パターン認識特論
～研究者から見たBoosting～

2011.10.04

牧田孝嗣@産業技術総合研究所

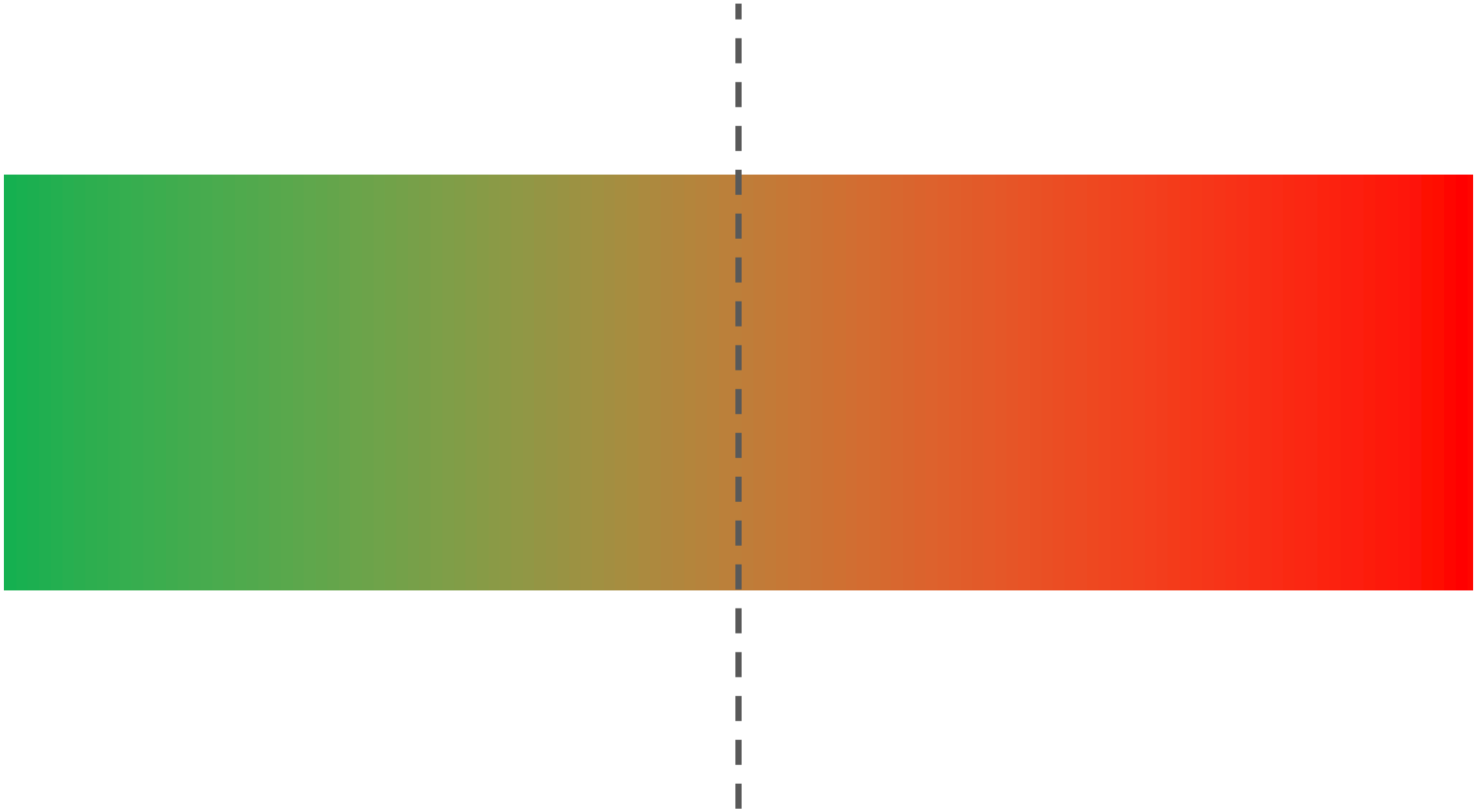
自己紹介

- 2009年3月：
  奈良先端科学技術大学院大学
  博士後期課程（工学）修了
- 2009年4月～2010年7月：
  奈良先端科学技術大学院大学
  博士研究員
- 2010年8月～現在：
  産業技術総合研究所
  特別研究員

