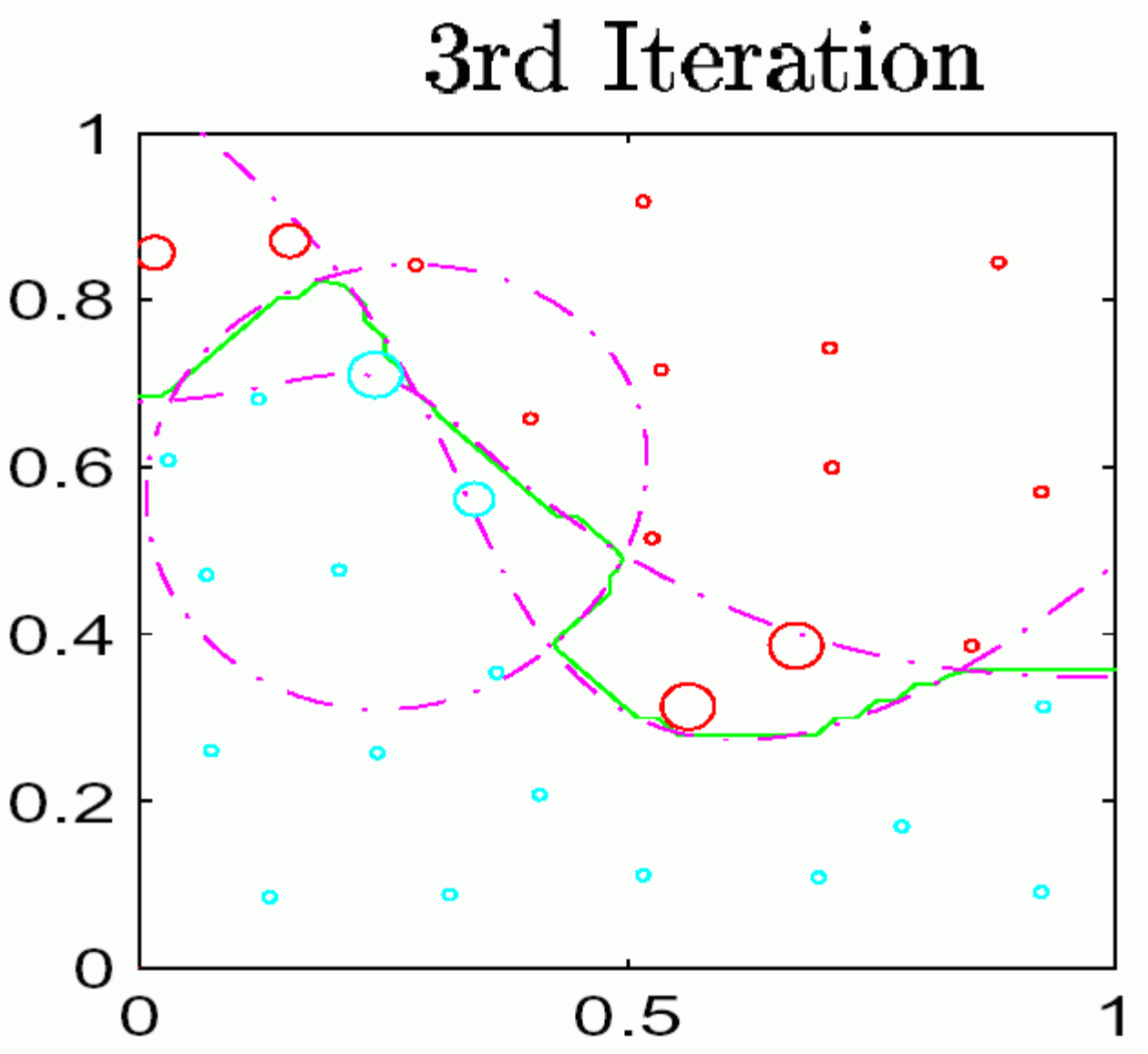
例



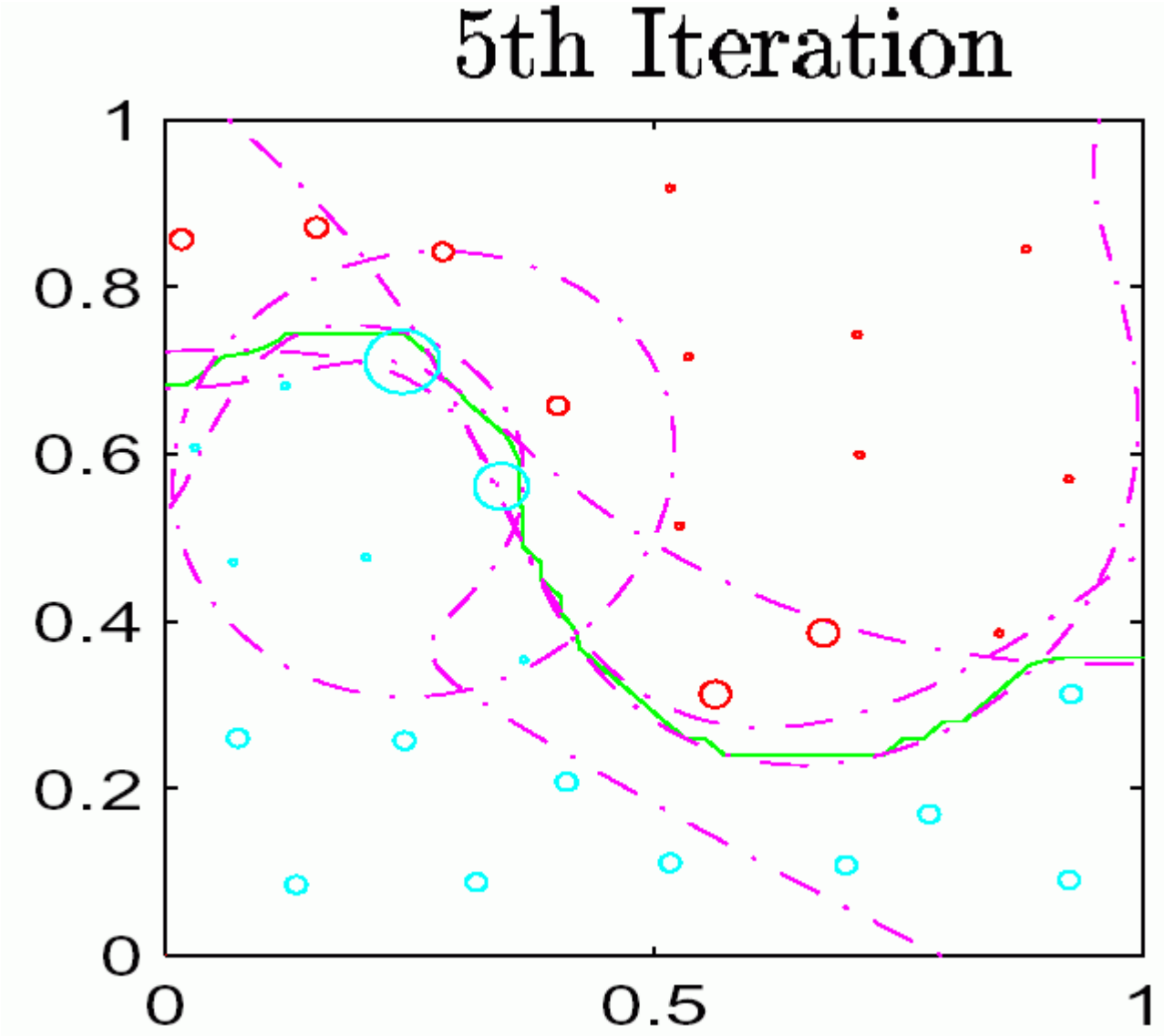
例



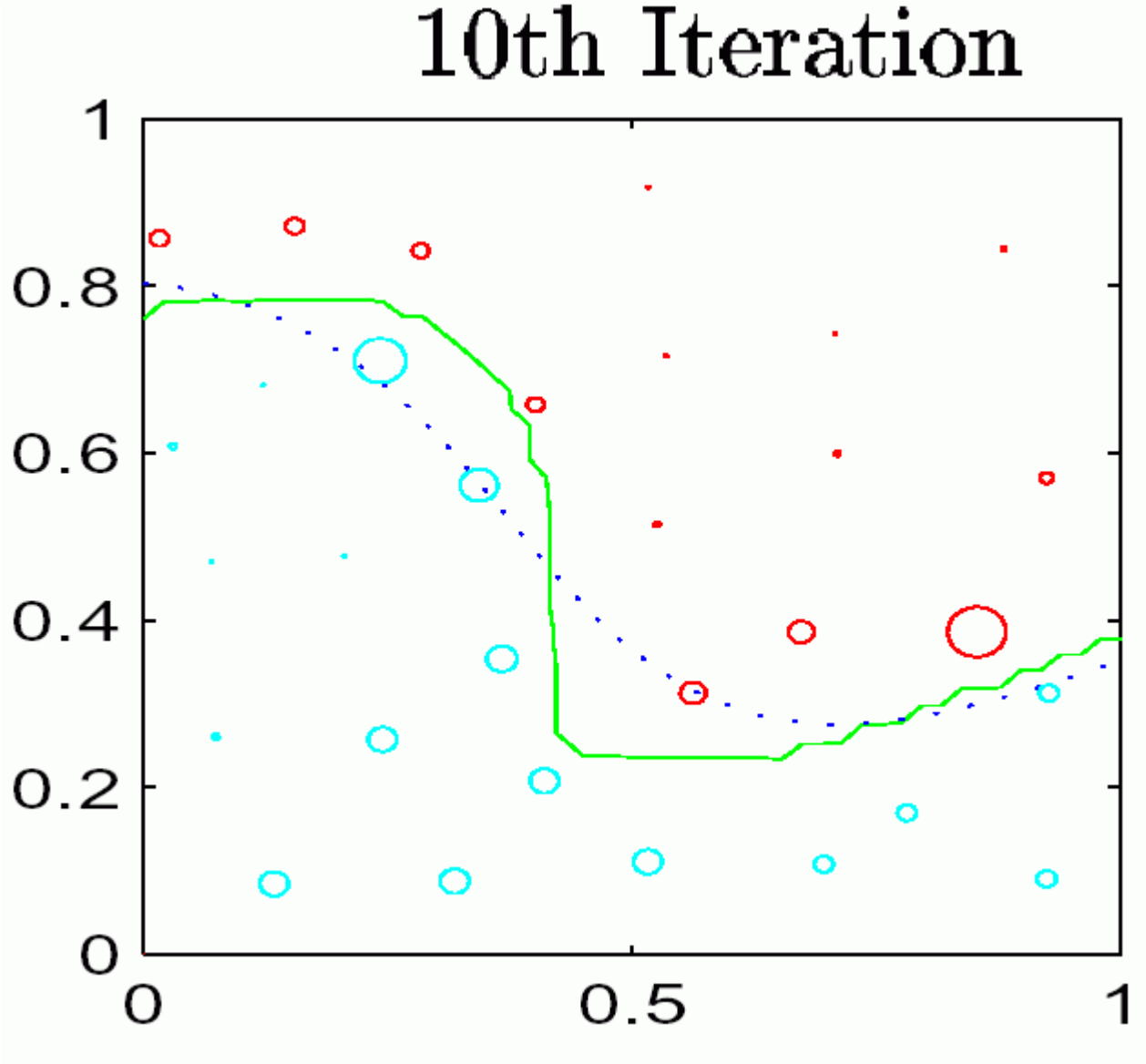
