

VSAM

動画像理解技術基礎講座

Video Understanding Technology and Its Application

Object
Detection

Segmentation

Object
Tracking

Sep.5 2008

Mapping

藤吉弘亘 Hironobu Fujiyoshi

Feature
Extraction

中部大学工学部情報工学科

E-mail: hf@cs.chubu.ac.jp

Object
Classification

URL: <http://www.vision.cs.chubu.ac.jp/>

VSAM

1. 画像理解技術を用いた次世代ビデオ監視システム: VSAM

Object
Detection

2. 移動体検出: Object Detection

Segmentation

3. 領域クラスタリング: Segmentation

Object
Tracking

4. 移動体追跡: Object Tracking

Mapping

5. 画像から実空間へのマッピング: Mapping to world coordinate

Feature
Extraction

6. 特徴抽出: Feature Extraction

Object
Classification

7. 物体識別: Object Classification

物体識別: Object classification

VSAM

- ・ 講義内容

Object
Detection

- 形状特徴量の抽出

- 識別器の構築

Segmentation

- ・ 線形判別関数

- ・ マハラノビス距離

Object
Tracking

- Adaboostによる識別

Mapping

- ニューラルネットワークによる識別

Feature
Extraction

- k-NN法による色推定

Object
Classification

- Part-based approach

物体識別における問題点:

VSAM

- ・ 同じクラスであっても、異なる色を持つ場合や照明環境が異なる場合で見え方が変化



Object
Detection

Segmentation

Object
Tracking

Mapping

- ・ 同一物体においても姿勢の違いにより見え方が変化



Feature
Extraction

Object
Classification

物体識別における問題点:

VSAM

- ・ 同じクラスであっても、異なる色を持つ場合や照明環境が異なる場合で見え方が変化



Object
Detection

Segmentation

Object
Tracking

Mapping

- ・ 同一物体においても姿勢の違いにより見え方が変化



Feature
Extraction

Object
Classification

→見え方の変化に影響を受けない特徴量の抽出が重要

形状特徴量の抽出:

- ・ 形状の複雑度
 - 輪郭線の凹凸に着目

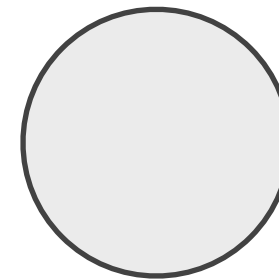
$$\text{複雑度 (dispersedness)} = \frac{\text{周囲長}^2}{\text{面積}}$$

周囲長：輪郭線を構成するピクセル数

面積：検出ピクセル数



複雑度：76.5



複雑度： 4π (最小値)

人と自動車の複雑度:

VSAM

Object
Detection

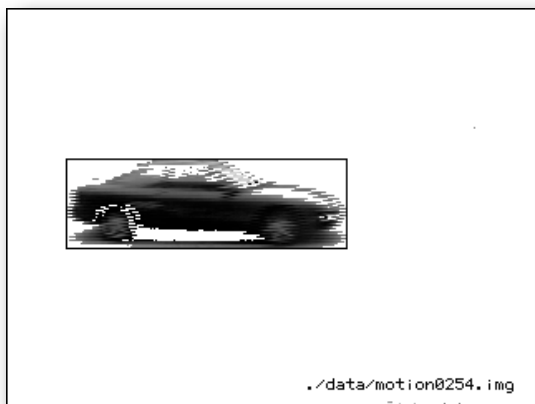
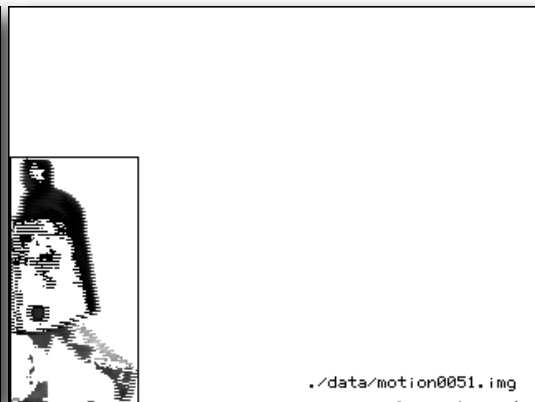
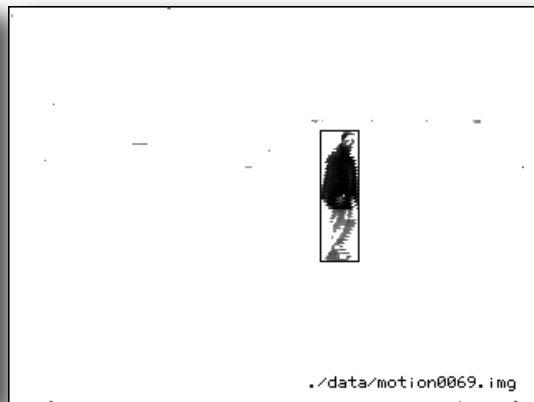
Segmentation

Object
Tracking

Mapping

Feature
Extraction

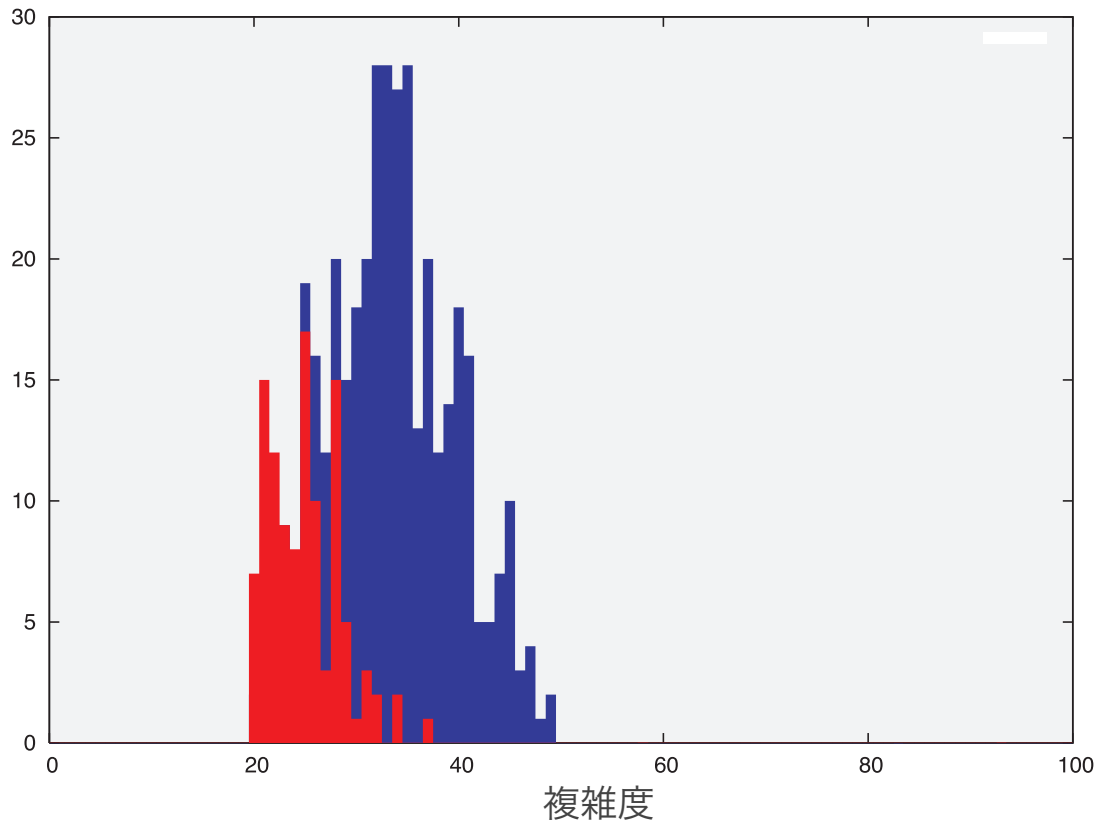
Object
Classification



複雑度のヒストグラム:

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

ヒストグラム



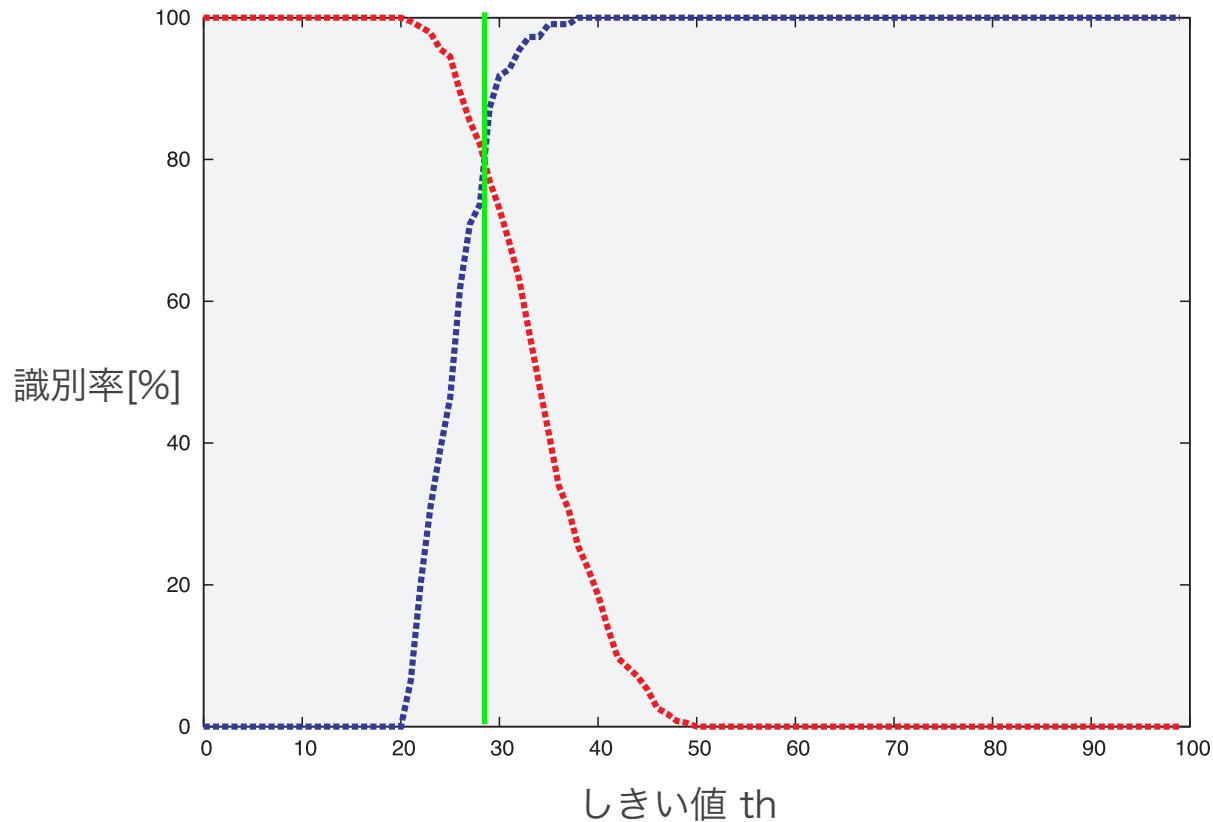
人(169パターン)：平均は高く、広い範囲に分布

車(44パターン)：平均は低い

しきい値処理による識別:

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

- ・ 複雑度によるしきい値処理
 - 複雑度 $< th$: 人クラスに属する
 - 複雑度 $> th$: 車クラスに属する



th=51 のとき
識別率 : 83.7 [%]

高精度な識別を実現するには:

VSAM

- ・ 特徴量の追加
 - クラス間の違いを表す特徴量の追加

Object
Detection

Segmentation

- ・ 多量の学習サンプルによる識別器の構築

Object
Tracking

- 線形判別関数
- マハラノビス距離
- ニューラルネットワーク(NN)
- サポートベクタマシン(SVM)

Mapping

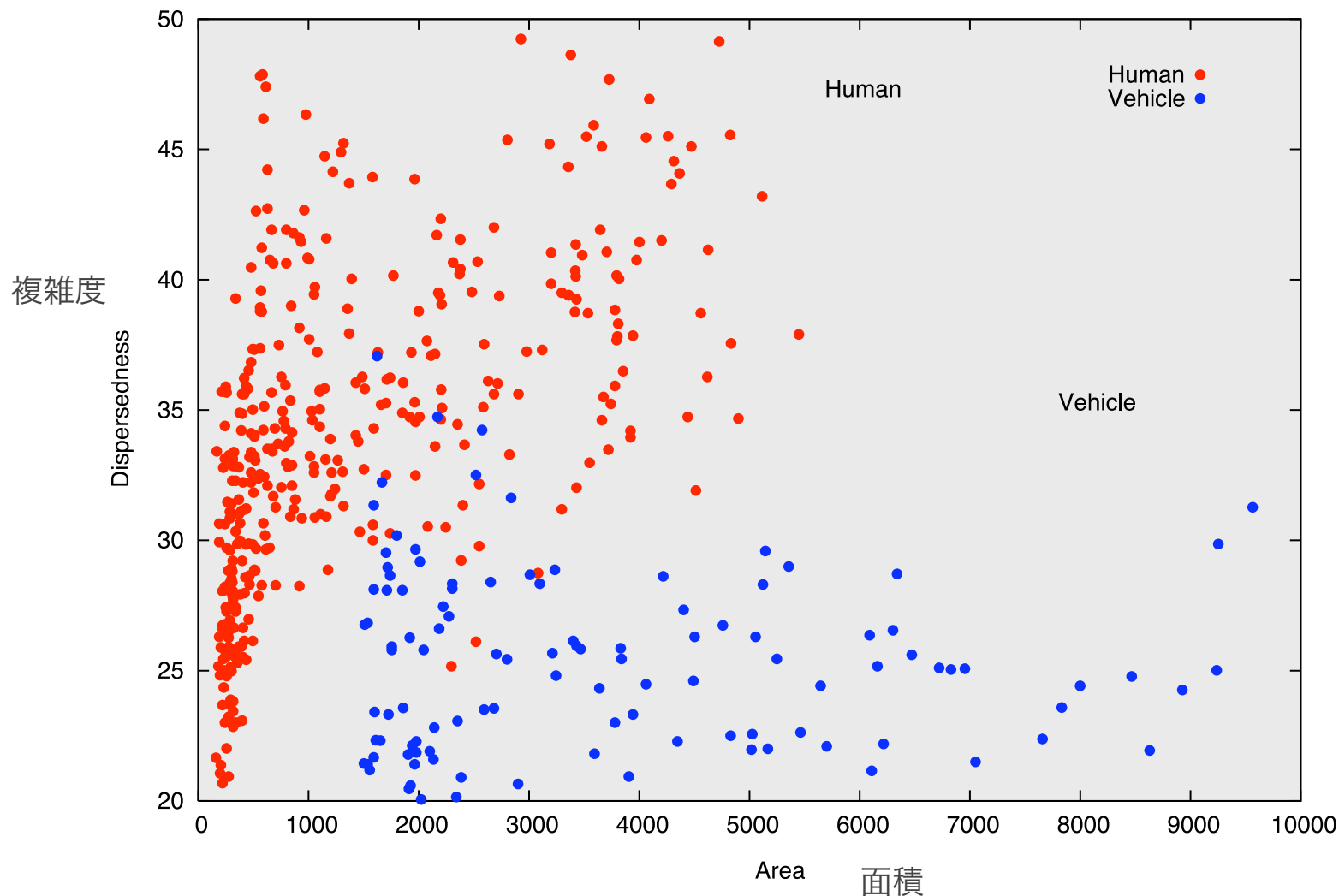
Feature
Extraction

- AdaBoost

Object
Classification

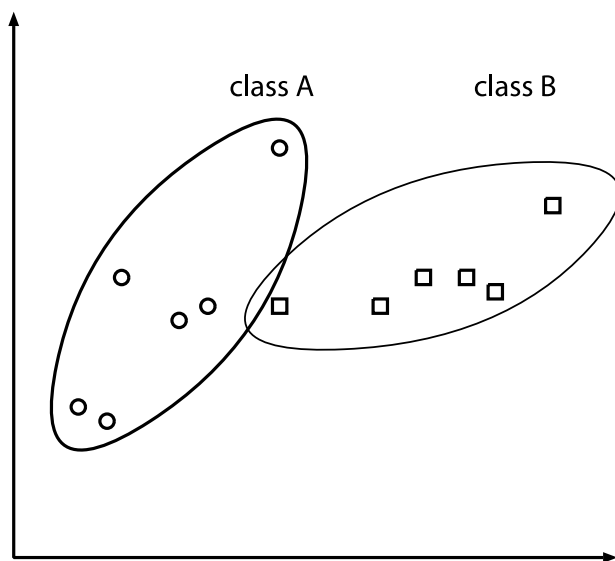
2変数(複雑度-面積)によるクラス分布:

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

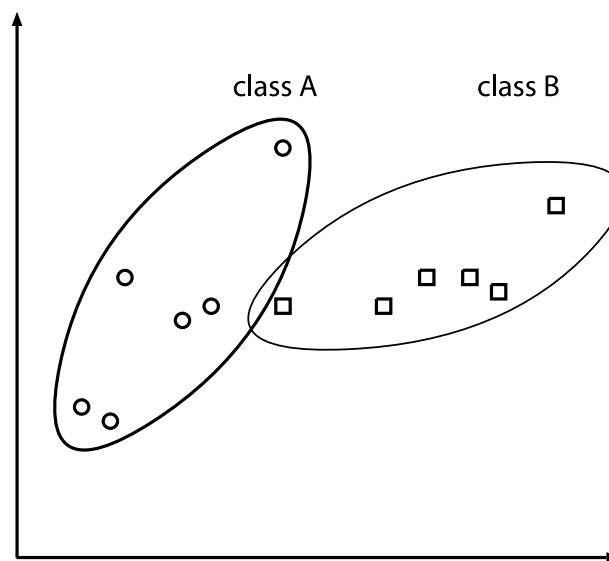


線形判別関数とマハラノビス距離:

- 学習サンプルデータから各クラスを分離する境界線を求める
 - 線形判別関数 (一次式による境界線)
 - マハラノビス距離 (二次式による境界線)



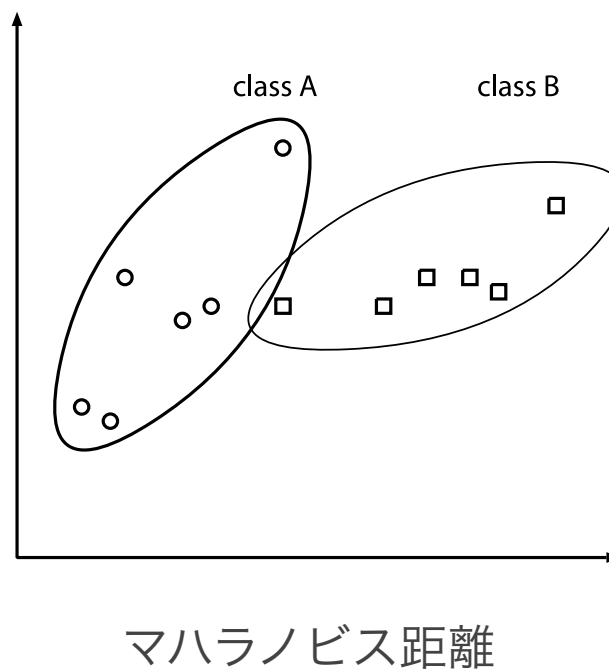
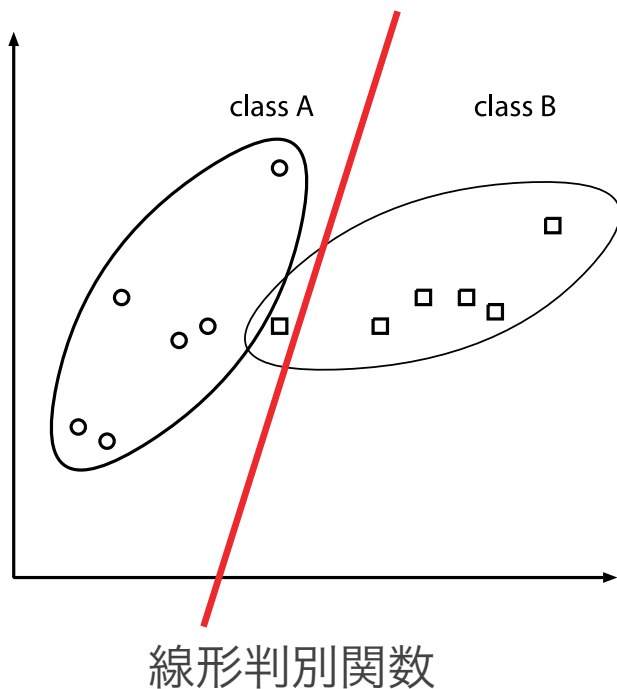
線形判別関数