パターン認識

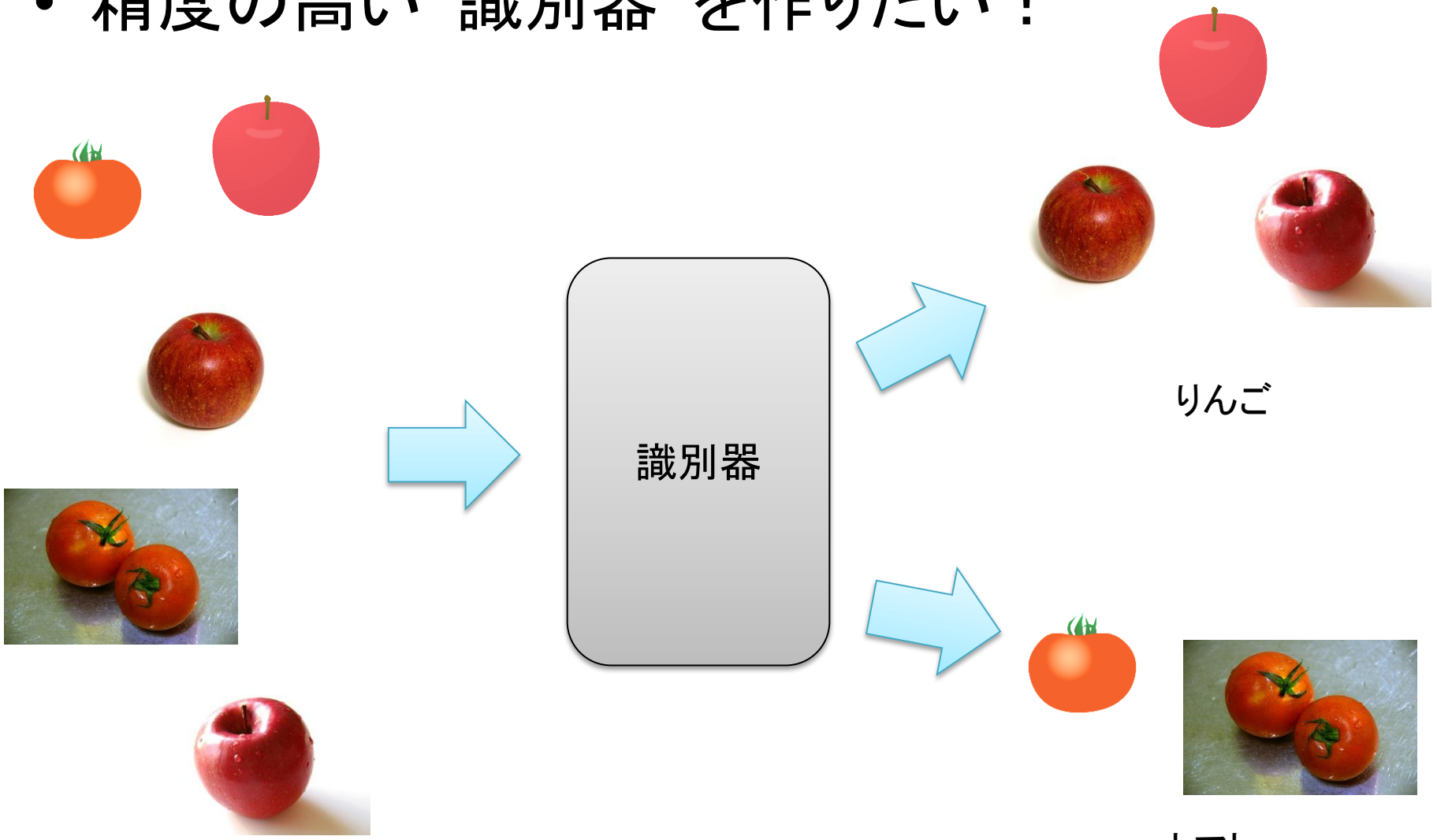
パターン認識とは？

パターン認識と境界



パターン認識でやりたいこと

- 精度の高い“識別器”を作りたい！

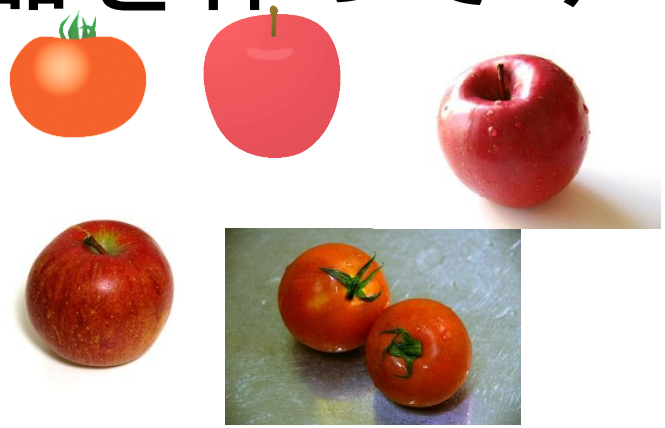


もし皆さんが識別器を作るとしたら・・・

- 案1：

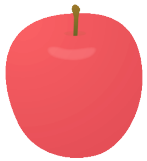
- 案2：

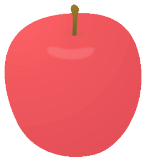
識別器を作ってみても・・・



Aさんの識別器

Bさんの識別器



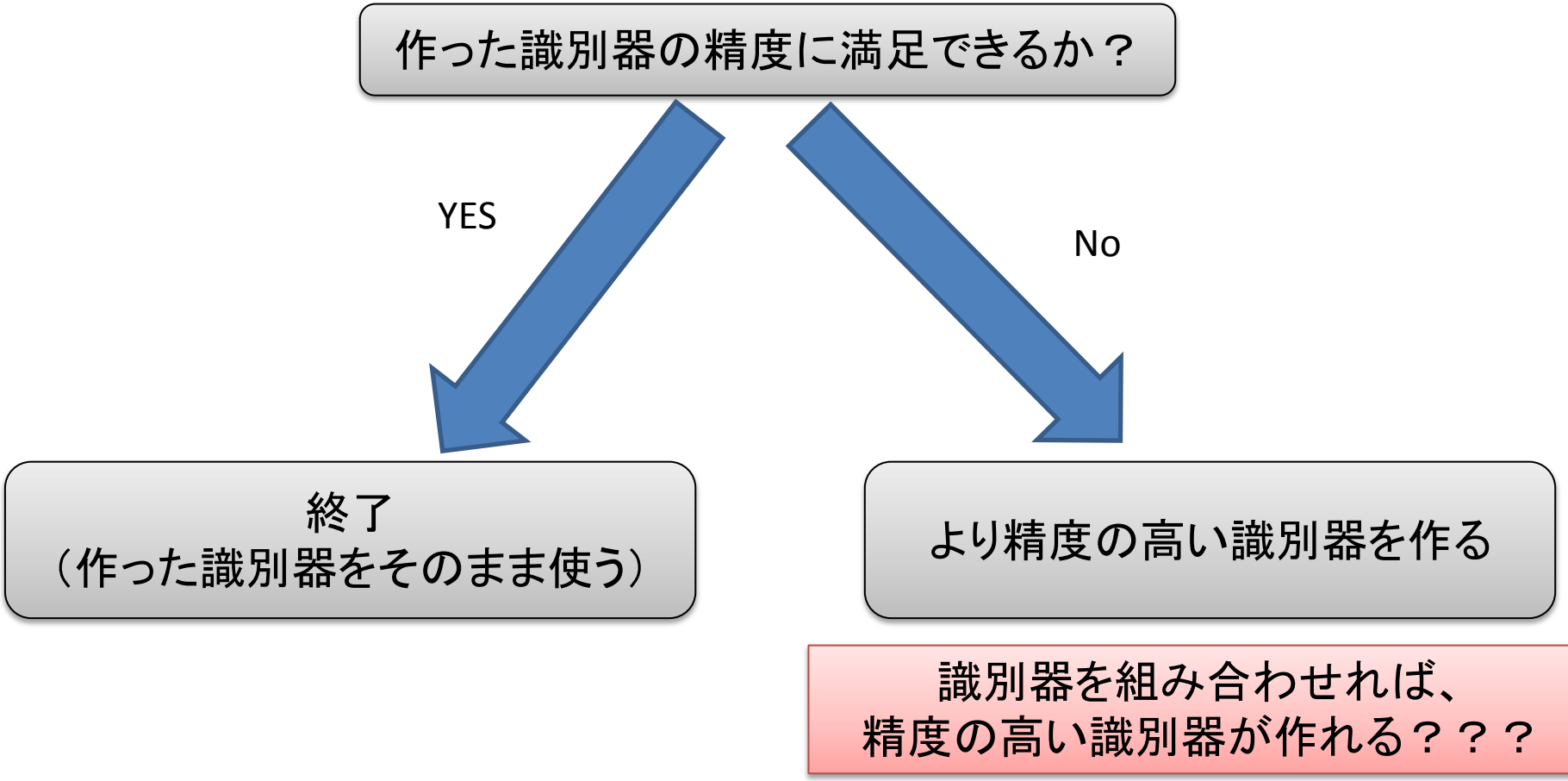