例



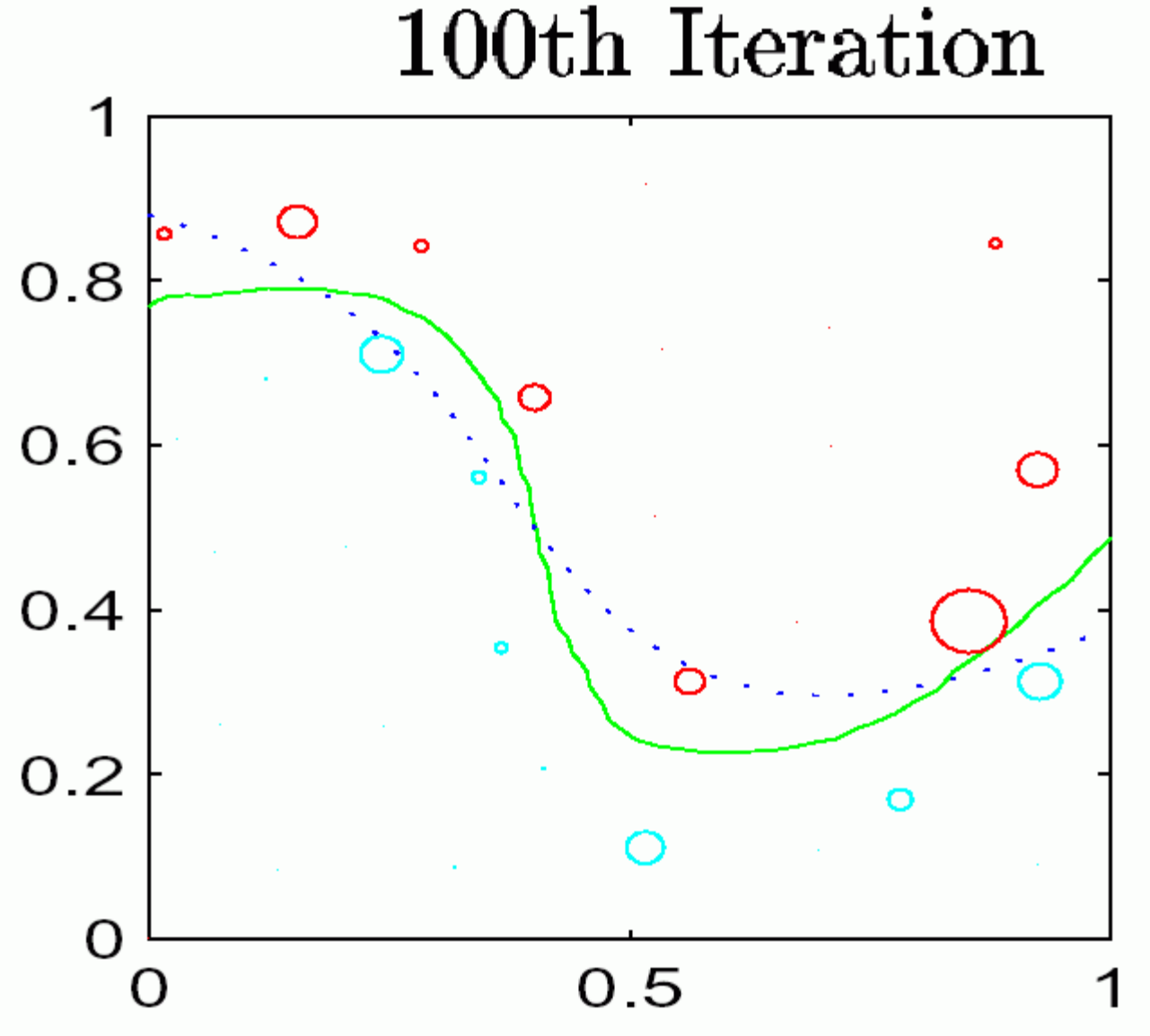
例



例



例



ADA Boosting の特性

- 誤識別確率 $\leq \frac{1}{N}\sum_{i=1}^{N}\exp\left(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)\right)=\prod_{t=1}^{T}Z_t$

$$Z_t=\sum_{i=1}^{N}D_i^t\exp\left(-\alpha_t y_i h_t(x_i)\right)$$

学習すればするほど誤りは少なくなる。

$$\prod_{t=1}^{T}Z_t\leq\exp\left(-2\sum_{t=1}^{T}\gamma_t^2\right)\qquad \gamma_t=1/2-\varepsilon_t$$