マハラノビス距離

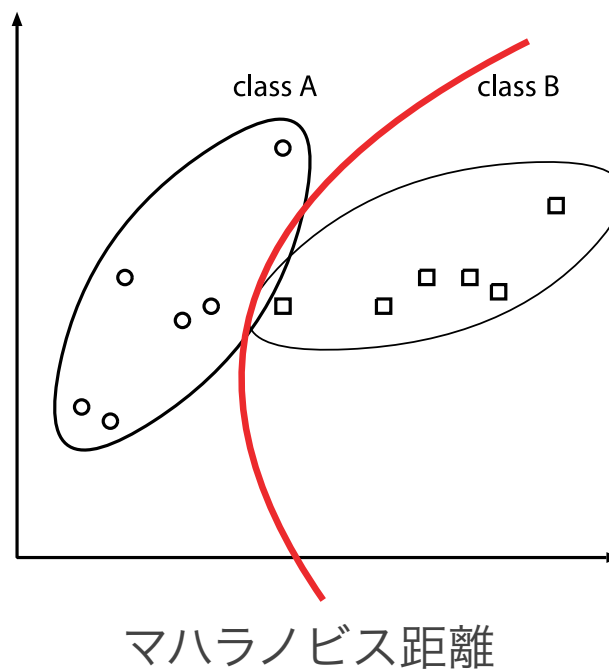
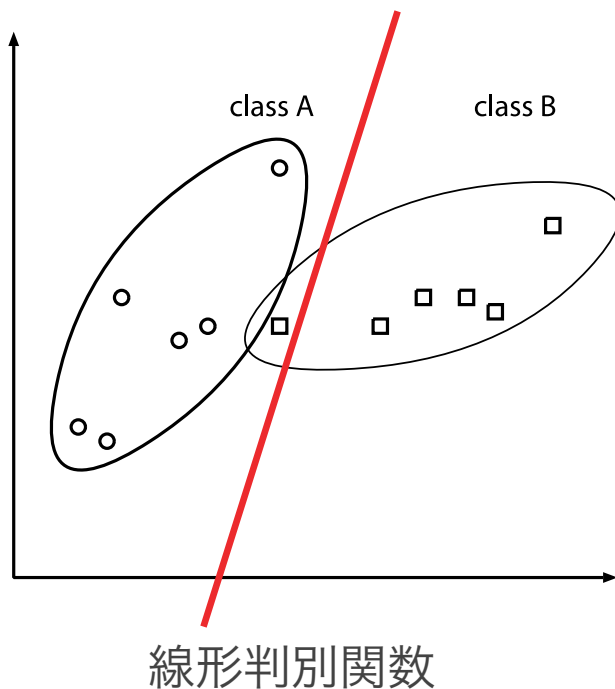
線形判別関数とマハラノビス距離:

- 学習サンプルデータから各クラスを分離する境界線を求める
 - 線形判別関数 (一次式による境界線)
 - マハラノビス距離 (二次式による境界線)



線形判別関数とマハラノビス距離:

- 学習サンプルデータから各クラスを分離する境界線を求める
 - 線形判別関数 (一次式による境界線)
 - マハラノビス距離 (二次式による境界線)



線形判別関数: Linear Discriminant Function (LDF)

線形判別関数

$$z = a_1 x_1 + a_2 x_2 + a_0$$

$$a_1 = \frac{1}{s_{11}s_{22} - s_{12}^2} \left\{ s_{22} \left(\frac{\sum x_{1i}^{(1)}}{N_1} - \frac{\sum x_{1i}^{(2)}}{N_2} \right) - s_{12} \left(\frac{\sum x_{2i}^{(1)}}{N_1} - \frac{\sum x_{2i}^{(2)}}{N_2} \right) \right\}$$

$$a_2 = \frac{1}{s_{11}s_{22} - s_{12}^2} \left\{ -s_{12} \left(\frac{\sum x_{1i}^{(1)}}{N_1} - \frac{\sum x_{1i}^{(2)}}{N_2} \right) + s_{11} \left(\frac{\sum x_{2i}^{(1)}}{N_1} - \frac{\sum x_{2i}^{(2)}}{N_2} \right) \right\}$$

$$a_0 = -\frac{1}{2} \left(\frac{a_1 \sum x_{1i}^{(1)} + a_2 \sum x_{2i}^{(1)}}{N_1} + \frac{a_1 \sum x_{1i}^{(2)} + a_2 \sum x_{2i}^{(2)}}{N_2} \right)$$

学習用サンプルから a_1, a_2, a_0 を決定

オンライン識別

$$z = -0.002382592 \times area + 0.027149614 \times dispersedness - 0.44778073$$

線形判別関数: Linear Discriminant Function (LDF)

線形判別関数

$$z = a_1 x_1 + a_2 x_2 + a_0$$

$$a_1 = \frac{1}{s_{11}s_{22} - s_{12}^2} \left\{ s_{22} \left(\frac{\sum x_{1i}^{(1)}}{N_1} - \frac{\sum x_{1i}^{(2)}}{N_2} \right) s_{12} \left(\frac{\sum x_{2i}^{(1)}}{N_1} - \frac{\sum x_{2i}^{(2)}}{N_2} \right) \right\}$$

$$a_2 = \frac{1}{s_{11}s_{22} - s_{12}^2} \left\{ s_{12} \left(\frac{\sum x_{1i}^{(1)}}{N_1} - \frac{\sum x_{1i}^{(2)}}{N_2} \right) s_{11} \left(\frac{\sum x_{2i}^{(1)}}{N_1} - \frac{\sum x_{2i}^{(2)}}{N_2} \right) \right\}$$

$$a_0 = -\frac{1}{2} \left(\frac{a_1 \sum x_{1i}^{(1)} + a_2 \sum x_{2i}^{(1)}}{N_1} + \frac{a_1 \sum x_{1i}^{(2)} + a_2 \sum x_{2i}^{(2)}}{N_2} \right)$$

学習用サンプルから a_1, a_2, a_0 を決定

オンライン識別

$$z = -0.002382592 \times area + 0.027149614 \times dispersedness - 0.44778073$$

$$\begin{cases} z > 0 & : \text{クラス 2(human) に属する} \\ z < 0 & : \text{クラス 1(vehicle) に属する} \end{cases}$$

マハラノビス距離: Mahalanobis Generalized Distance

マハラノビス距離:

ユークリッド距離を標準偏差で割った値の二乗

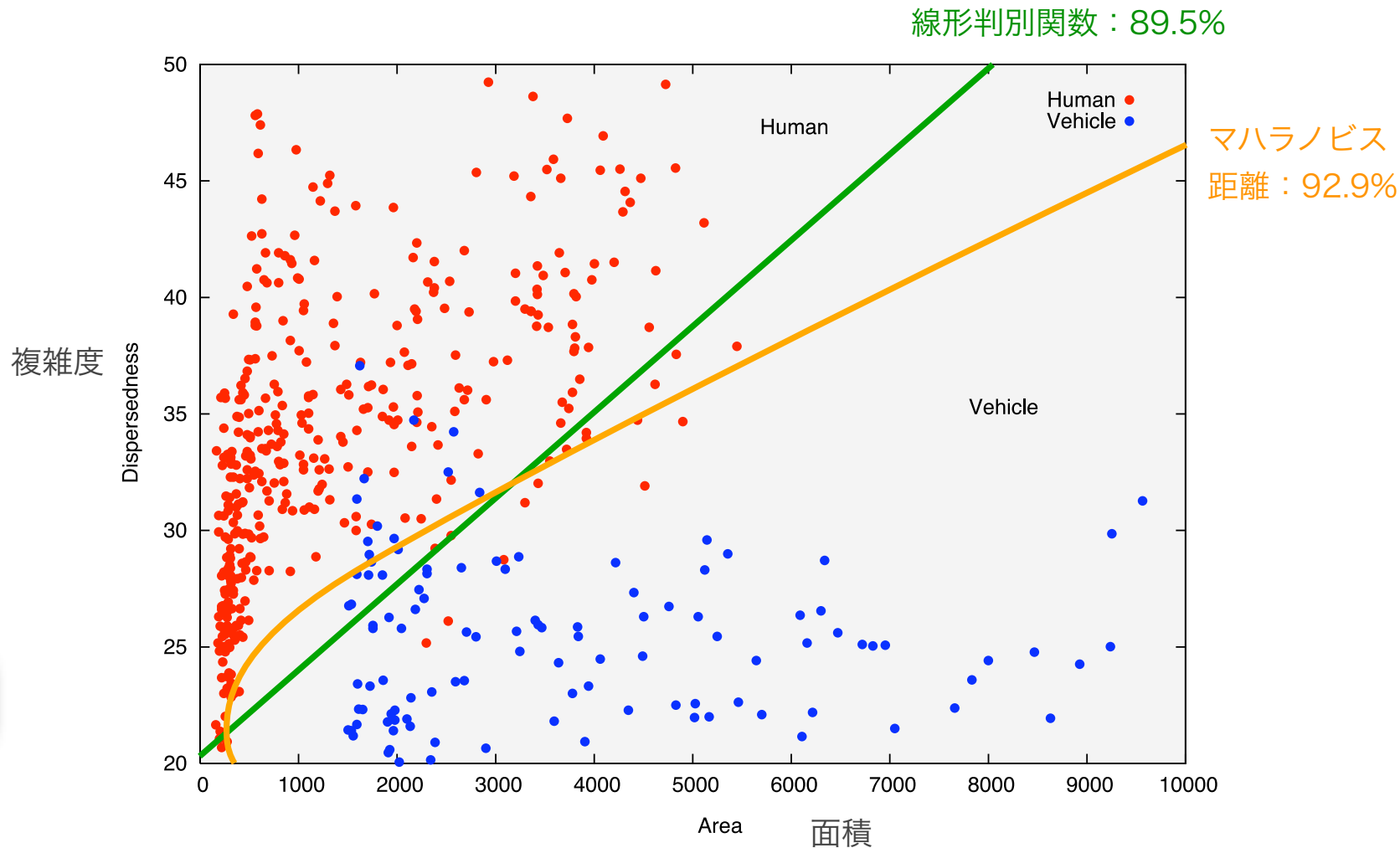
$$D_1^2 = \frac{1}{1 - \rho_{(1)}^2} \left(\frac{x_1 - \mu_1^{(1)}}{\sigma_{1(1)}} \right)^2 + \left(\frac{x_2 - \mu_2^{(1)}}{\sigma_{2(1)}} \right)^2 - 2\rho_{(1)} \left(\frac{x_1 - \mu_1^{(1)}}{\sigma_{1(1)}} \right) \left(\frac{x_2 - \mu_2^{(1)}}{\sigma_{2(1)}} \right)$$
$$D_2^2 = \frac{1}{1 - \rho_{(2)}^2} \left(\frac{x_1 - \mu_1^{(2)}}{\sigma_{1(2)}} \right)^2 + \left(\frac{x_2 - \mu_2^{(2)}}{\sigma_{2(2)}} \right)^2 - 2\rho_{(2)} \left(\frac{x_1 - \mu_1^{(2)}}{\sigma_{1(2)}} \right) \left(\frac{x_2 - \mu_2^{(2)}}{\sigma_{2(2)}} \right)$$

オンライン識別

$$\begin{cases} D_2^2 > D_1^2 & : \text{クラス 2 (human) に属する} \\ D_2^2 < D_1^2 & : \text{クラス 1 (vehicle) に属する} \end{cases}$$

LGFとMGDによる識別境界:

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification



ニューラルネットワークによる識別: ANN

VSAM

Object
Detection

Segmentation

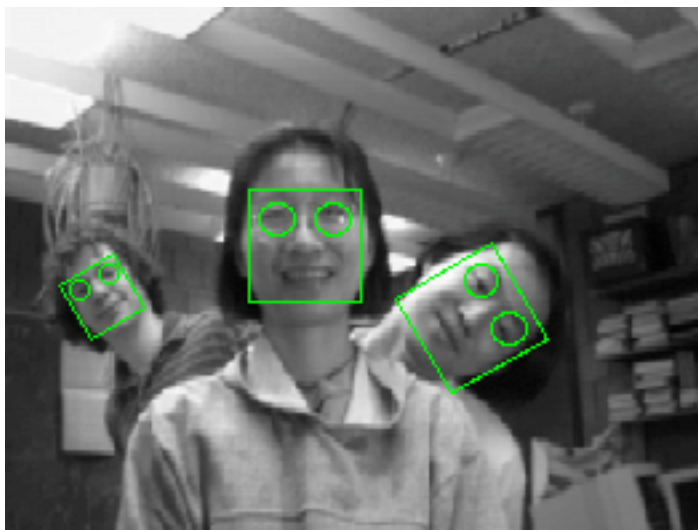
Object
Tracking

Mapping

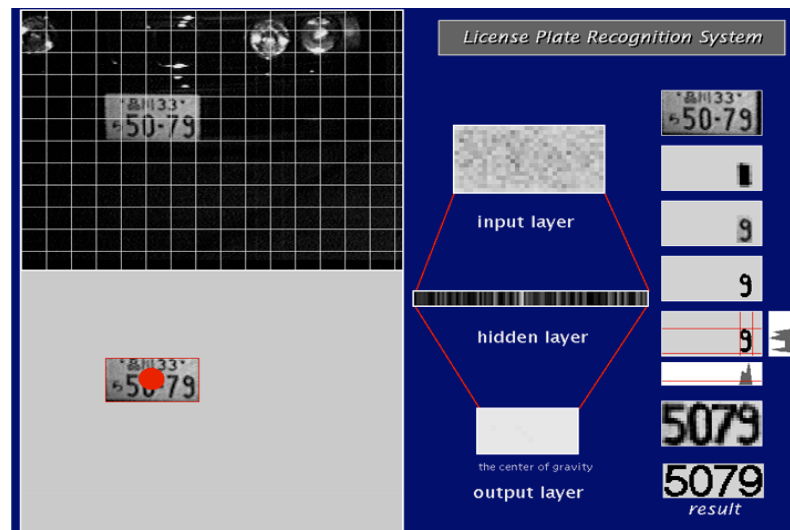
Feature
Extraction

Object
Classification

- ニューラルネットワーク (Artificial Neural Network)
 - 多数の学習サンプル
 - 設計が比較的容易
- 応用例
 - 顔検出, ナンバープレート検出等



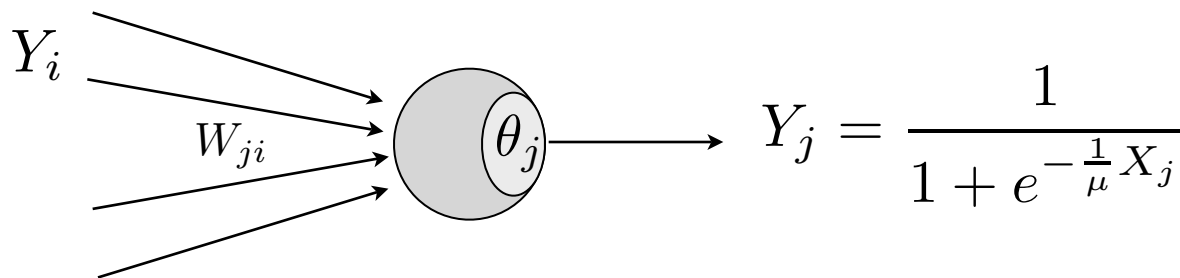
Neural Network-Based Face Detection,
H. Rowley, S. Baluja, and T. Kanade, IEEE PAMI, 1998



ANNによるナンバープレートの位置検出,
藤吉, 梅崎, 今村, 金出, 信学論, Vol.J80-D-II, No.6, 1997

ニューラルネットワーク: ANN

他のユニットからの結合加重に対応した重み付けの出力

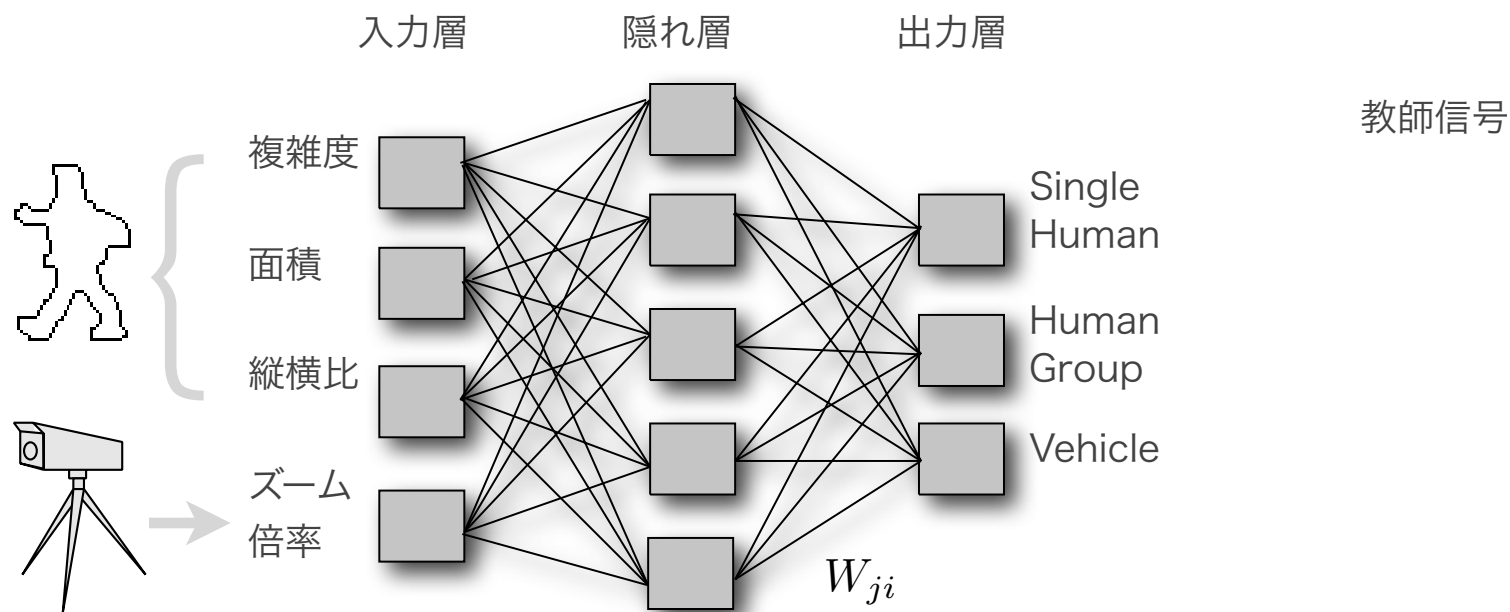


$$X_j = \sum_i W_{ji} \cdot Y_i + \theta_j$$

- X_j : 上層ユニット j の入力総和
- Y_i : 下層ユニット i の出力
- W_{ji} : 下層ユニット i から上層ユニット j への結合係数 (重み)
- θ_j : 上層ユニット j のオフセット
- T_j : 出力層ユニット j における教師信号
- Y_j : 上層ユニット j の出力