正解率100%は難しい・・・場合が多い

正解、不正解の4パターン

真値 / 推定結果	りんご	トマト
りんご		
トマト		

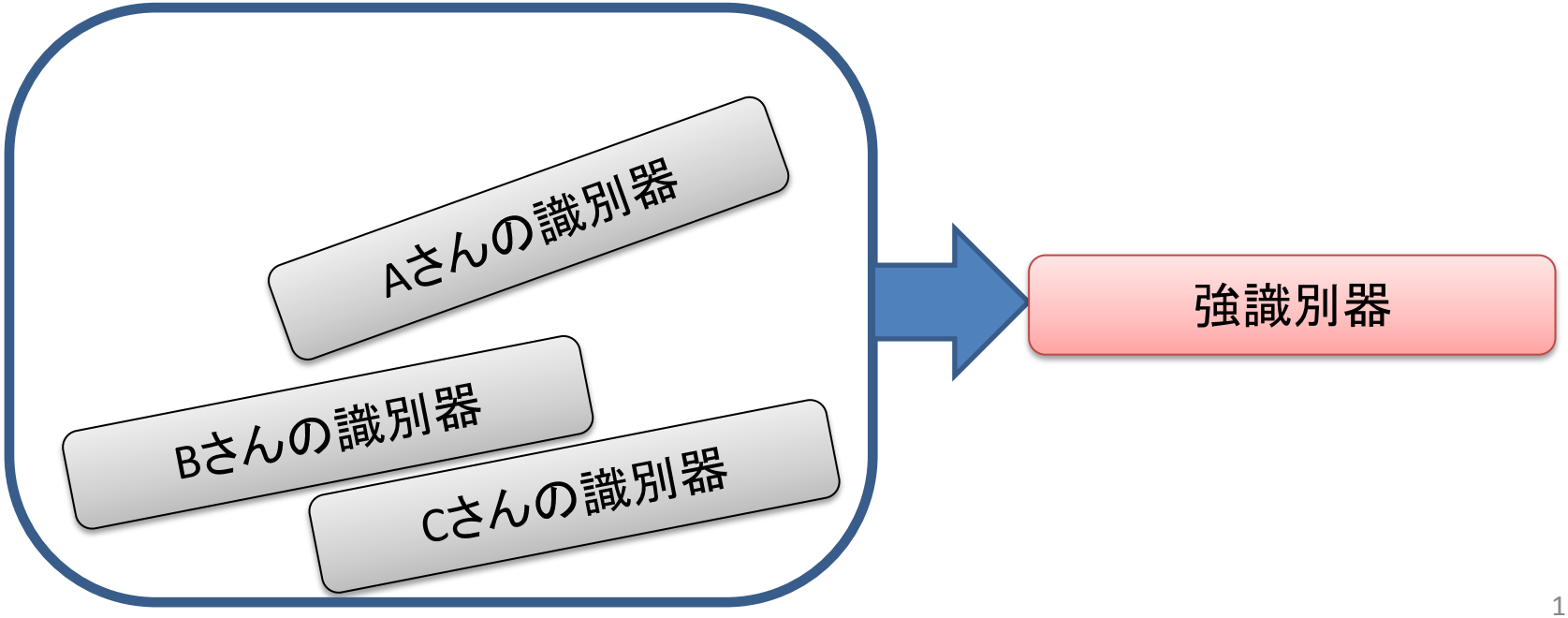
識別器の満足度


作った識別器の精度に満足できるか？
YES
No
終了
（作った識別器をそのまま使う）
より精度の高い識別器を作る
識別器を組み合わせれば、
精度の高い識別器が作れる？？？

Boosting

Boostingとは？

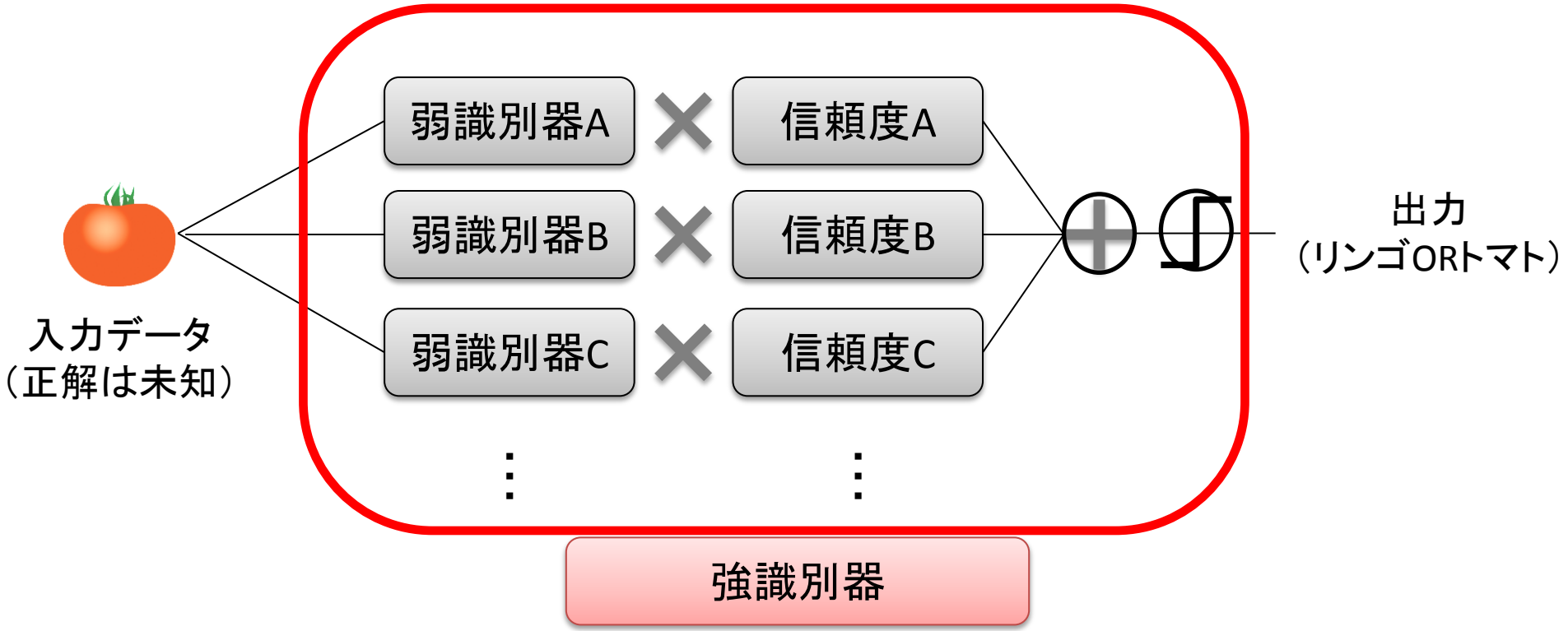
- 精度の低い識別器の集合から、精度の高い識別器を作る手法、のこと
- 精度の低い識別器のことを弱識別器、精度の高い識別器のことを強識別器、と呼ぶ