しかも、誤識別が減るのが速い。

未知入力に対するリスク

トレーニングサンプルに対する誤りの確率

$$\Pr[H(x) \neq y]$$

未知入力に対する誤りのオーダ $O\left(\sqrt{\frac{Td}{N}}\right)$

d: VC dimension

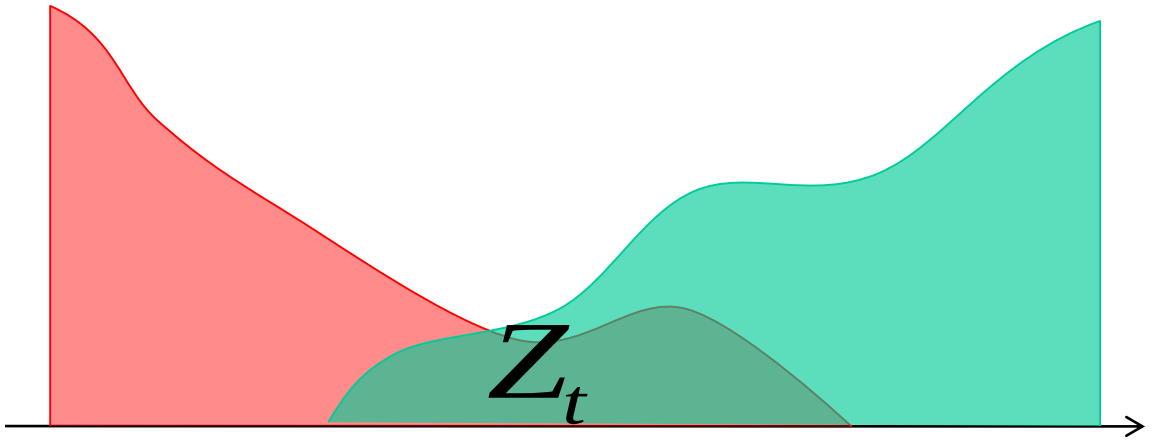
$$\Pr[H(x) \neq y] + O\left(\sqrt{\frac{Td}{N}}\right)$$