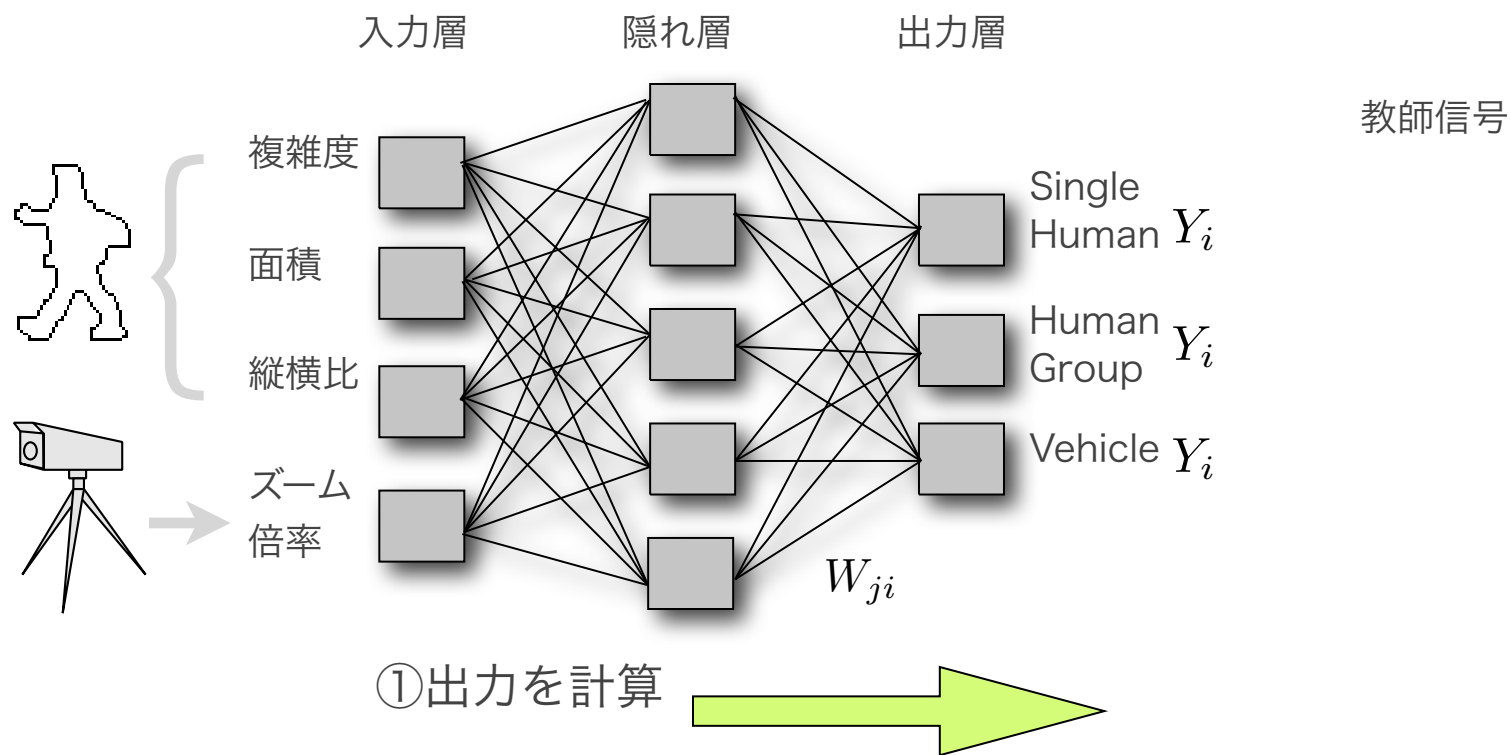
ニューラルネットの学習: ANN

学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



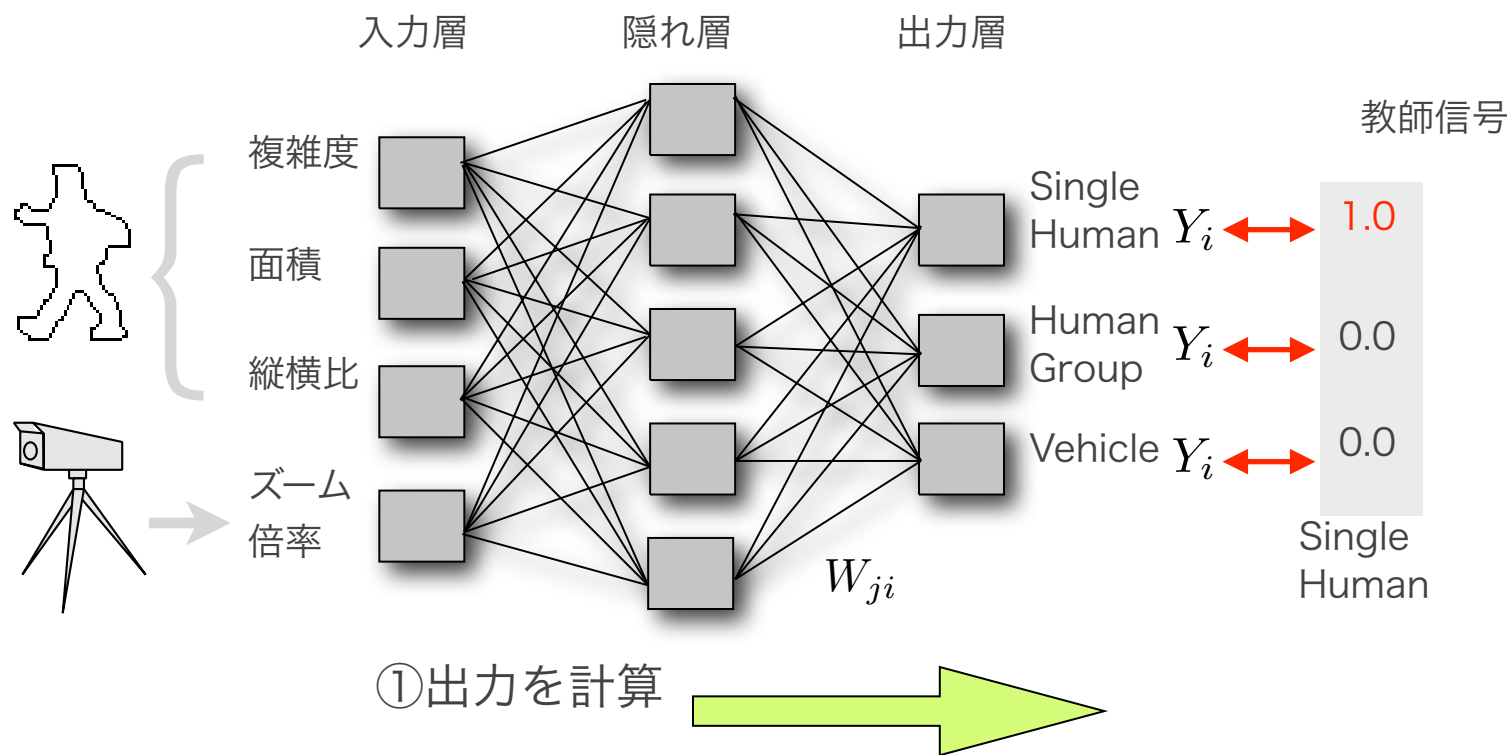
ニューラルネットの学習: ANN

学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



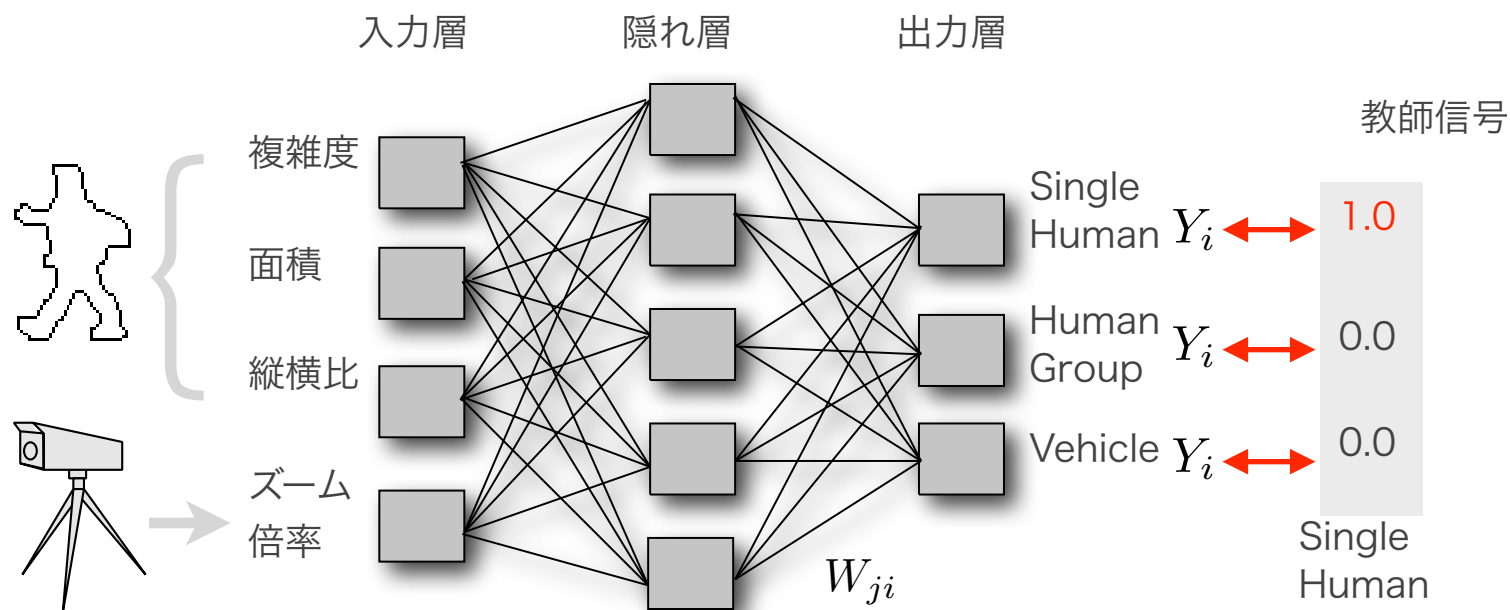
ニューラルネットの学習: ANN

学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



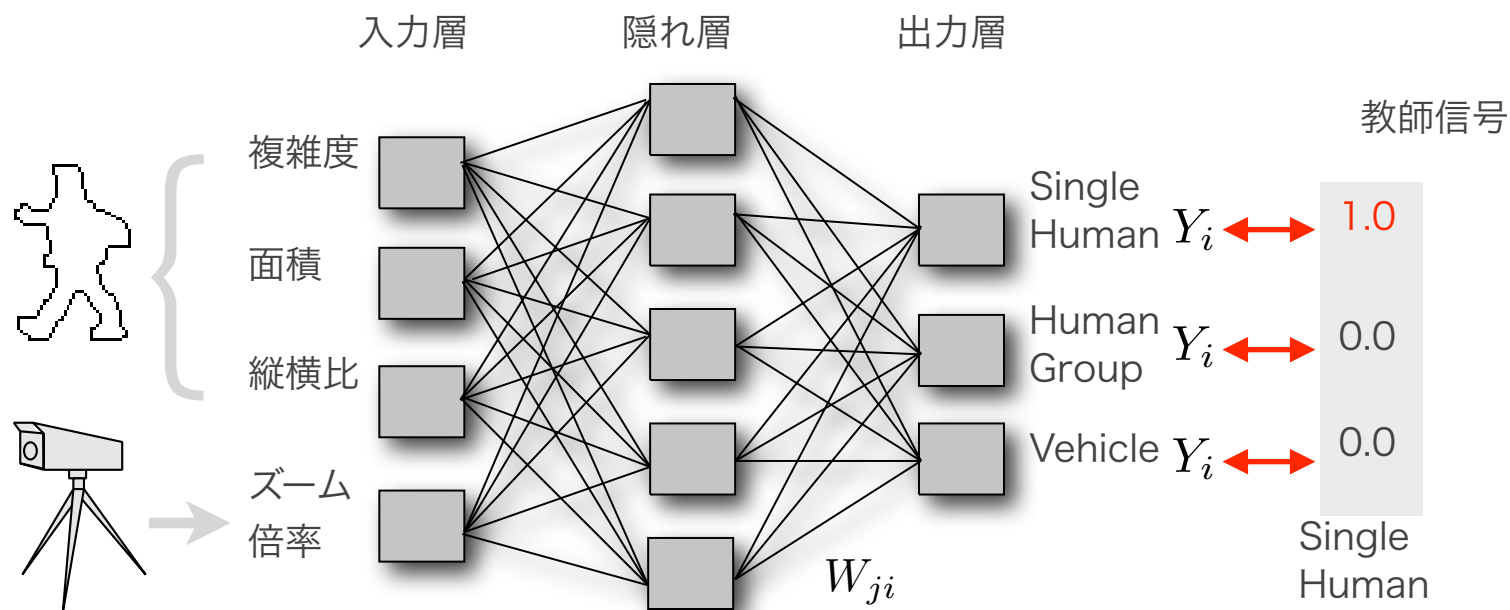
ニューラルネットの学習: ANN

学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



ニューラルネットの学習: ANN

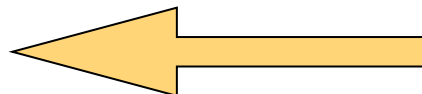
学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