AdaBoost

AdaBoostとは？

- Boostingによって強識別器を作成する手法の1つ
- 弱識別器の出力値と信頼度の積の和を計算し、その符号によって入力データを判別



AdaBoostのイメージ

- 多数決、です
- ただし、“信頼度”の概念あり

AdaBoostによる強識別器作成アルゴリズム

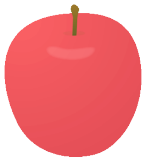
- １：事前準備
  １．１：学習サンプルの準備
  １．２：学習サンプルに“重み”を設定
  １．３：弱識別器を準備

- １：事前準備
  - １．１：学習サンプルの準備
  - １．２：学習サンプルに“重み”を設定
  - １．３：弱識別器を準備

各サンプルに、ラベルを付与
（りんご：＋１、トマト：－１）





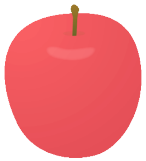


ラベル	＋１	－１	－１	＋１	＋１

- １：事前準備
  - １．１：学習サンプルの準備
  - １．２：学習サンプルに“重み”を設定
  - １．３：弱識別器を準備
    - 重みを設定
      - 最初は、全て同じ大きさの重みとする
      - 重みの合計は、常に1





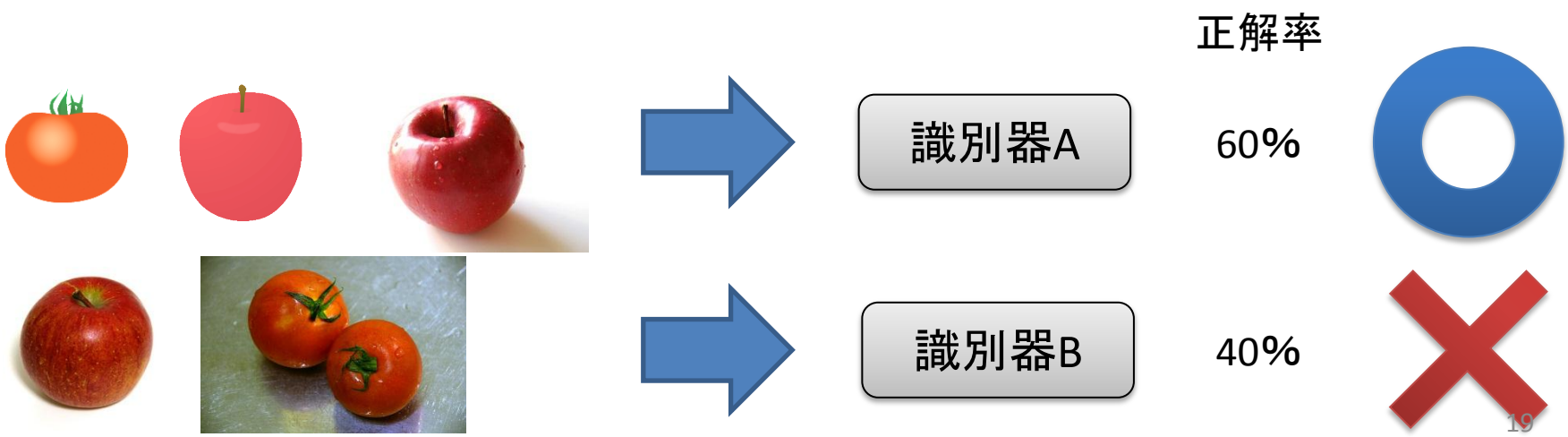


ラベル	＋1	－1	－1	＋1	＋1
重み	0.2	0.2	0.2	0.2	0.2

- １：事前準備
  - １．１：学習サンプルの準備
  - １．２：学習サンプルに“重み”を設定
  - １．３：弱識別器を準備

弱識別器のルール

1. 入力データに対して、+1、もしくは-1を出力する2値関数である
2. 弱識別器の正解率は、サンプルの重みを利用して計算する
3. 学習サンプルに対して、正解率が50%以上である