デモ

- http://www.cs.technion.ac.il/~rani/LocBoost/

Real ADABoost

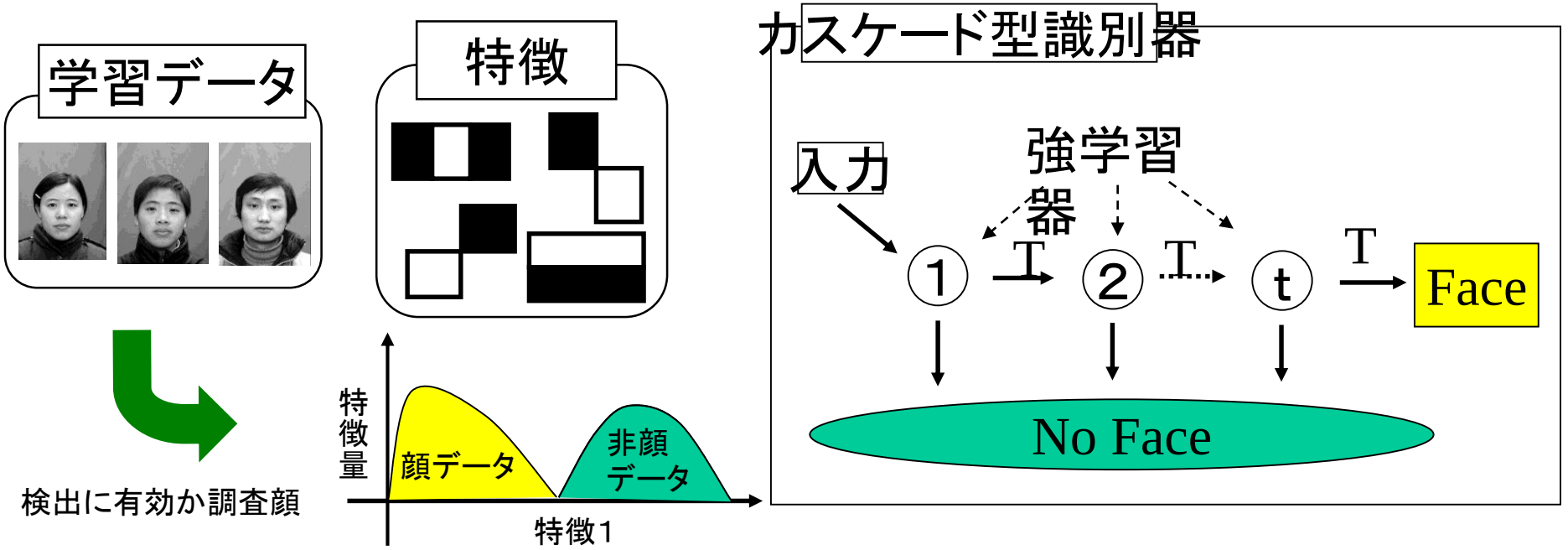
- Step3の重み更新式が違う



$$\alpha_t = \frac{1}{Z_t}$$

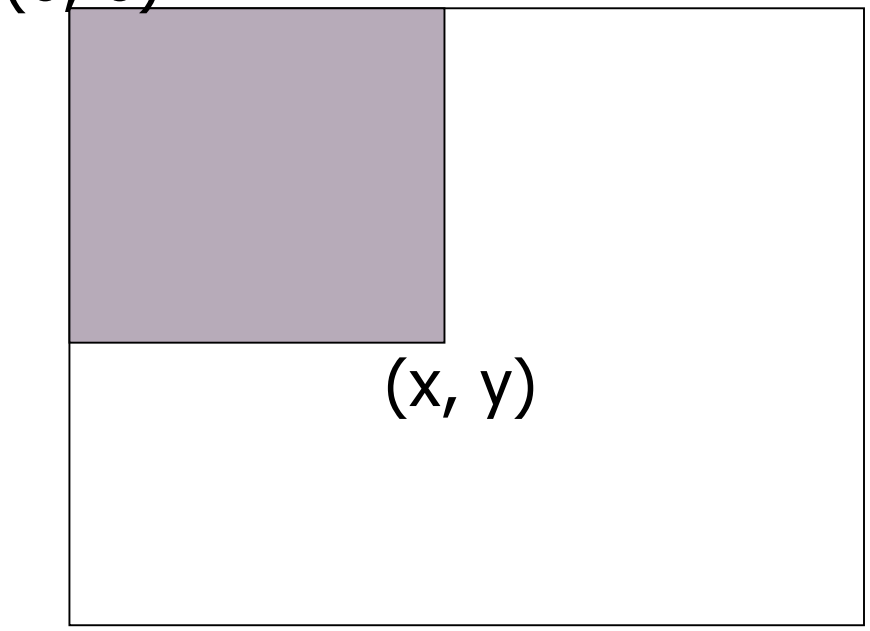
識別による検出
（クラスモデルとのマッチング）
Cascaded ADABoosting