①出力を計算



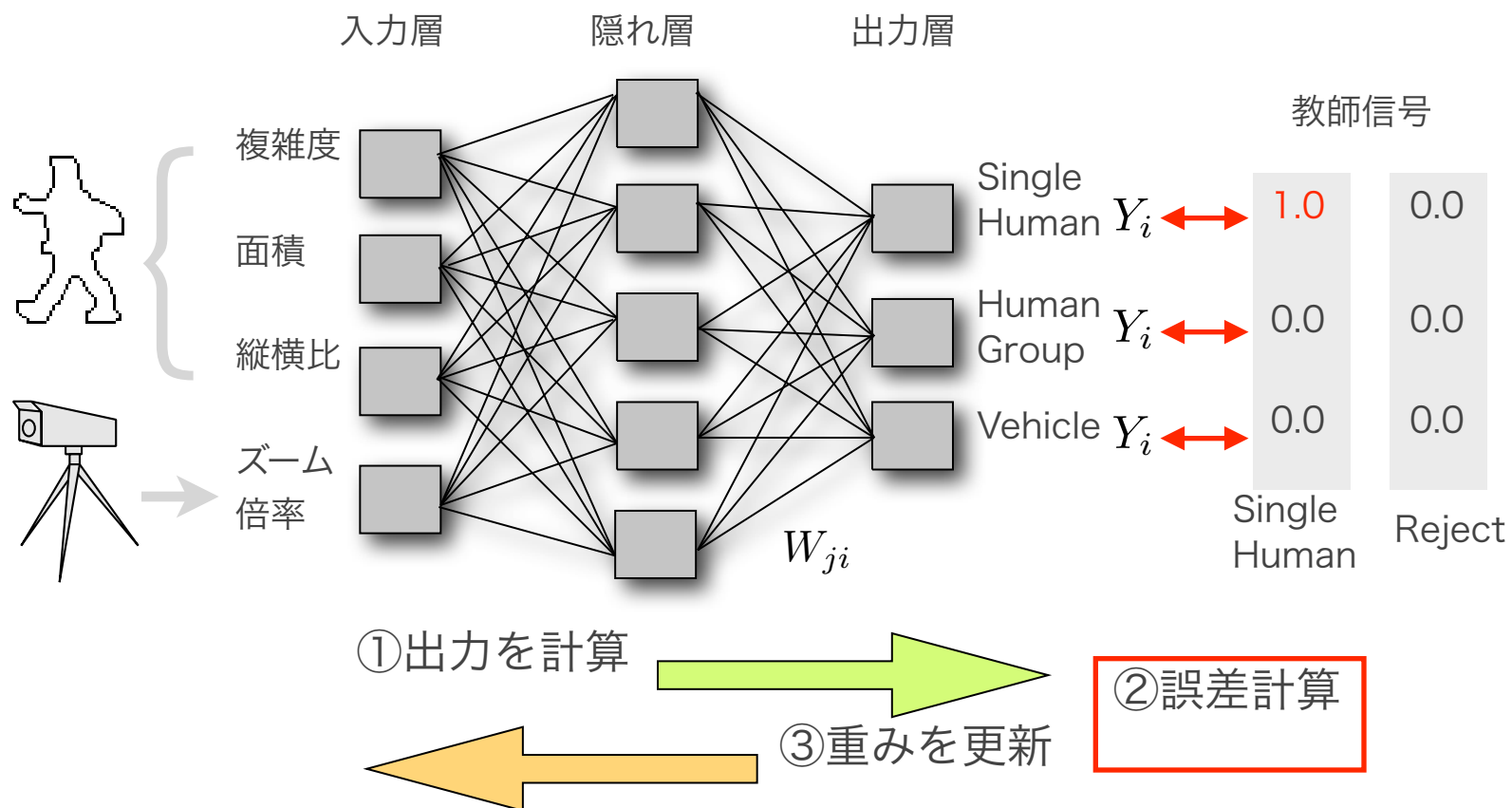
②誤差計算



③重みを更新

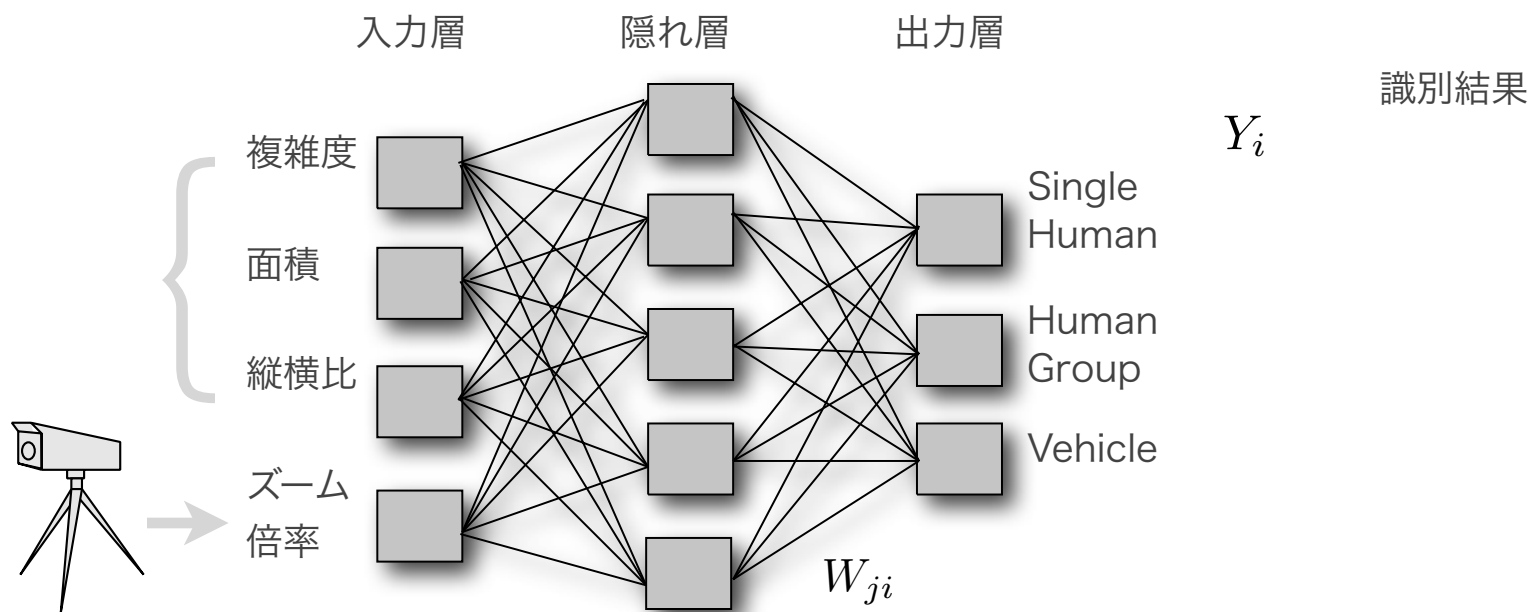
ニューラルネットの学習: ANN

学習：教師信号との誤差を逆伝搬して重み係数の修正を繰り返す



ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算

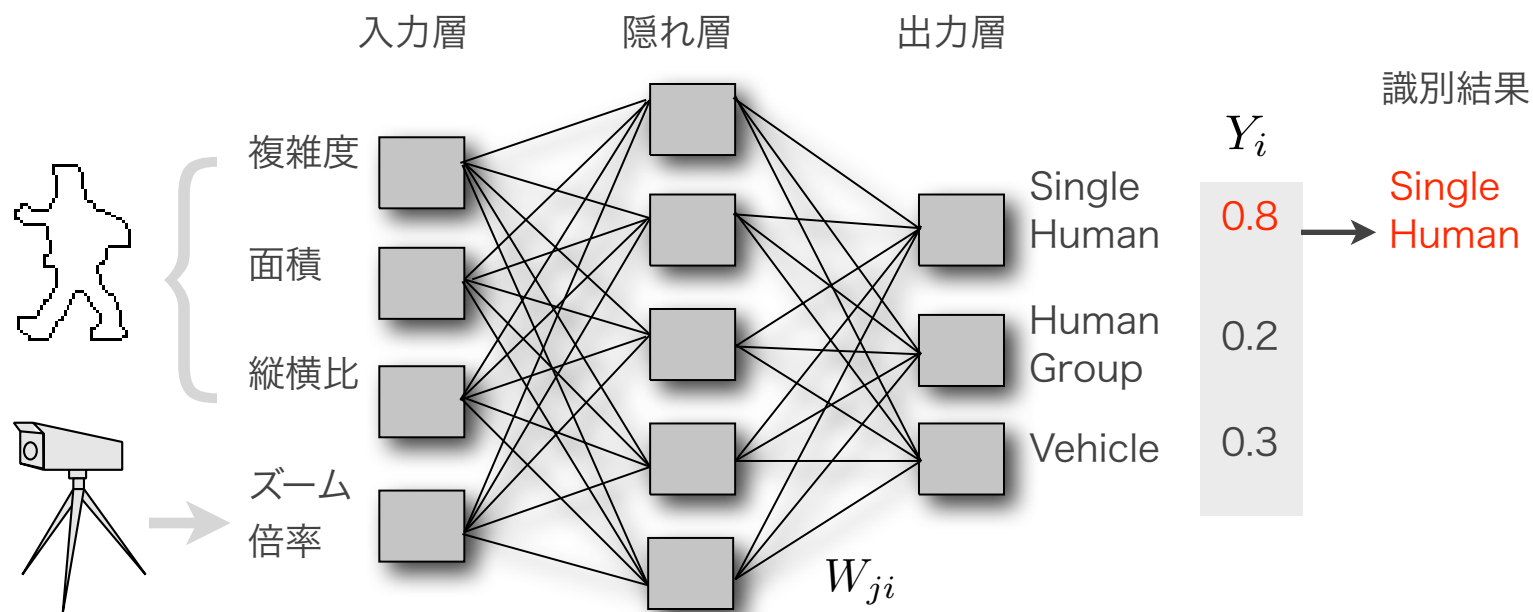


オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算

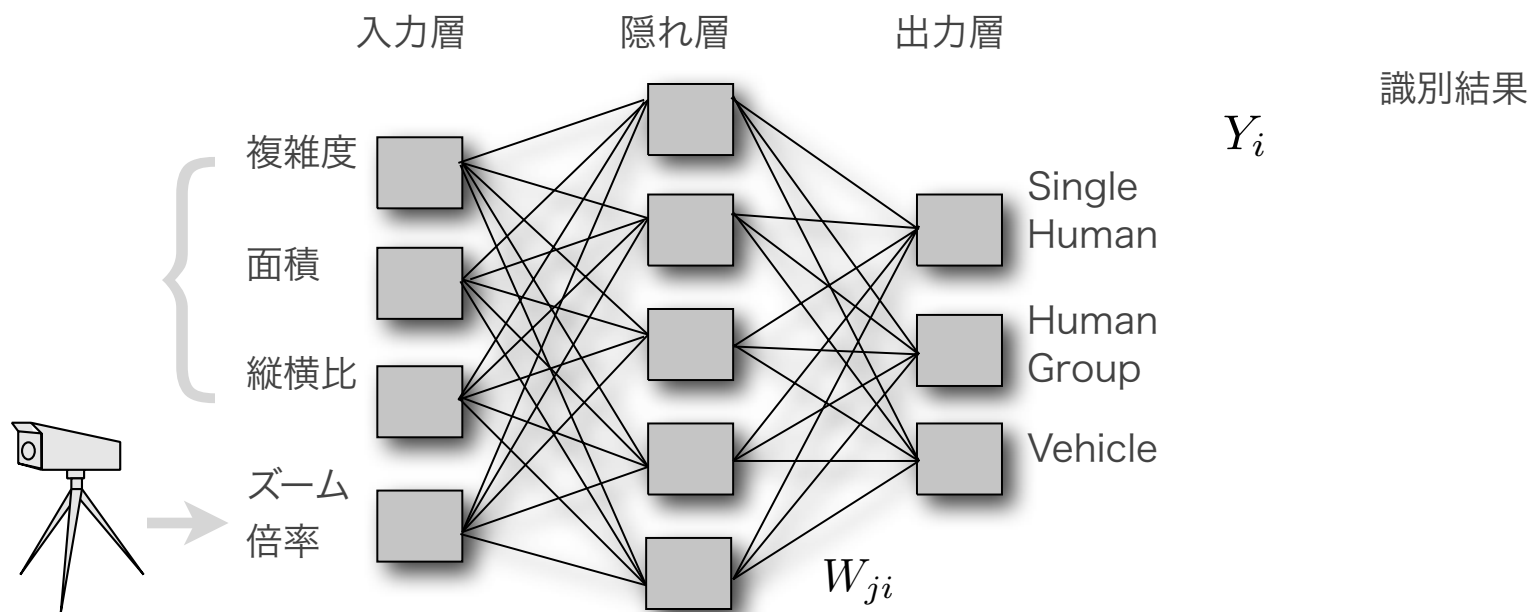


オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算

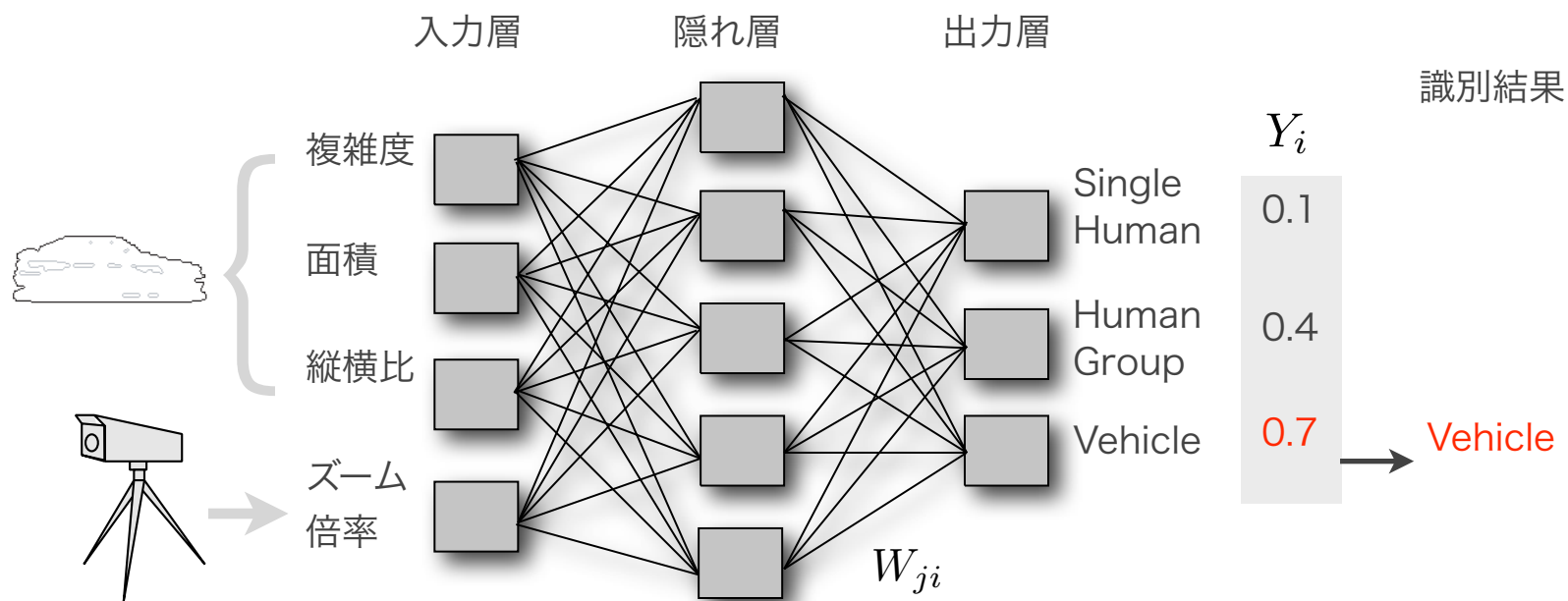


オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算

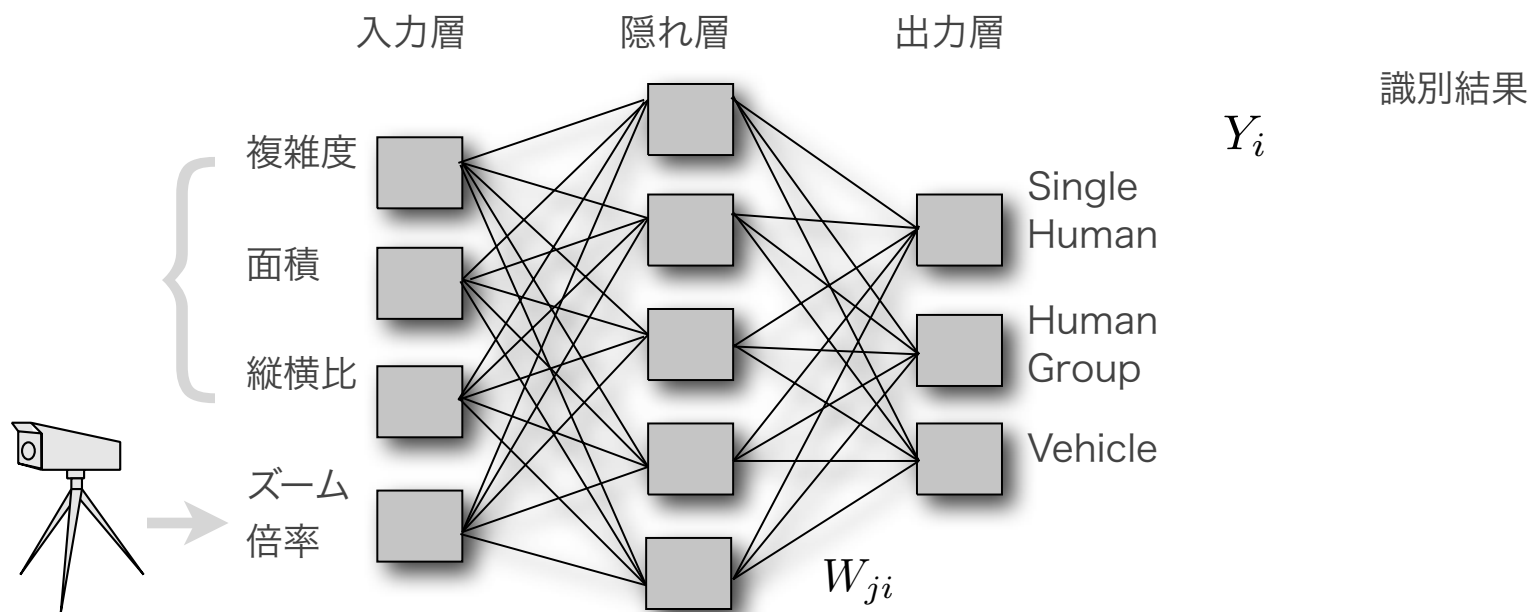


オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算

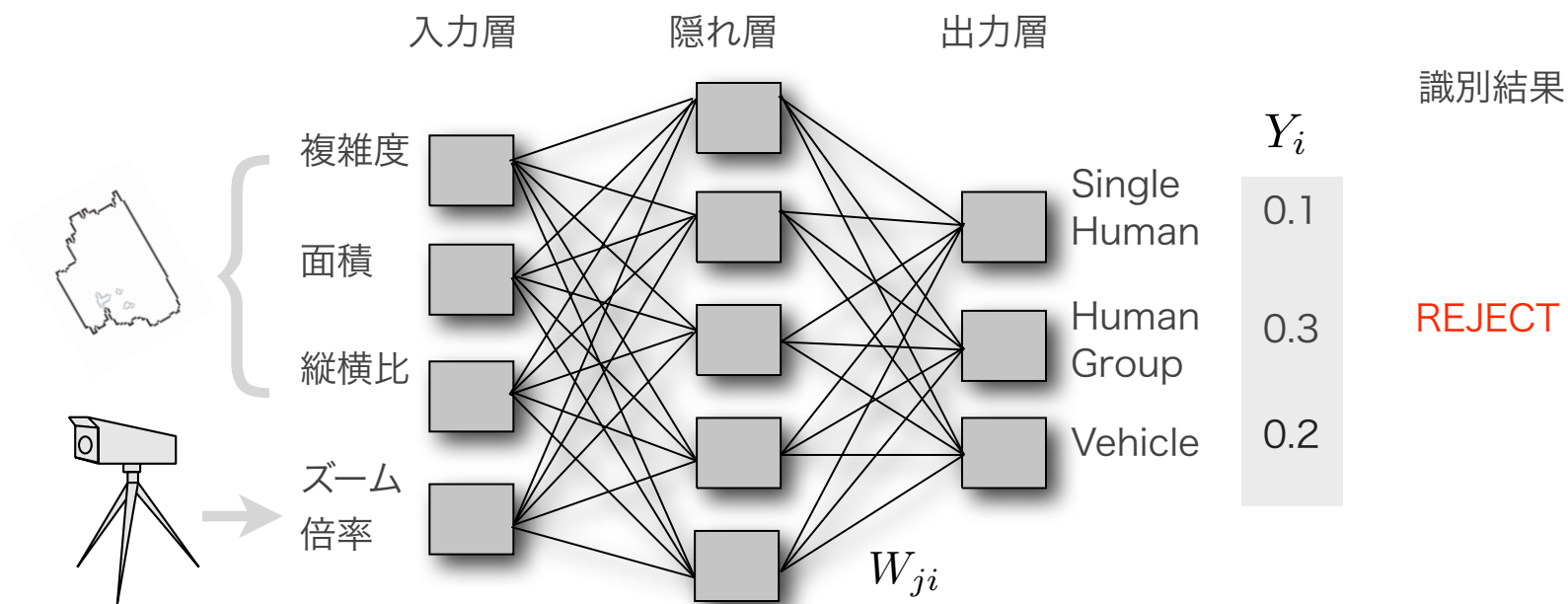


オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

ニューラルネットによる識別:

学習したニューラルネットに検出したパラメータを入力し出力を計算



オンライン判定基準

- 最大値 $> th$ リジェクト
- 最大値 $< th$ 最大値となる出力ユニットに対応するクラスに属する

N.N.による識別例:

VSAM

Object
Detection

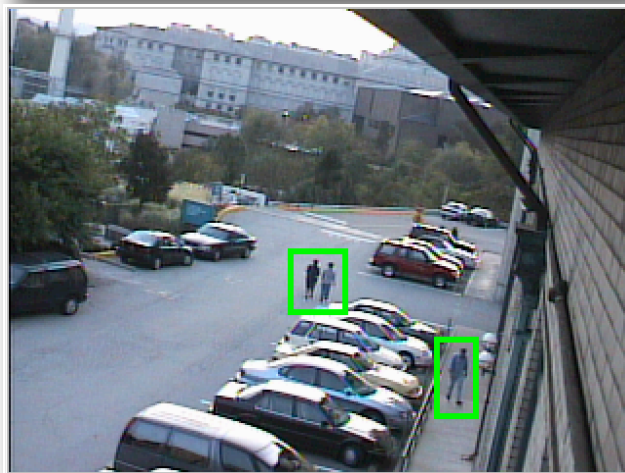
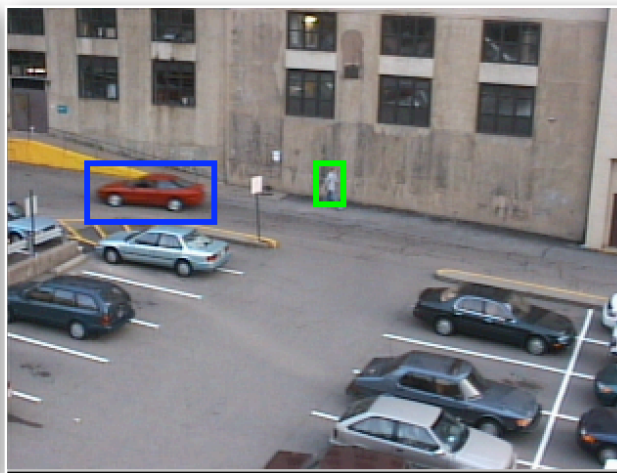
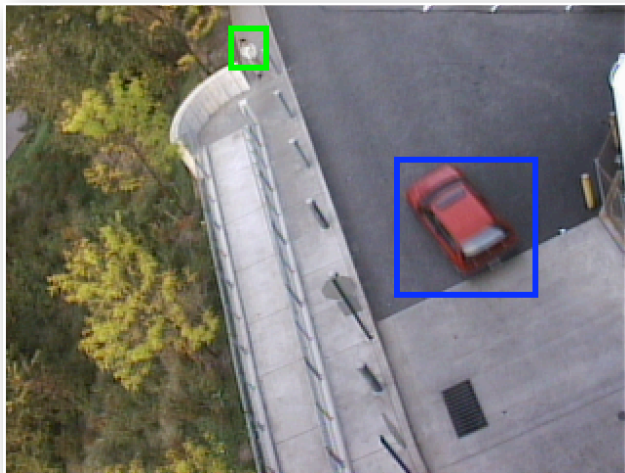
Segmentation

Object
Tracking

Mapping

Feature
Extraction

Object
Classification



識別率 Single Human : 91.1[%]
Vehicle : 99.2[%]

Human Group : 85.9[%]
Reject : 67.4[%]

Support Vector Machinesによる識別: SVM

VSAM

- ・ マージン最大化により高精度な識別関数を導く手法

Object Detection

$$\text{sgn}(f(\mathbf{x})) = \begin{cases} \geq 0, & 1 \\ < 0, & -1 \end{cases}$$

$$\begin{aligned} f(x) &= \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \\ &= \sum_{i=1}^n w_i x_i + b \end{aligned}$$

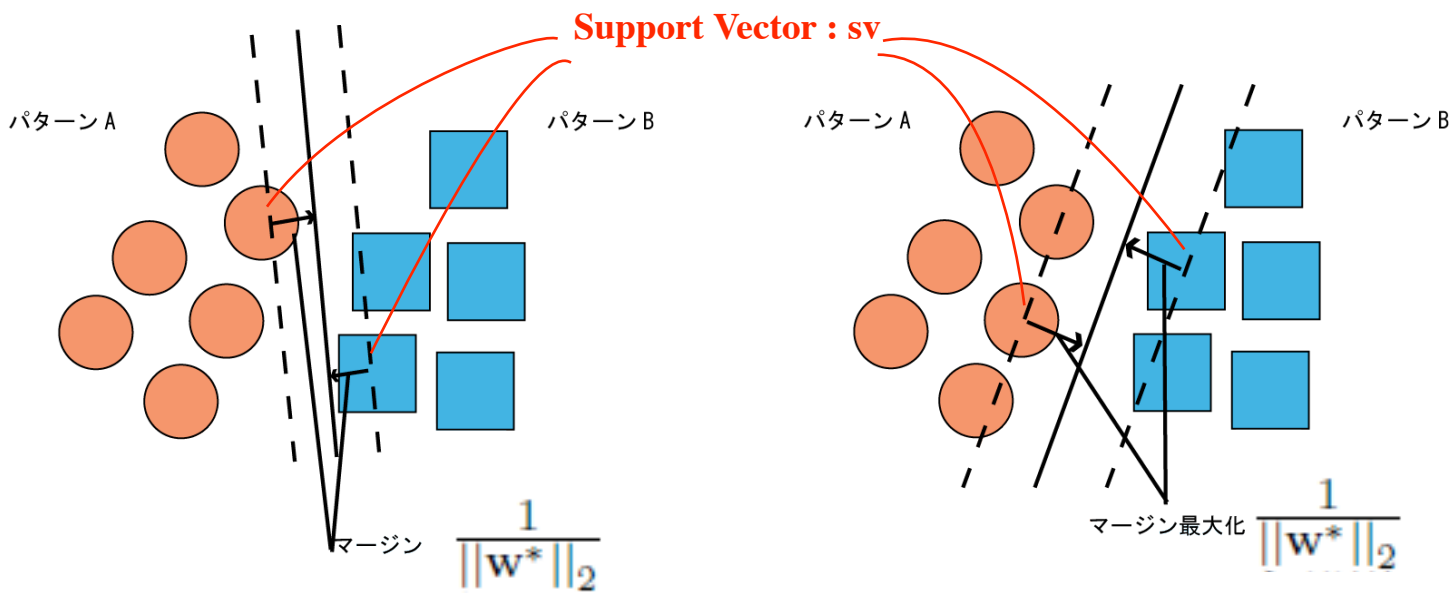
Segmentation

Object Tracking

Mapping

Feature Extraction

Object Classification



(a) マージン小

(b) マージン大

最大マージンSVM: SVM

- マージンを最大化

与えられた線形分離可能なトレーニング標本

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n))$$

に対し, 超平面 (\mathbf{w}, b) は最適化問題

$$\begin{aligned} & \text{minimise}_{\mathbf{w}, b} \quad \langle \mathbf{w} \cdot \mathbf{w} \rangle, \\ & \text{subject to} \quad y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \\ & \quad \quad \quad i = 1, \dots, n. \end{aligned}$$

の最適解であるとする. すると, 幾何マージン $\gamma = \frac{1}{\|\mathbf{w}\|_2}$ をもつ最大マージンクラス分類器を実現する.