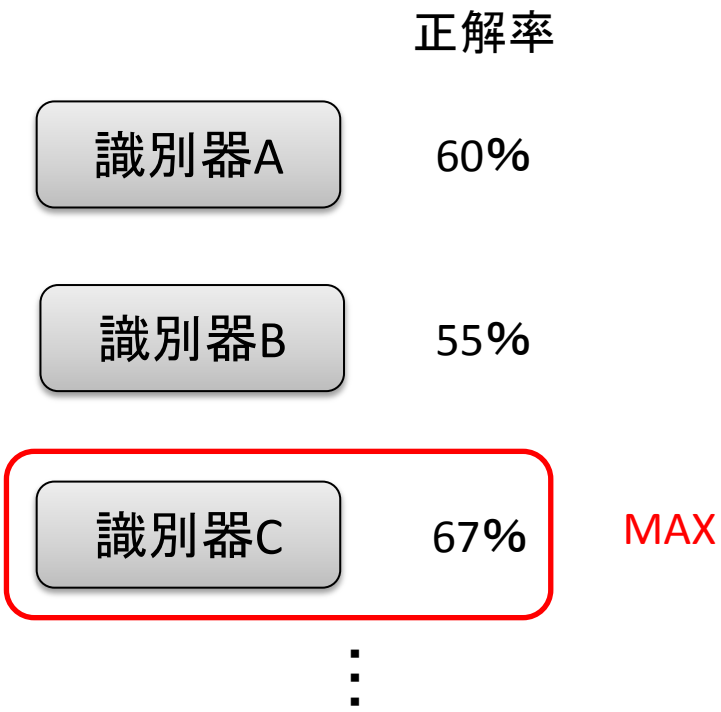
正解率が50%未満の場合・・・

AdaBoostによる強識別器作成アルゴリズム

- ２：弱識別器群の作成
  - ２．１：弱識別器の選択
  - ２．２：弱識別器の信頼度計算
  - ２．３：サンプルの重みの更新

- ２：弱識別器群の作成
  - ２．１：弱識別器の選択
  - ２．２：弱識別器の信頼度計算
  - ２．３：サンプルの重みの更新

各弱識別器の正解率を計算し、最も正解率が高い弱識別器を選択



- ２：弱識別器群の作成
  - ２．１：弱識別器の選択
  - ２．２：弱識別器の信頼度計算
  - ２．３：サンプルの重みの更新

正解率

識別器A　60%

識別器B　55%

識別器C　67%

⋮

信頼度　α
エラー率　e
とすると

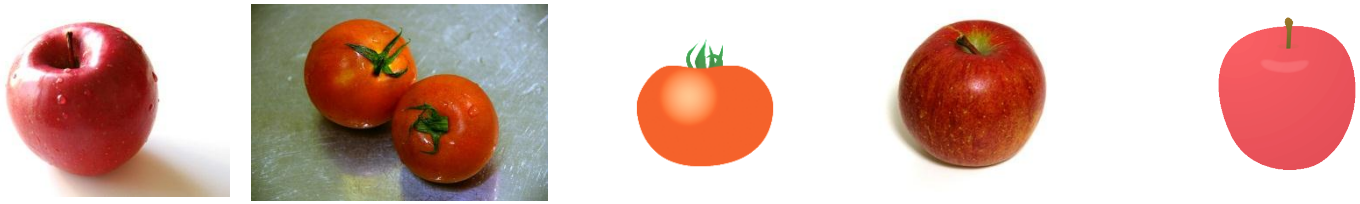
$$\alpha = \frac{1}{2}\ln\left(\frac{1-e}{e}\right)$$

$$\alpha = \frac{1}{2}\ln\left(\frac{1-0.33}{0.33}\right)$$

- ２：弱識別器群の作成
  - ２．１：弱識別器の選択
  - ２．２：弱識別器の信頼度計算
  - ２．３：サンプルの重みの更新

選択した弱識別器で学習サンプルのラベルを推定

- 推定が正解⇒重みを小さく
- 推定が不正解⇒重みを大きく



ラベル	＋1	－1	－1	＋1	＋1
重み（更新前）	0.2	0.2	0.2	0.2	0.2
推定結果	＋1	－1	＋1	＋1	－1
重み（更新後）	小さく	小さく	大きく	小さく	大きく