- ADABoosting で作られた強識別器を多段に接続する（速度と精度を両立させる工夫）



内積計算の高速化
Integral Image（Viola&Jones）

- Integral Imageとは矩形領域内の画素値の総和を画素値とする画像



$$ii(x,y)=\sum_{x'\le x,y'\le y} i(x',y')$$

$ii(x,y)$：integral image

$i(x,y)$：画素

$s(x,y)$：縦の画素の総和

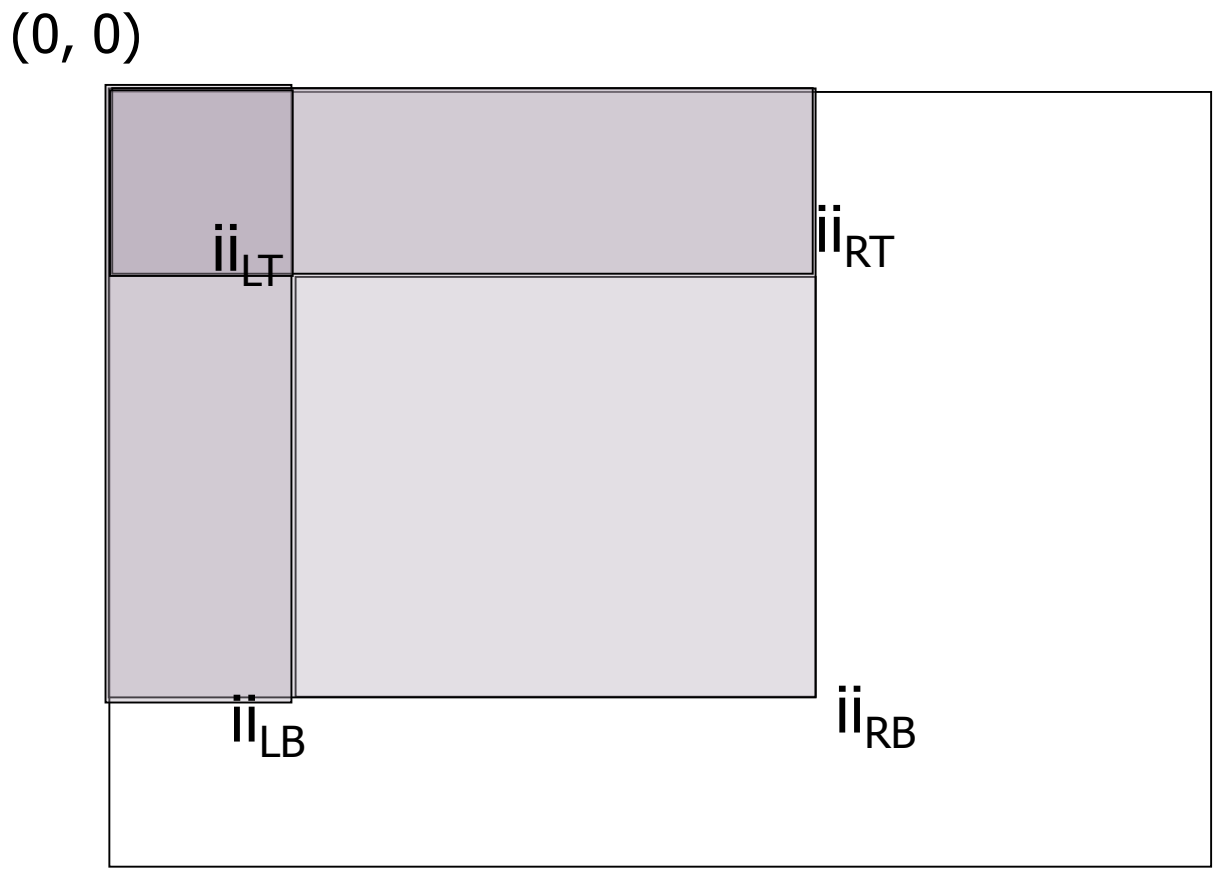
$$s(x,y)=s(x,y-1)+i(x,y)$$

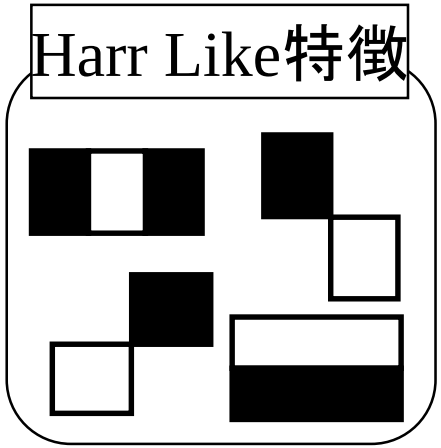
$$ii(x,y)=ii(x-1,y)+s(x,y)$$

$$s(x,-1)=0, ii(-1,y)=0$$

Integral Imageを用いた矩形内積分

$$ii_{RB} - ii_{RT} - ii_{LB} + ii_{LT}$$





わずか3回の加減算で，任意の矩形領域の積分が行え，顔検出で用いられるHarr Like特徴も高速に計算可能

検出時の位置・大きさの変化に対する対処法

- イメージピラミッドの走査