- ラグランジの未定乗数法によるラグランジアン

$$\begin{aligned} L(\mathbf{w}, b, \alpha) &= \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^n \alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \\ &= \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \sum_{i=1}^n \alpha_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \end{aligned}$$

- ラグランジ乗数 α について最適化

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

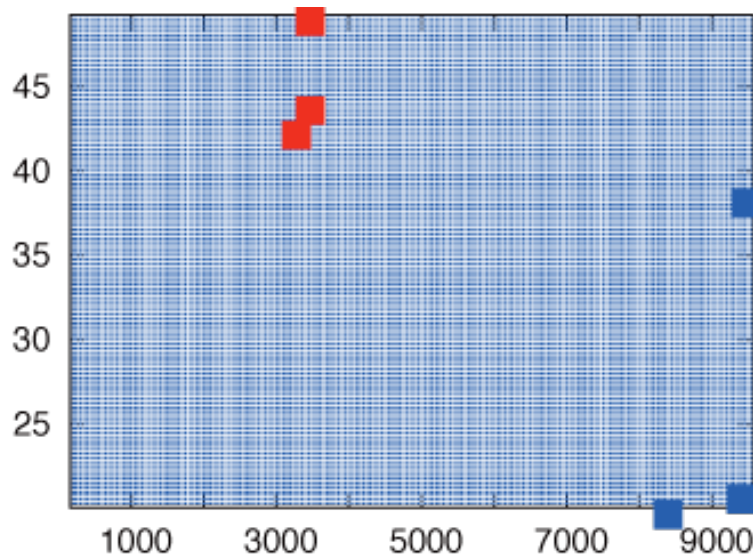
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(a)初期状態

$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

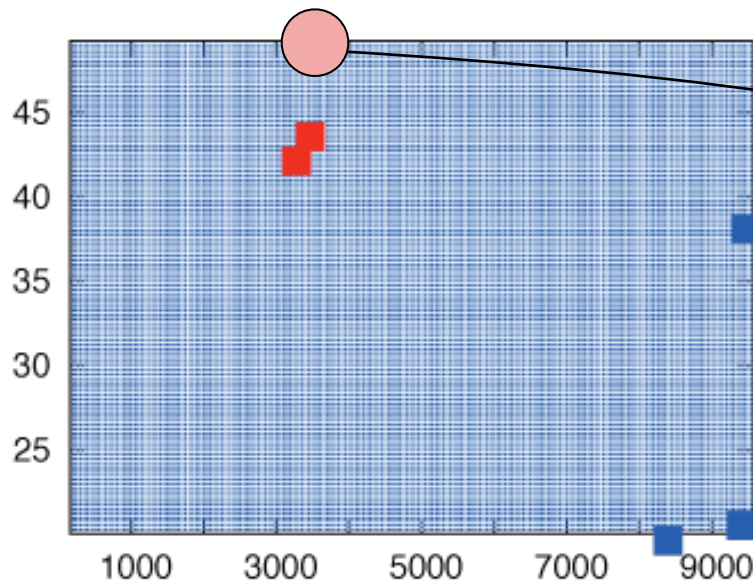
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

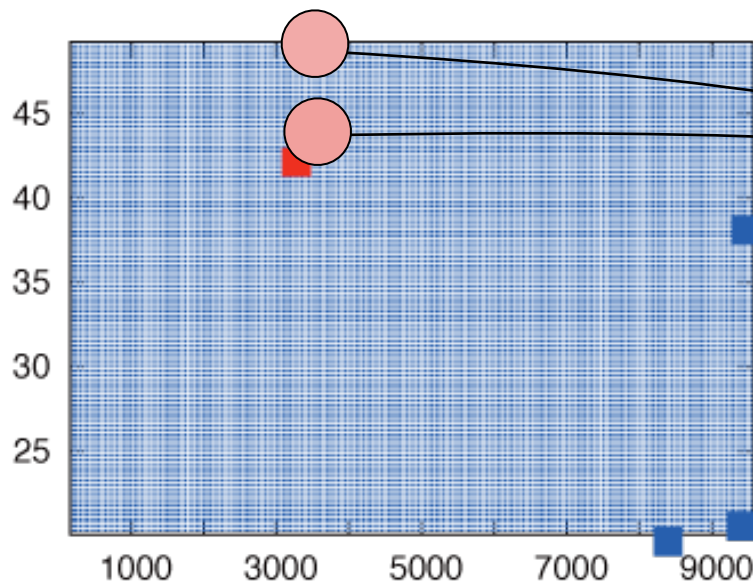
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

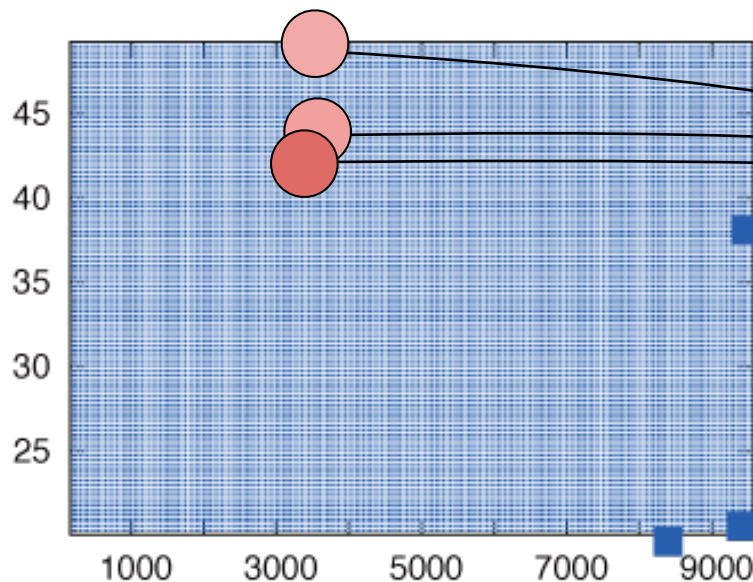
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i \left(1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i \right)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

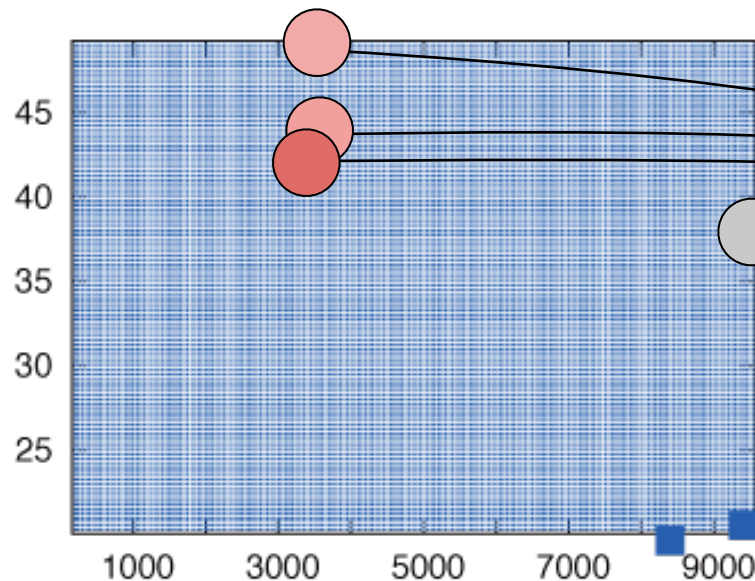
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x)$ $\left\{ \begin{array}{l} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{array} \right.$

(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

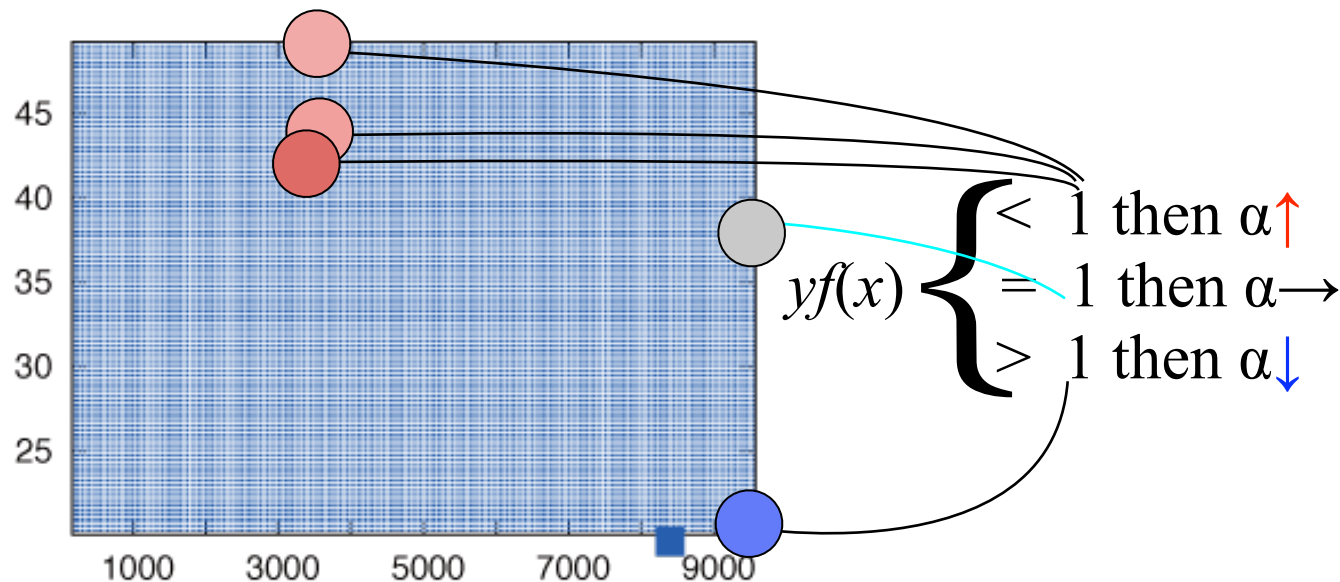
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

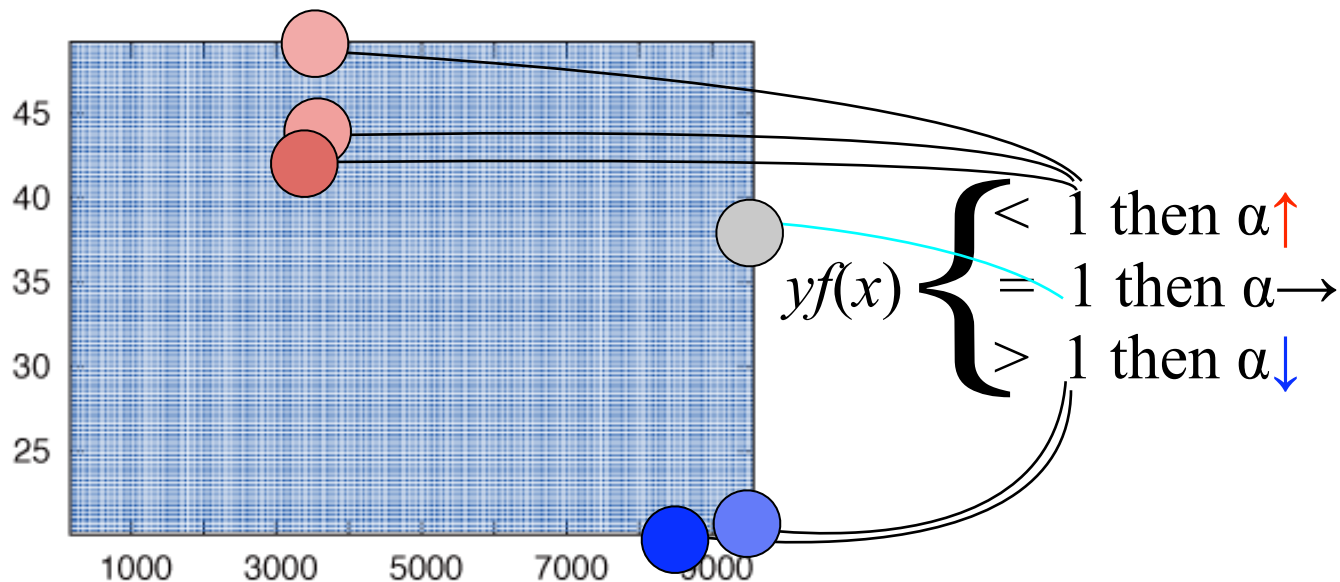
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(a)初期状態

再急降下法による最大マージンSVM: SVM

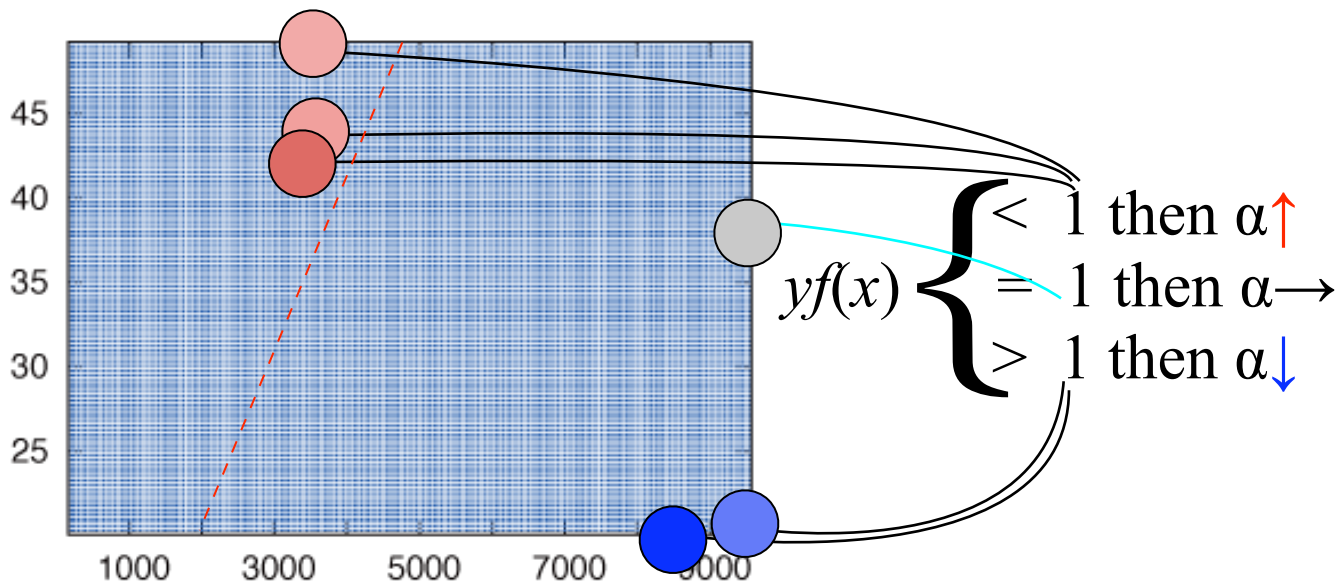
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(a)初期状態

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

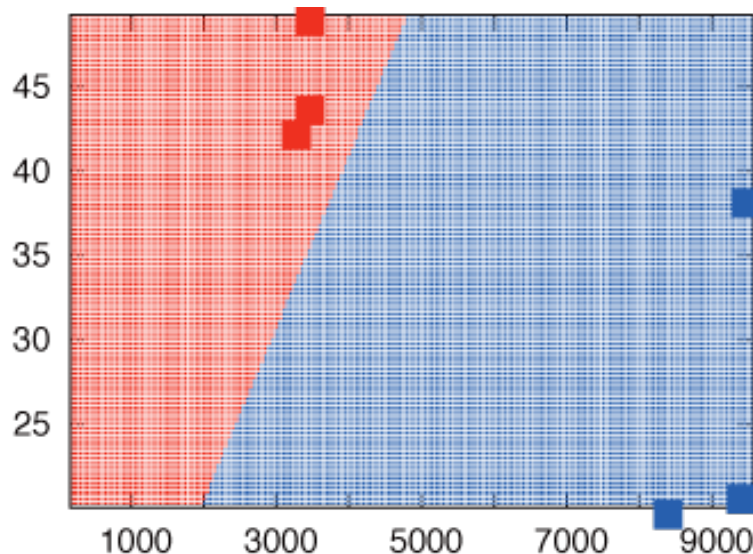
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

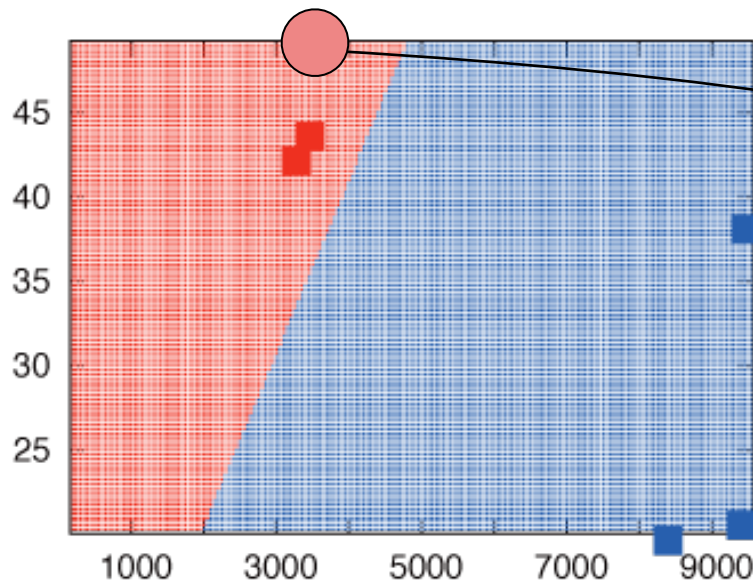
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$$

(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

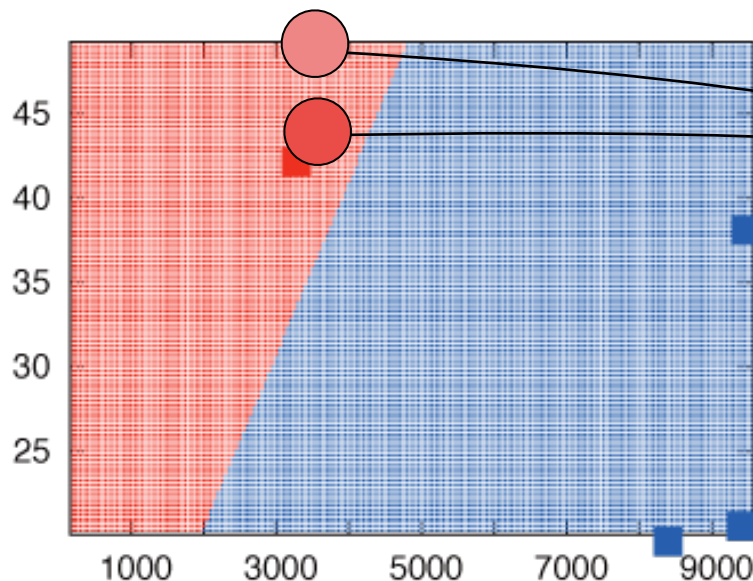
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