次の弱識別器を作成する場合、２．１に戻る

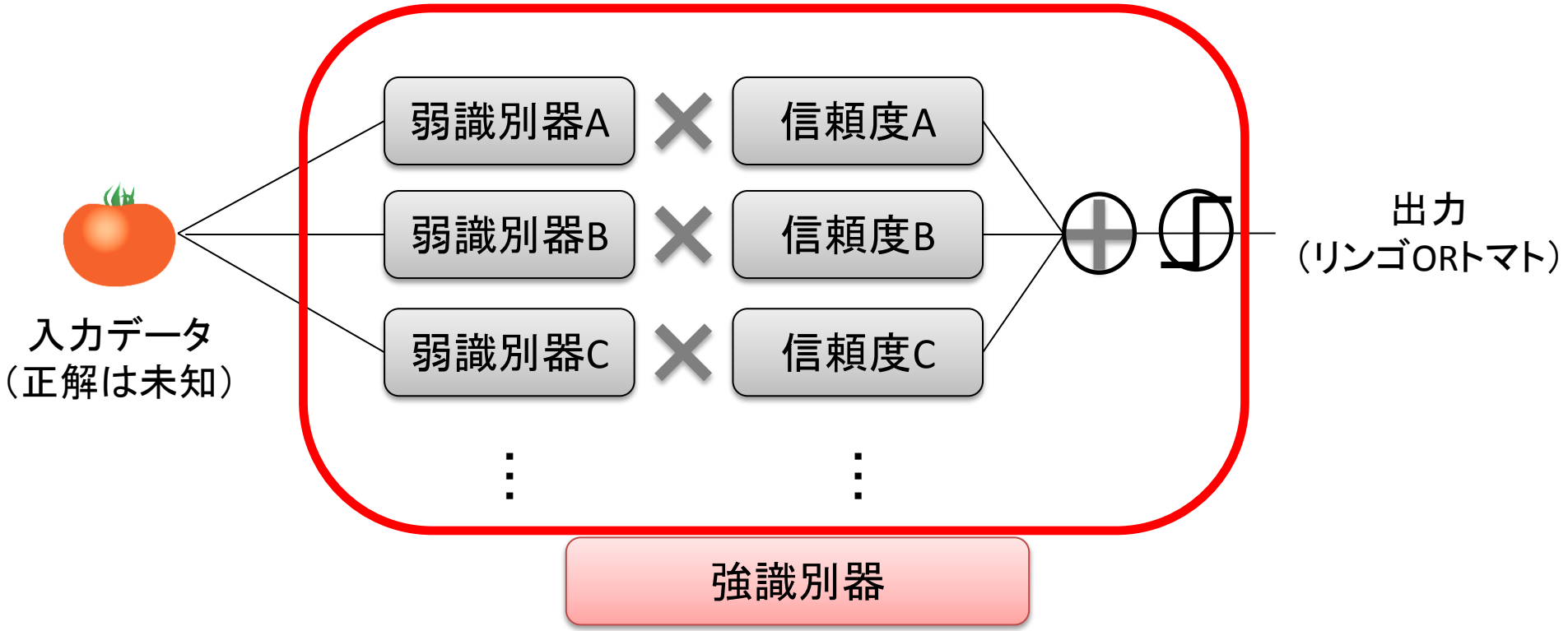
AdaBoostによる強識別器作成アルゴリズム

- ３：強識別器の作成
  ３．１：弱識別器の選択の打ち切り

- ３：強識別器の作成

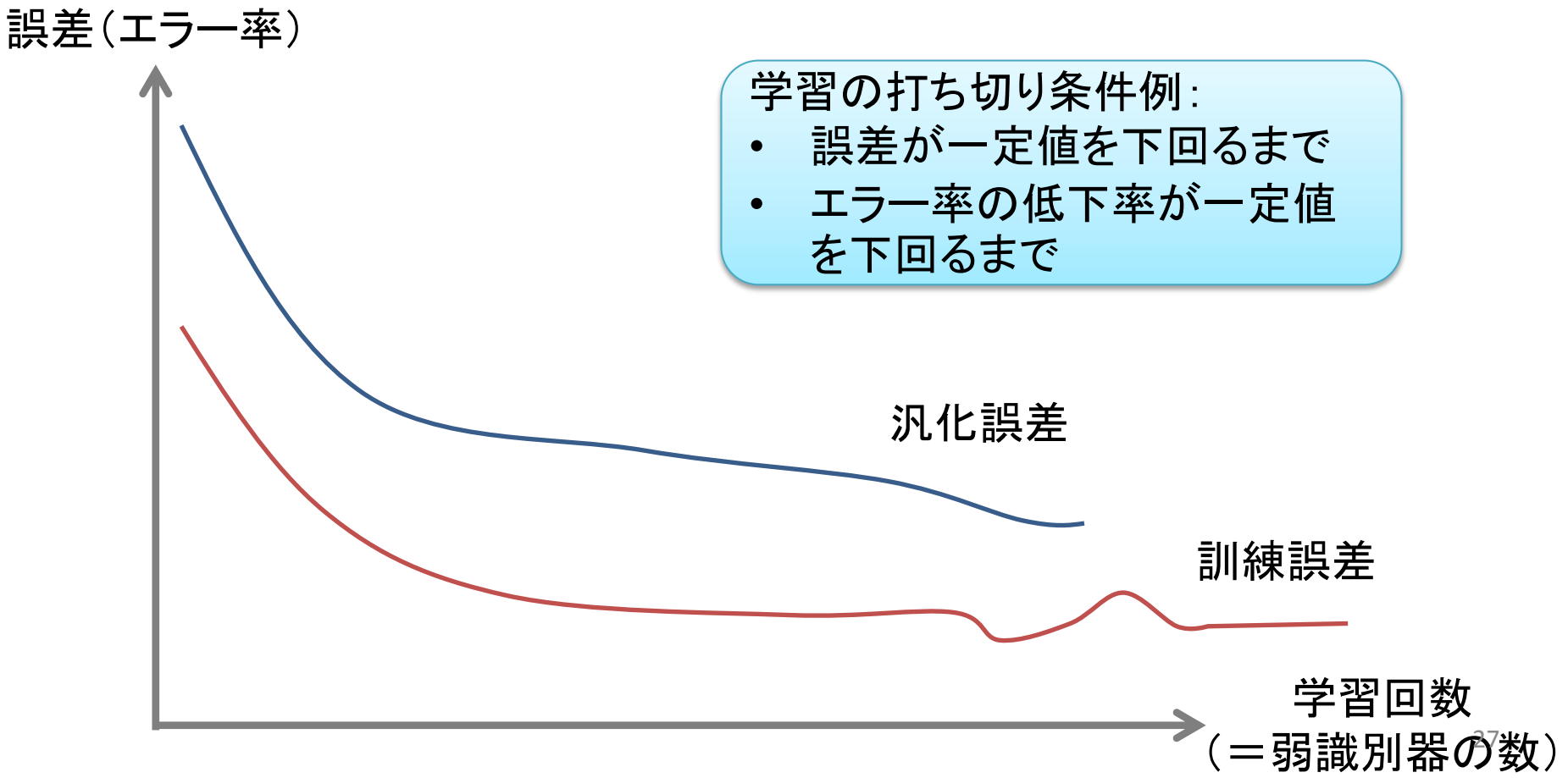
３．１：弱識別器の選択の打ち切り

- 学習サンプルに対する強識別器の正解率を観察しながら、適当な回数で弱識別器の選択を打ち切る
- 打ち切り条件は自由



訓練誤差と汎化誤差

- 訓練誤差：学習サンプルに対する認識誤差
- 汎化誤差：未学習のサンプルに対する認識誤差



AdaBoostのポイント

- 弱識別器をたくさん用いて、強識別器を作成

- 2つ目以降の弱識別器を作成する際には、前回までの学習によって識別が困難であるタイプの物が選ばれる（そうなるように、サンプルの重みが更新されている）

産総研での取り組み

学習用センサデータ作成例

台本：
5秒おきに、「立つ」と「すわる」を繰り返す

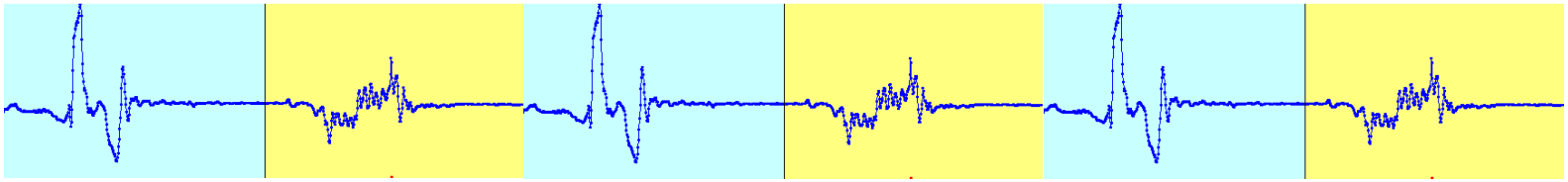
～動作の流れ～
30:00:000～30:05:000：すわる
30:05:000～30:10:000：立つ
30:10:000～30:15:000：すわる
⋮ ⋮