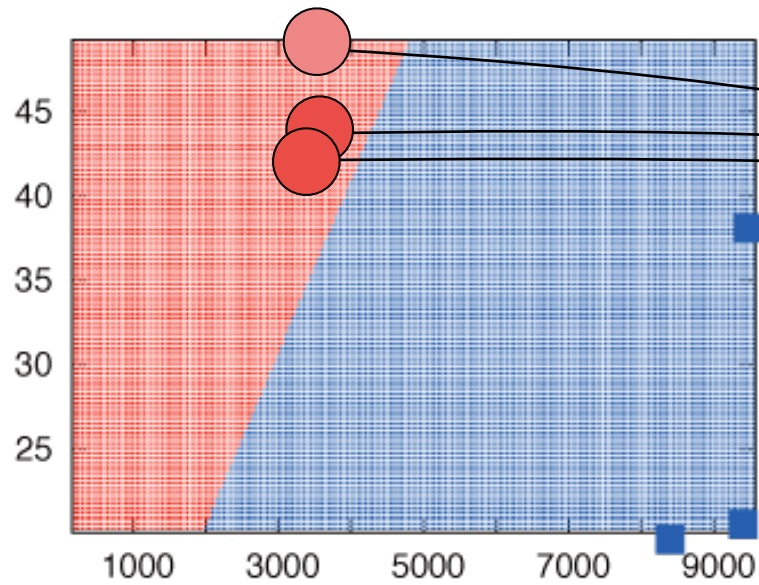
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

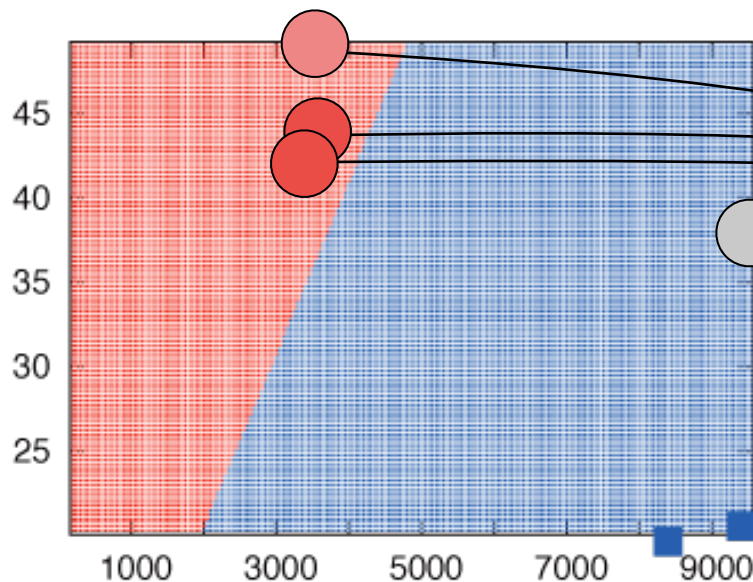
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

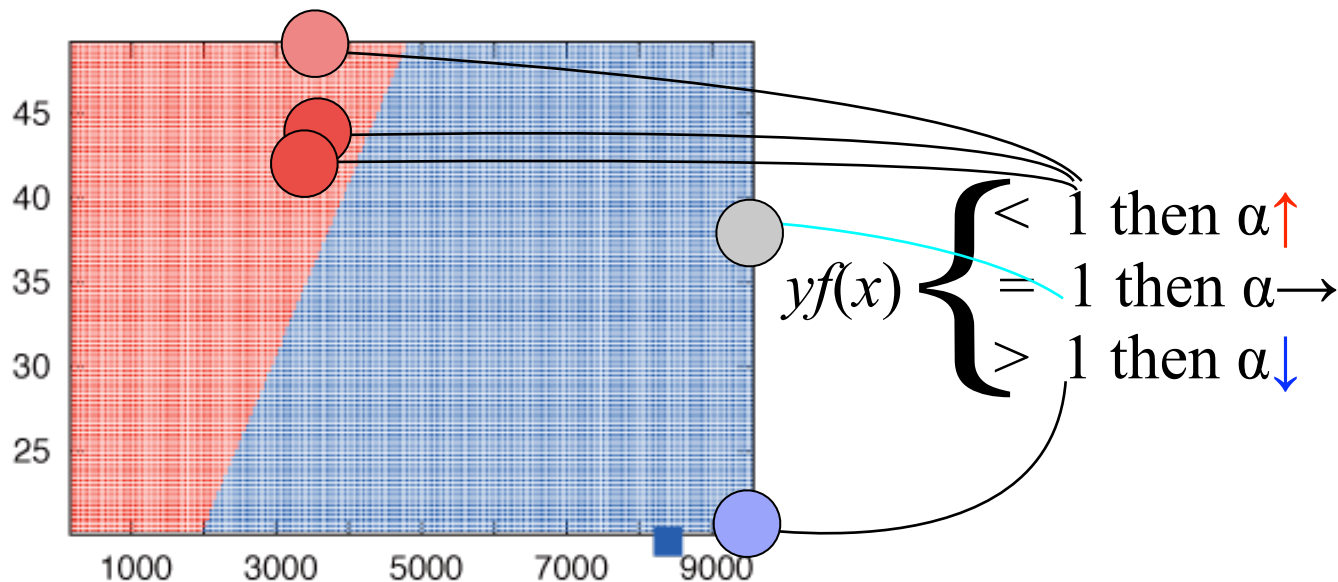
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

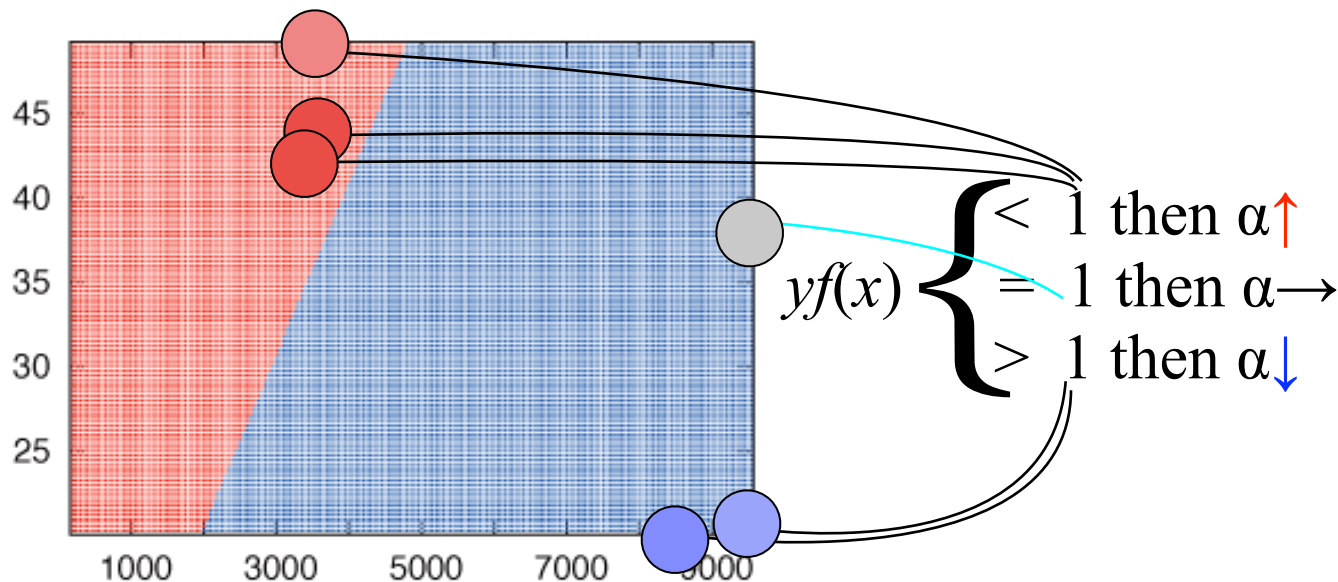
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

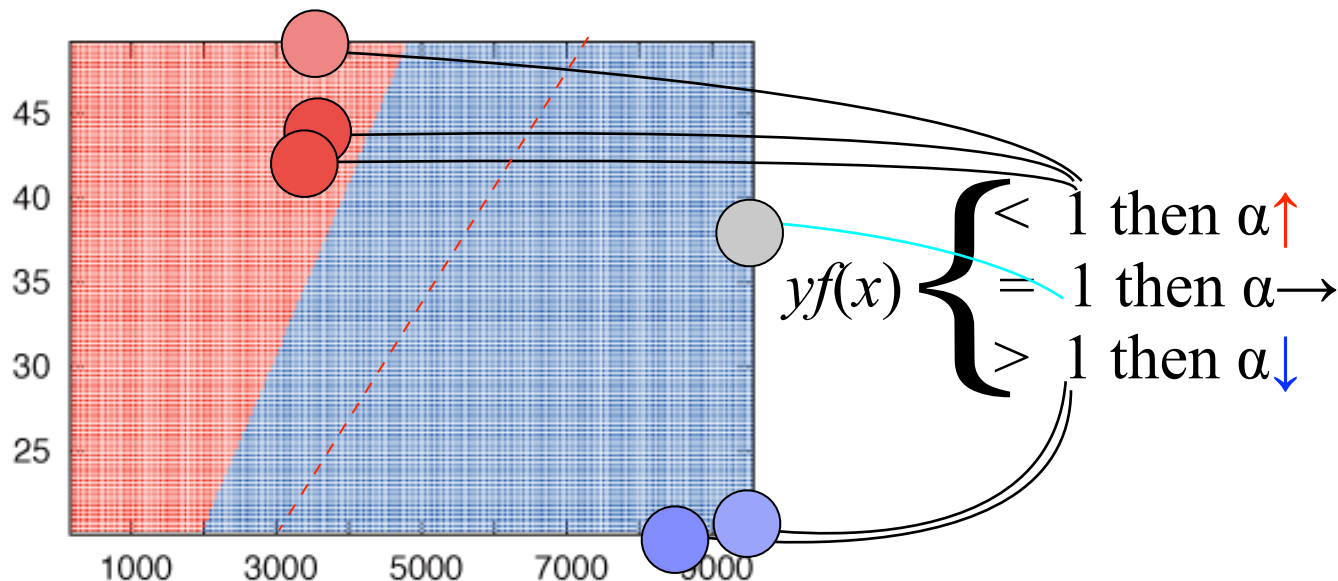
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(b)イテレーション4回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

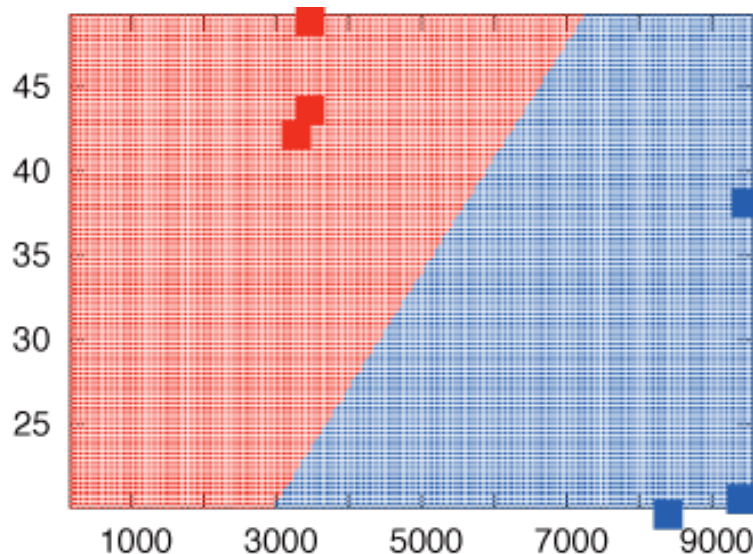
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

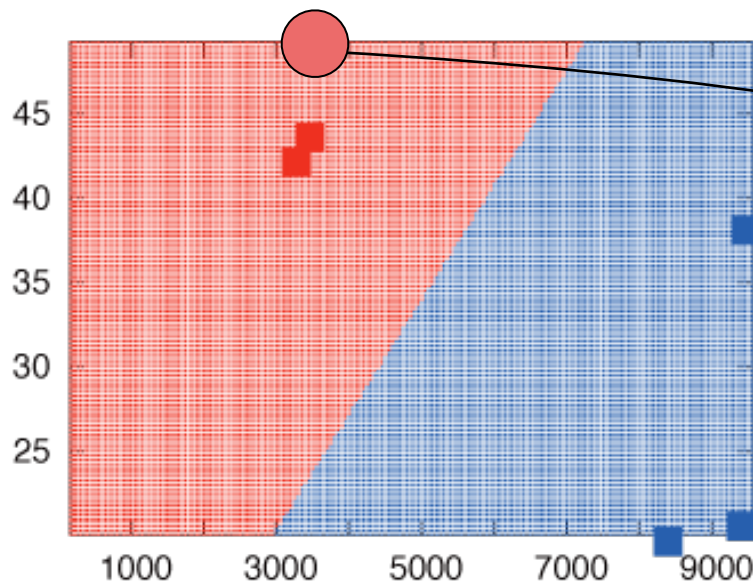
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

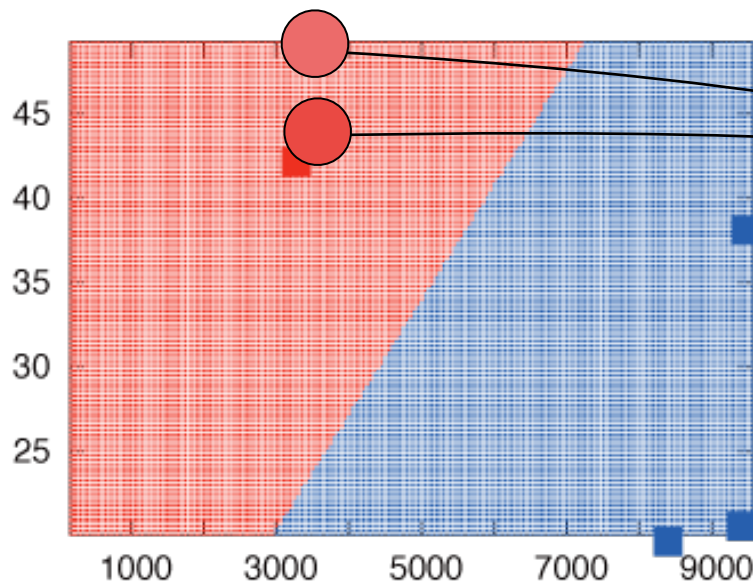
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

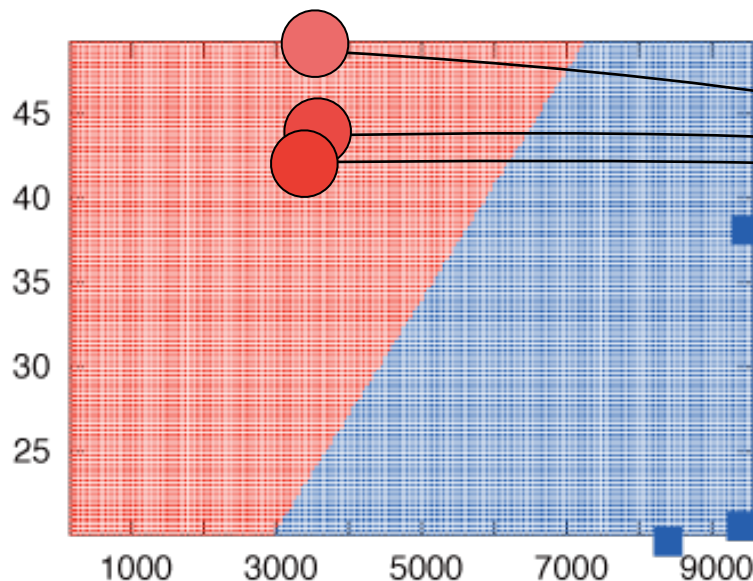
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$$

(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

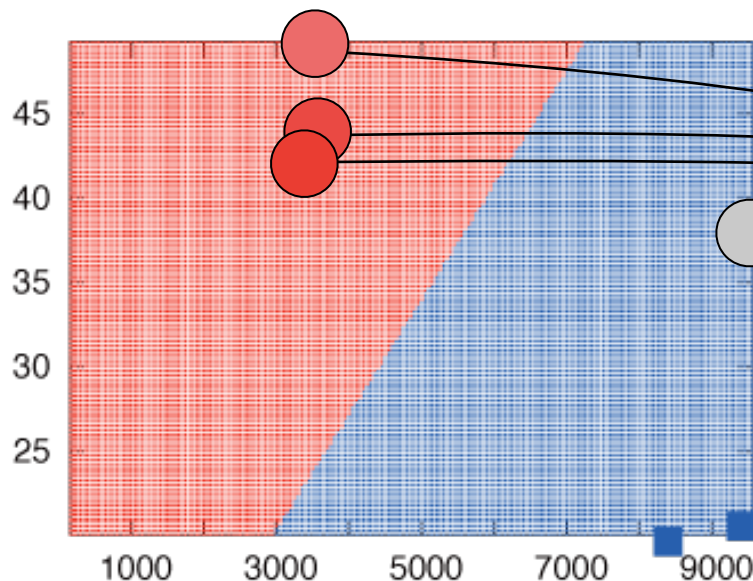
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$$

(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

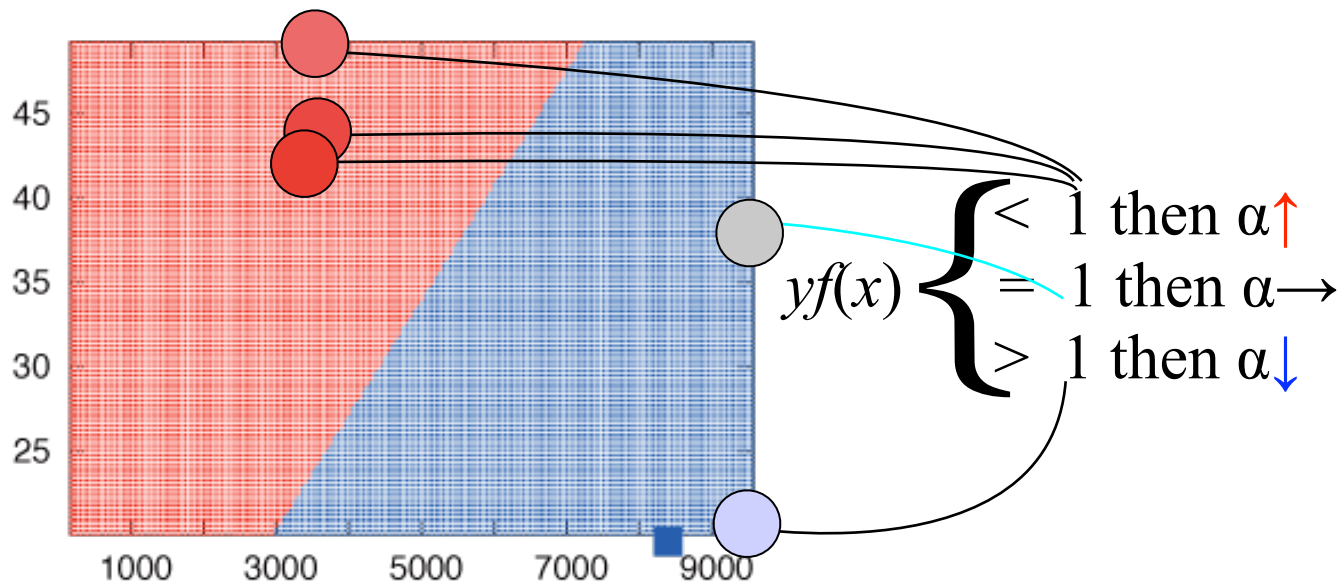
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

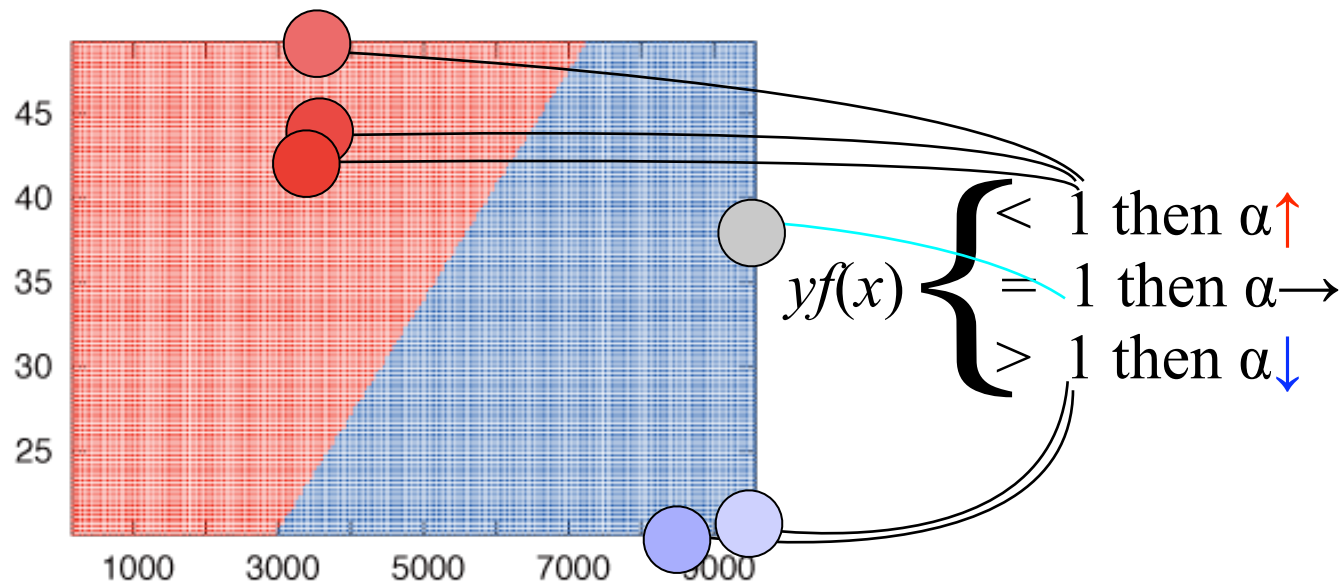
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

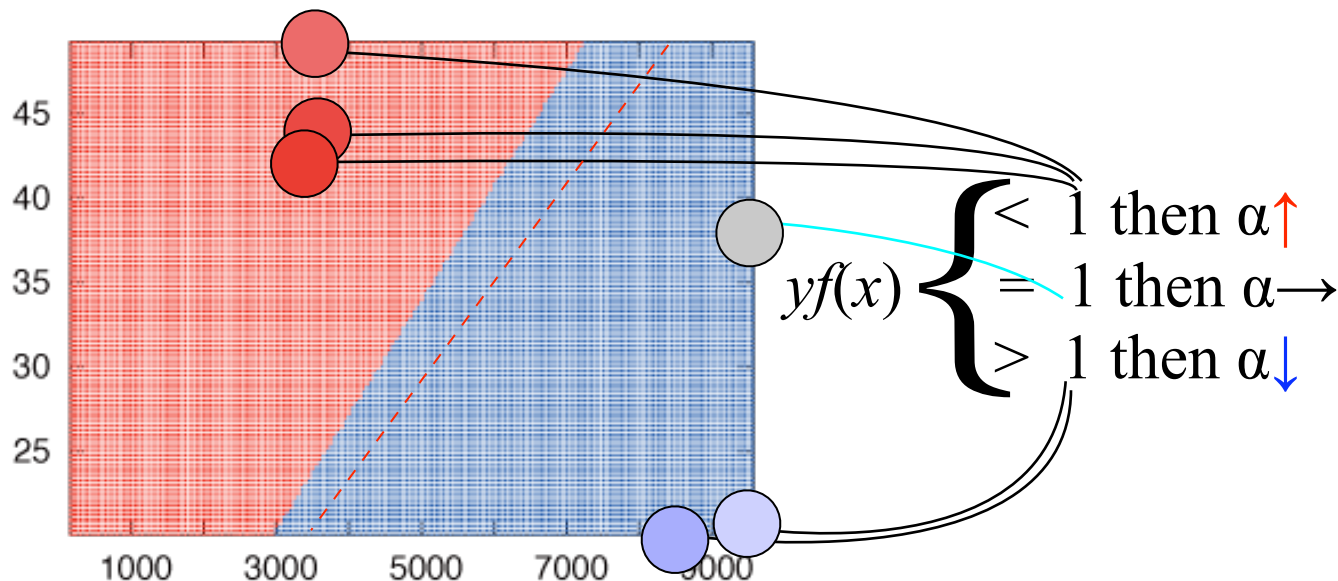
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(c)イテレーション6回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

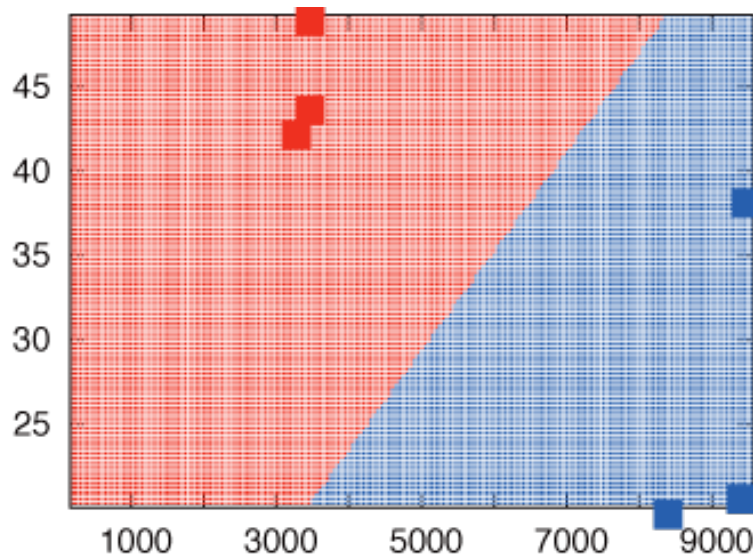
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$

(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

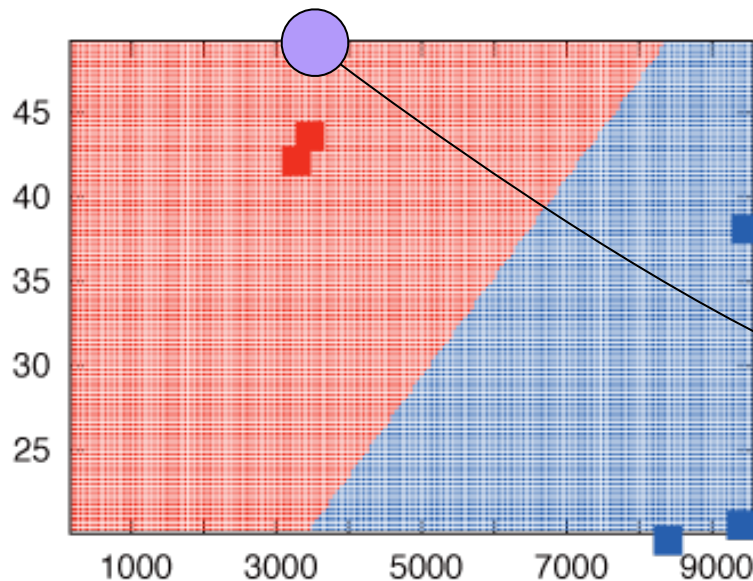
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$

(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

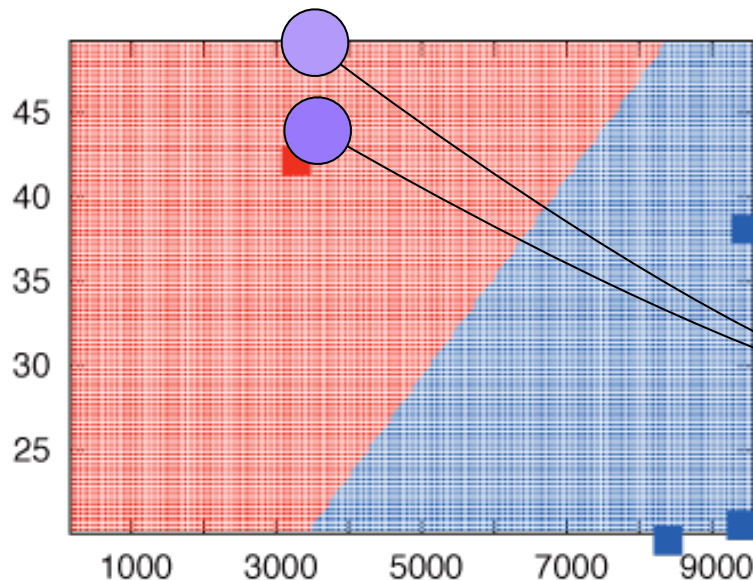
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 \text{ then } \alpha \uparrow \\ = 1 \text{ then } \alpha \rightarrow \\ > 1 \text{ then } \alpha \downarrow \end{cases}$

(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

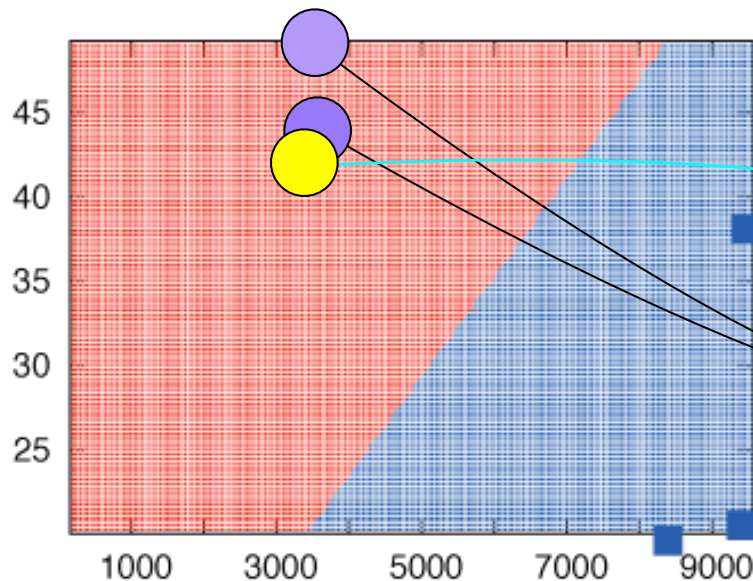
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$

(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

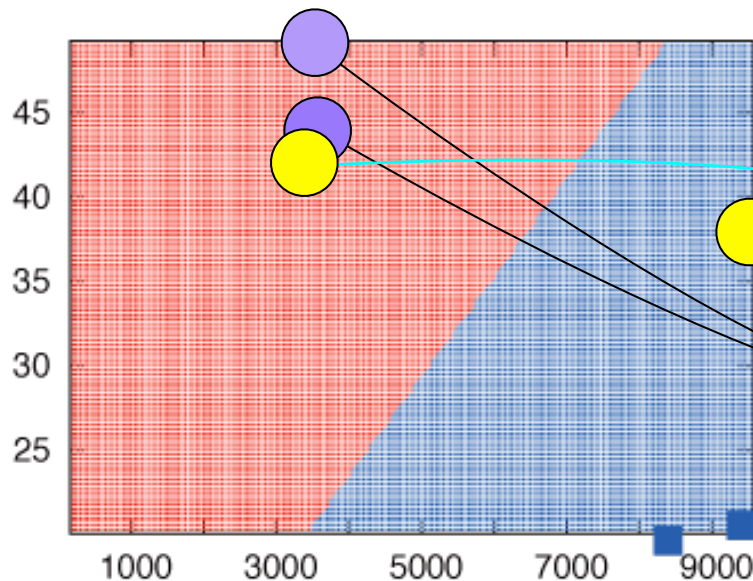
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



$$yf(x) \begin{cases} < 1 & \text{then } \alpha \uparrow \\ = 1 & \text{then } \alpha \rightarrow \\ > 1 & \text{then } \alpha \downarrow \end{cases}$$

(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

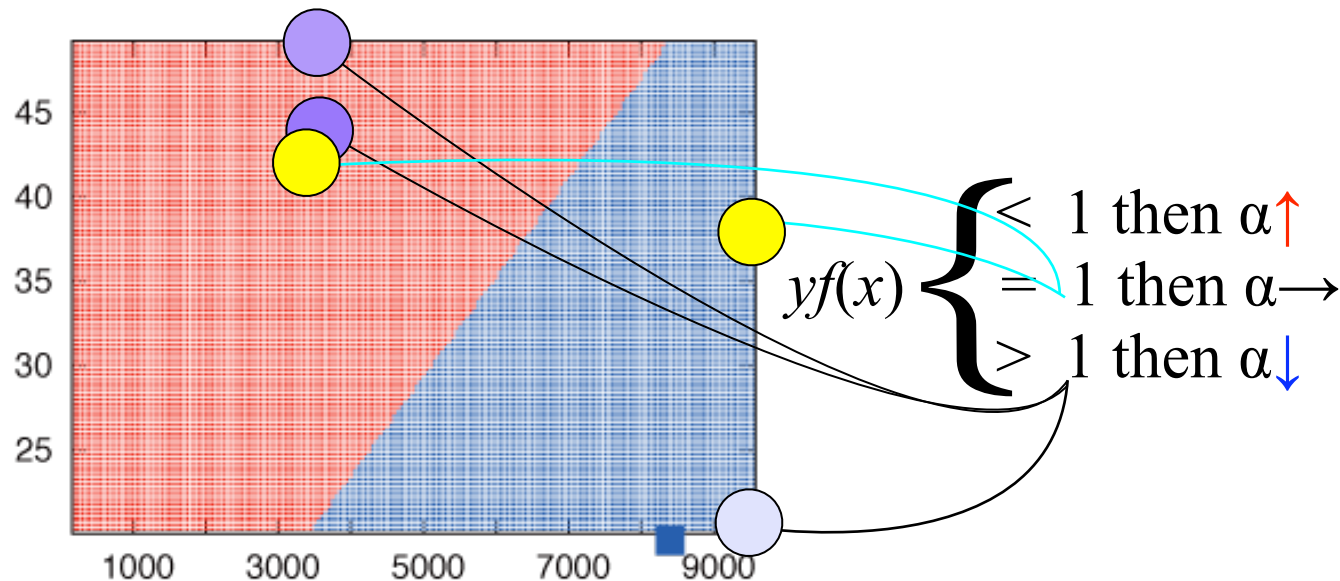
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(d)イテレーション8回目

再急降下法による最大マージンSVM: SVM

- VSAM
- Object Detection
- Segmentation
- Object Tracking
- Mapping
- Feature Extraction
- Object Classification

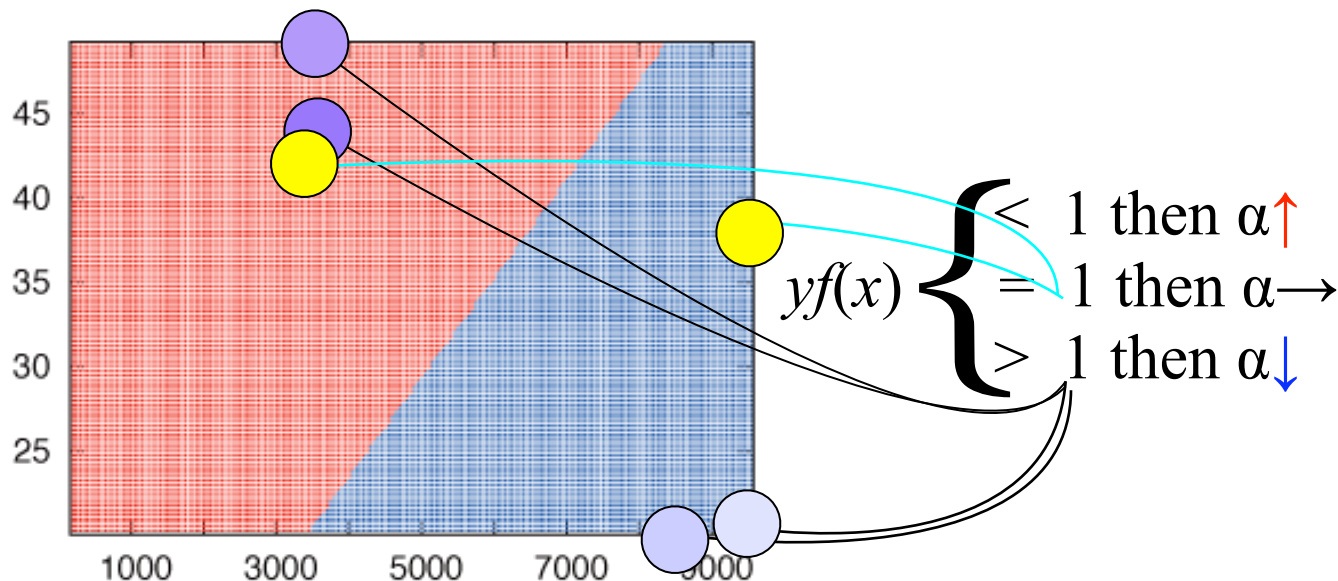
- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