…      …

t

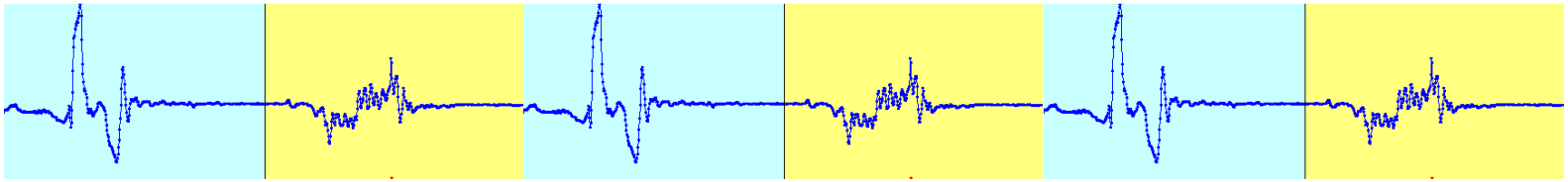
Adaboostを用いたデータの2分類（概要）

0. データを用意する



・・・

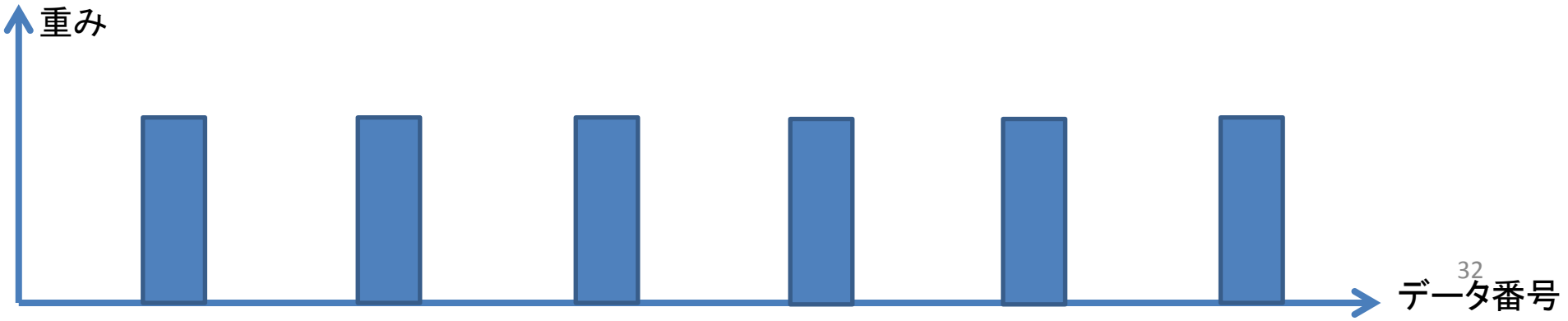
1. データ毎に「重み」を定義して、重みを付与する（最初は全て同じ大きさを付与する）



・・・

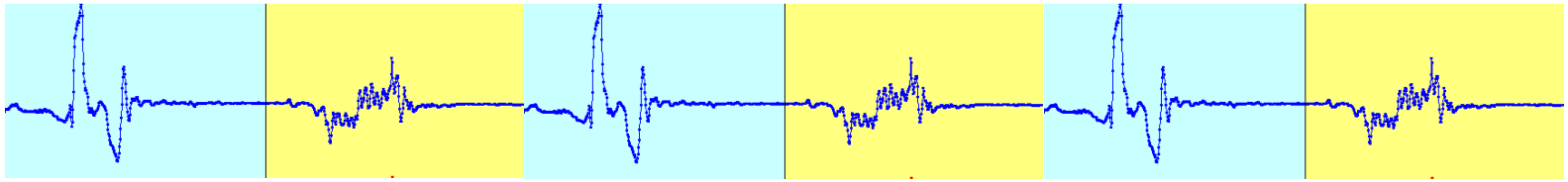
w1　w2　w3　w4　w5　w6　・・・

(w1=w2=w3=w4=w5=w6= ・・・　)



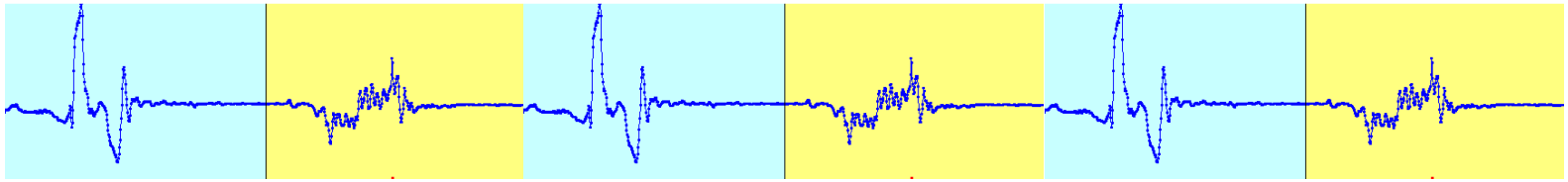
Adaboostを用いたデータの2分類（概要）

2. 正解データを用意する（「立つ」ならば1、「すわる」ならば-1、を付与する）



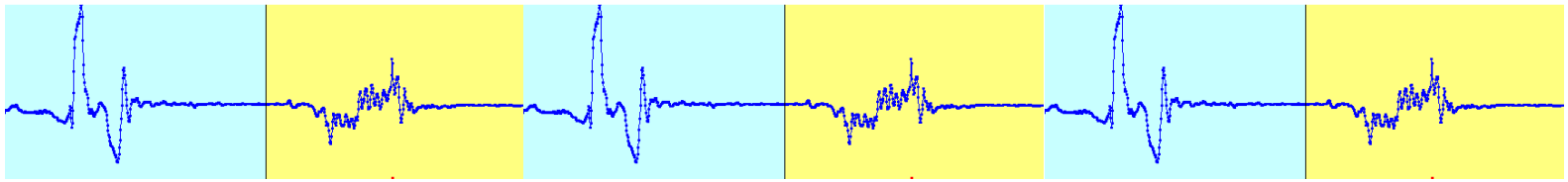
1 -1 1 -1 1 -1 …

3. 弱識別器を作成し、正解の推定を行う（「立つ」ならば1、「すわる」ならば-1、を付与する）



1 -1 1 1 -1 -1 …

4. 弱識別器の正解状況を調べる（正解ならば1、不正解ならば0を付与する）

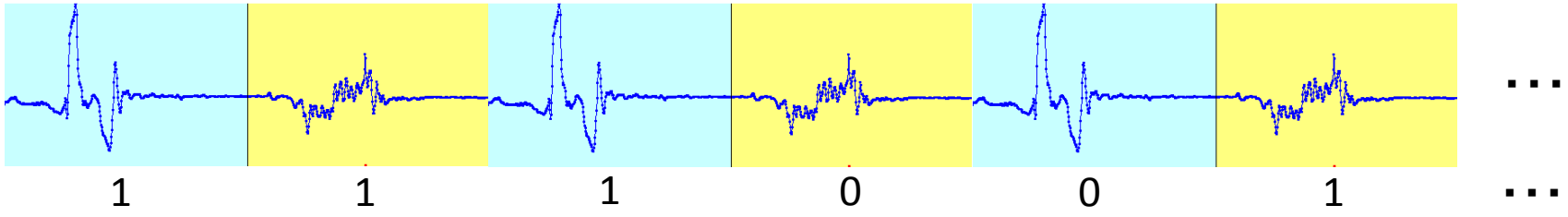


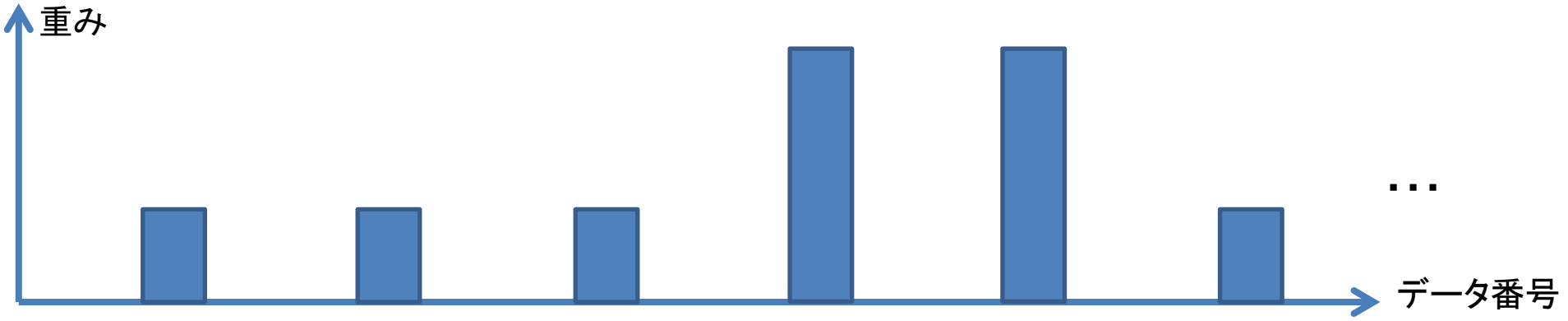
1 1 1 0 0 1 …

Adaboostを用いたデータの2分類（概要）

5. 弱識別器の信頼度を計算する（信頼度は、推定に失敗したデータの重み和に依存）

6. 重みを更新する（推定が正解ならば小さくし、不正解ならば大きくする）





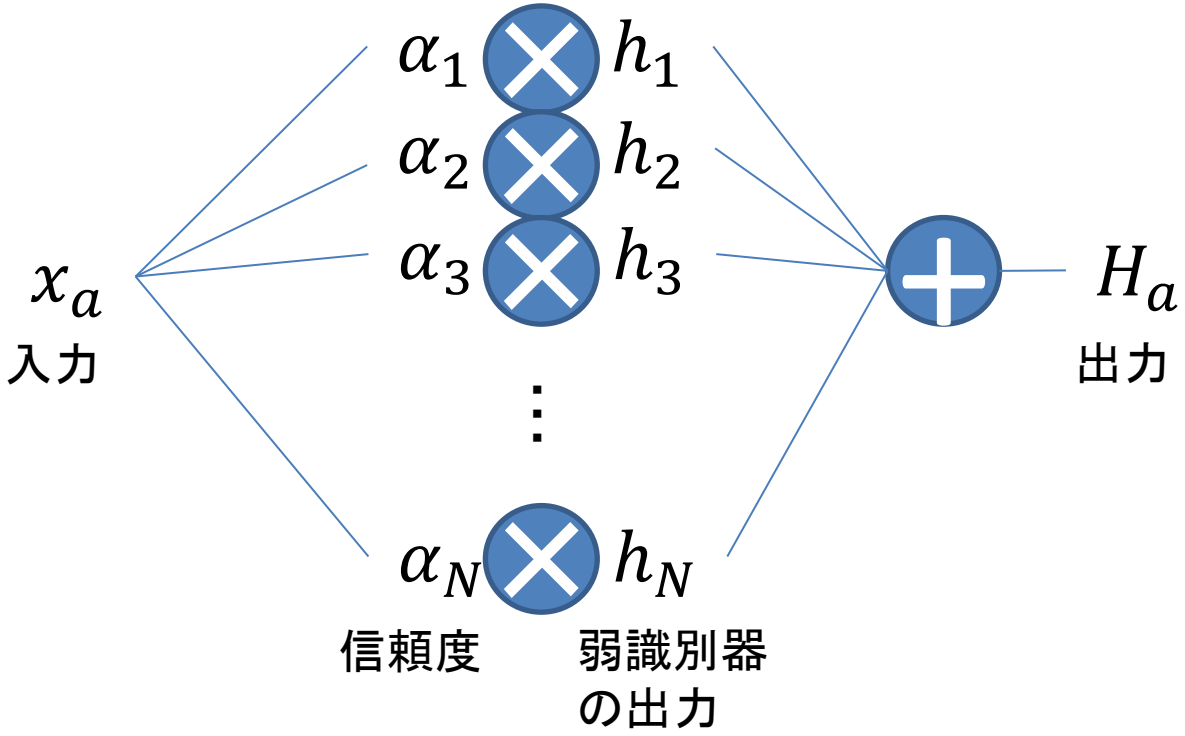
7. 3に戻り、新たな弱識別器を作成する（幾つかの弱識別器を作成後、次のステップへ）

Adaboostを用いたデータの2分類（概要）

8. 強識別器の作成

$$H_a = sign \sum_{t=1}^{N} (\alpha_t h_t(x_a))$$

強識別器の出力：
各識別器の識別結果を足し合わせた値の符号



誤検出と未検出

- 通常はセットで考える
- 場合によっては、どちらかを重視することも・・・

学習サンプルについて

- サンプルが多いに越したことは無い
- サンプルを作る際には、セグメンテーション作業が必要なことが多い（結構大変）
- ネガティブサンプルを作るのは、意外と大変