- 重みベクトル

$$\mathbf{w}^* = \sum_{i=1}^n y_i \alpha_i^* \mathbf{x}_i$$



(d)イテレーション8回目

ソフトマージンSVM: SVM

- ・ 最大マージンSVM
 - 線形判別可能な問題にのみ適用可能
 - 線形分離不可能な問題に対し境界算出不可能
- ・ マージンスラック変数 ξ を導入
 - 分類ミスを許し, 柔らかなマージンを評価
- ・ ソフトマージンSVMのラグランジアン

$$L(\mathbf{w}, b, \alpha, \mathbf{r}) = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle + C \sum_{i=1}^n \xi_i \sum_{i=1}^n -\alpha_i [y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^l r_i \xi_i$$

$$= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

$$\frac{\partial L}{\partial \xi} = C\xi - \alpha = \mathbf{0}$$

再急降下法によるソフトマージンSVM: SVM

- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_i)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

if $\alpha_i > C$ *then* $\alpha \leftarrow C$

VSAM

Object
Detection

Segmentation

Object
Tracking

Mapping

Feature
Extraction

Object
Classification

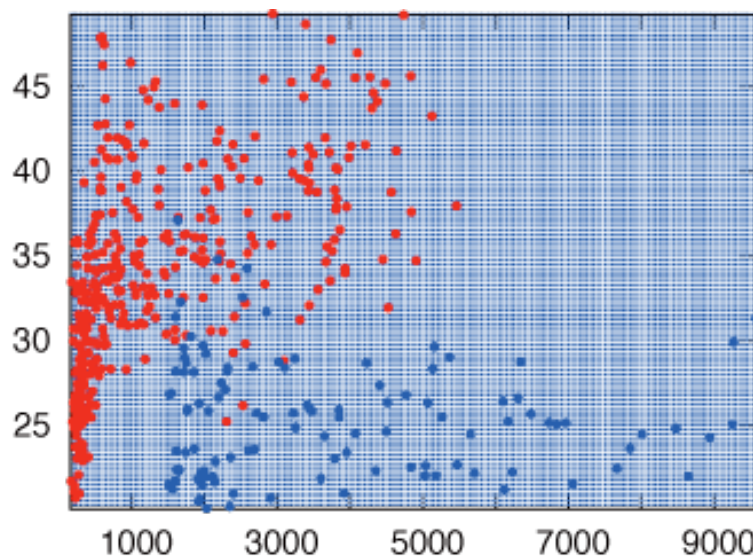
再急降下法によるソフトマージンSVM: SVM

- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \cdot \mathbf{x}_j)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

if $\alpha_i > C$ *then* $\alpha \leftarrow C$



(a)初期状態

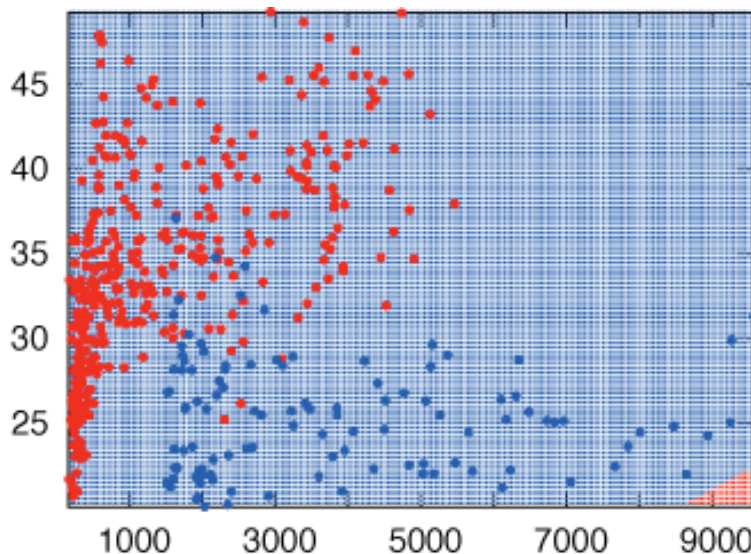
再急降下法によるソフトマージンSVM: SVM

- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_j)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

if $\alpha_i > C$ *then* $\alpha \leftarrow C$



(b)イテレーション10回目

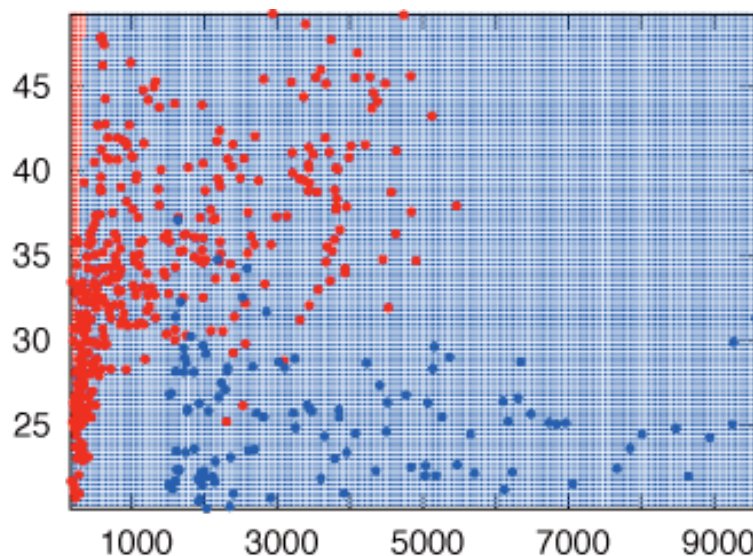
再急降下法によるソフトマージンSVM: SVM

- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \mathbf{x}_j)$$

if $\alpha_i < 0$ *then* $\alpha \leftarrow 0$

if $\alpha_i > C$ *then* $\alpha \leftarrow C$



(c)イテレーション30回目

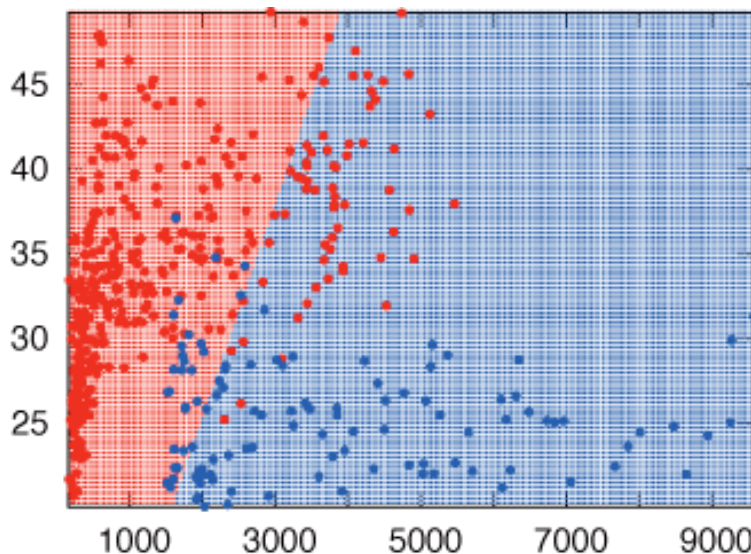
再急降下法によるソフトマージンSVM: SVM

- 適当な学習率 η による逐次更新

$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i \sum_{j=1}^m \alpha_j y_j \mathbf{x}_i^t \cdot \mathbf{x}_j)$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha_i \leftarrow 0$$

$$\text{if } \alpha_i > C \text{ then } \alpha_i \leftarrow C$$



(d)イテレーション70回目

カーネルSVM: SVM

VSAM

- カーネルトリックにより非線形境界を構築
 - 非線形写像空間を特徴空間として識別

$$f(\mathbf{x}) = \sum_{i=1}^n y_i \alpha_i \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \rangle + b$$
$$\langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \rangle = K(\mathbf{x}_i, \mathbf{x})$$

Object
Detection

Segmentation

Object
Tracking

- 代表的なカーネル

Mapping

- 線形カーネル $K(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^t \mathbf{x}_2)$

- 多項式カーネル $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \langle \mathbf{x}_1 \cdot \mathbf{x}_2 \rangle)^p$

Feature
Extraction

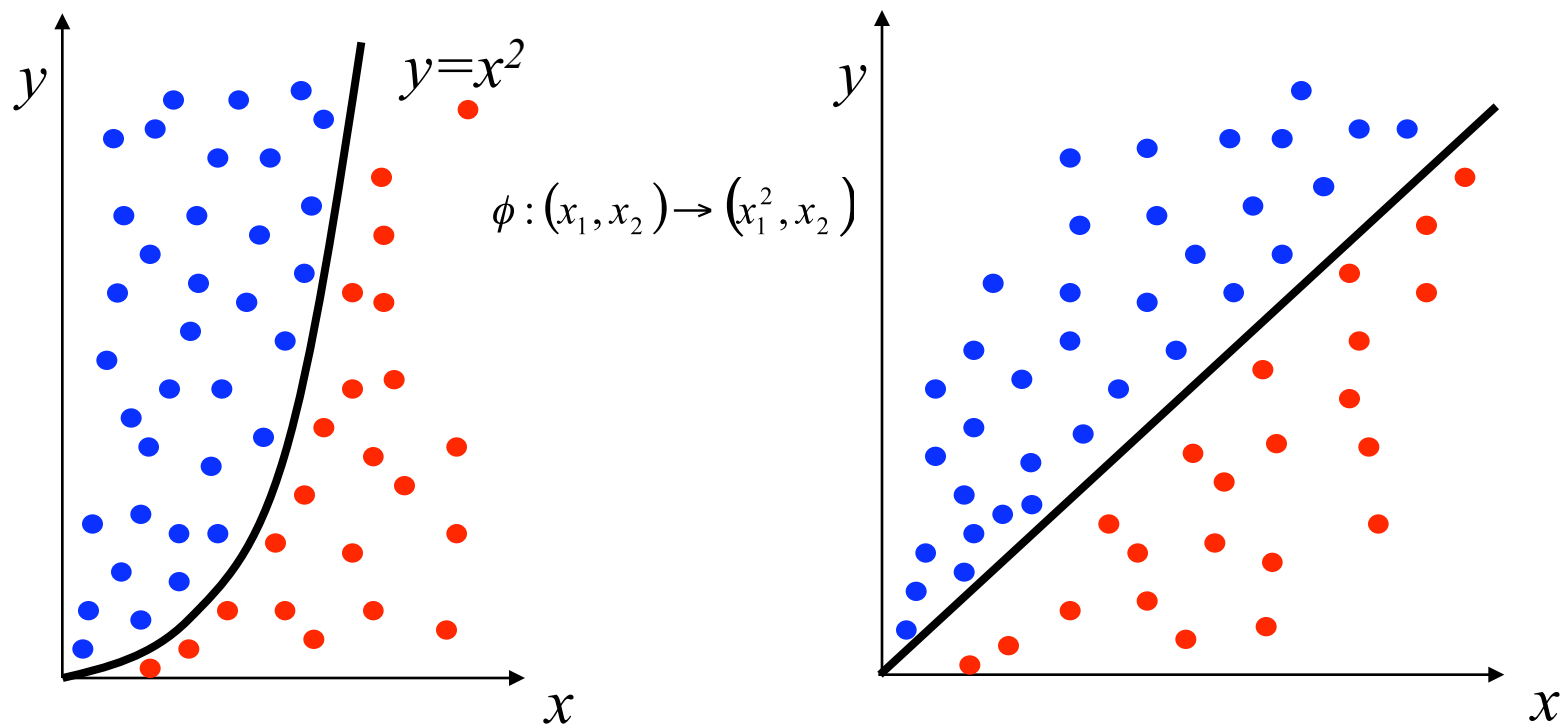
- ガウスカーネル $K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2}{2\sigma^2}\right)$

Object
Classification

- 非線形境界構築時に一般的
- σ の大小によりフィッティングの度合いを調整

カーネルトリック

- SVM：非線形判別不可能
 - サンプルを線形判別可能なカーネル特徴空間へ写像
 - 最悪でもサンプル数と同等のカーネル特徴空間なら線形判別可能



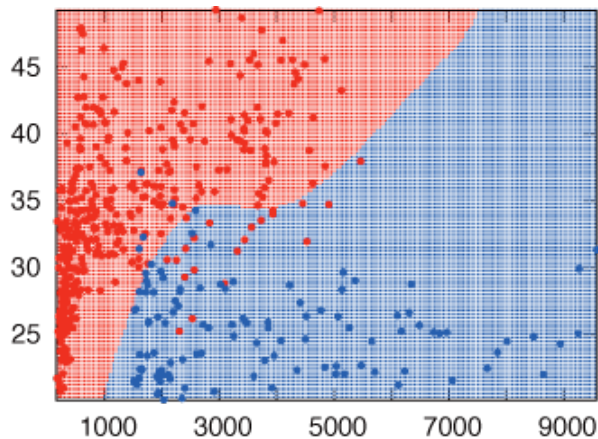
再急降下法によるカーネルSVM: SVM

- 適当な学習率 η による逐次更新

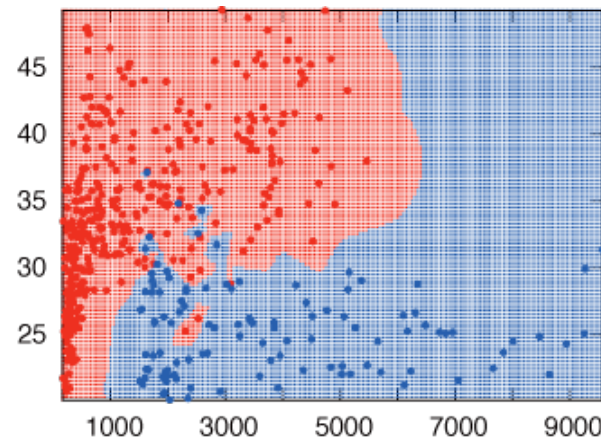
$$\alpha_i \leftarrow \alpha_i + \eta_i (1 - y_i) \sum_{j=1}^m \alpha_j y_j \underline{K(\mathbf{x}_i, \mathbf{x}_j)}$$

$$\text{if } \alpha_i < 0 \text{ then } \alpha \leftarrow 0$$

$$\underline{\text{if } \alpha_i > C \text{ then } \alpha \leftarrow C}$$



(a) σ^2 :大



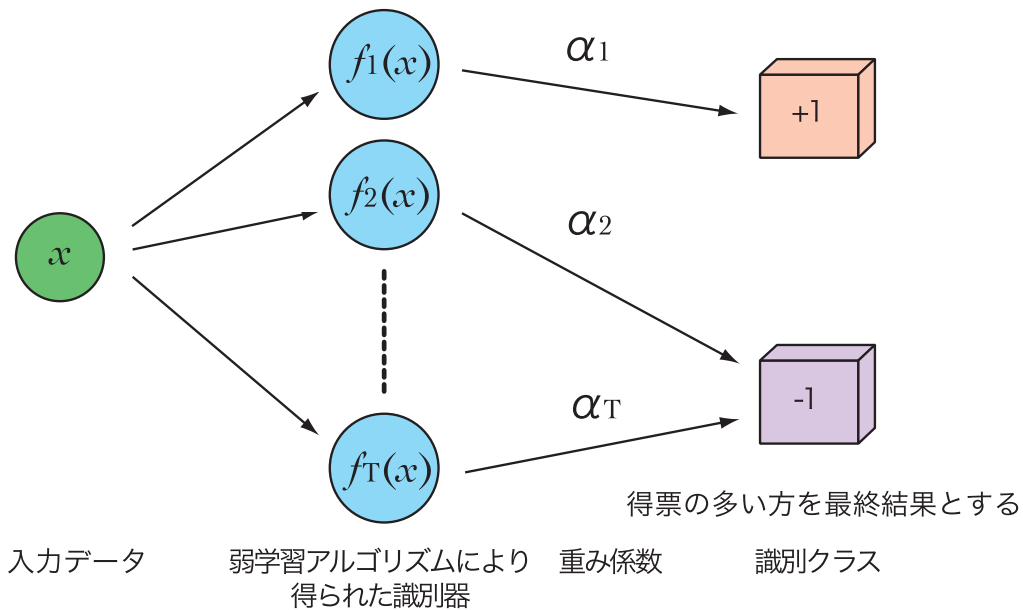
(b) σ^2 :小

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2}{2\sigma^2}\right)$$

Adaboostによる識別: Adaboost

- 低精度の識別関数の集合から高精度な識別関数を導く手法

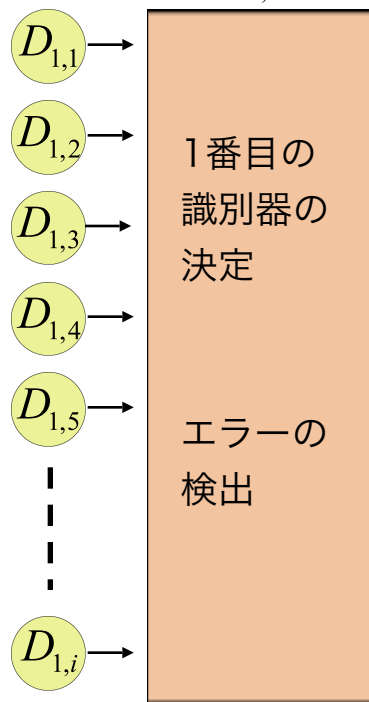
$$F(x) = \sum_t \alpha_t f_t(x) \quad f_t(x) = \begin{cases} +1 \\ -1 \end{cases}$$



Adaboostの学習:

- ・ 誤ったデータに対し比重が大きくなる
→次の識別器では比重の大きいデータに着目する

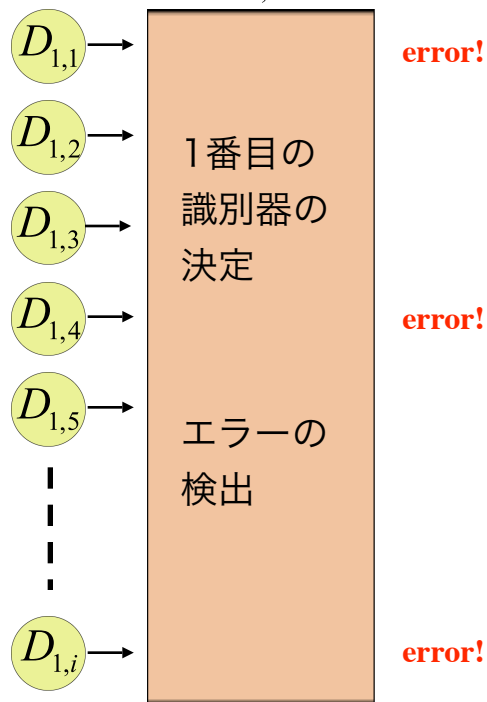
データの重み $D_{t,i}$



Adaboostの学習:

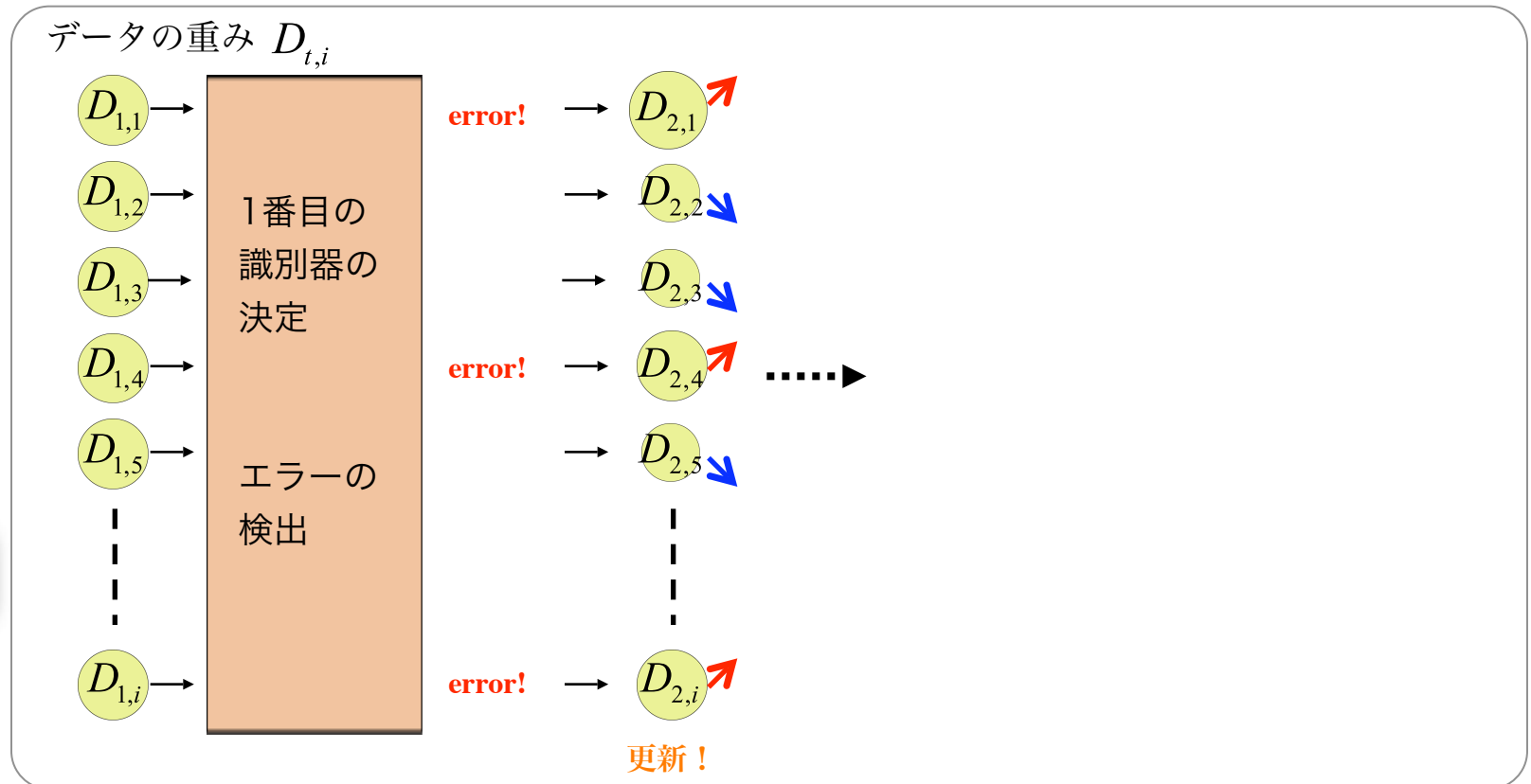
- ・ 誤ったデータに対し比重が大きくなる
→次の識別器では比重の大きいデータに着目する

データの重み $D_{t,i}$



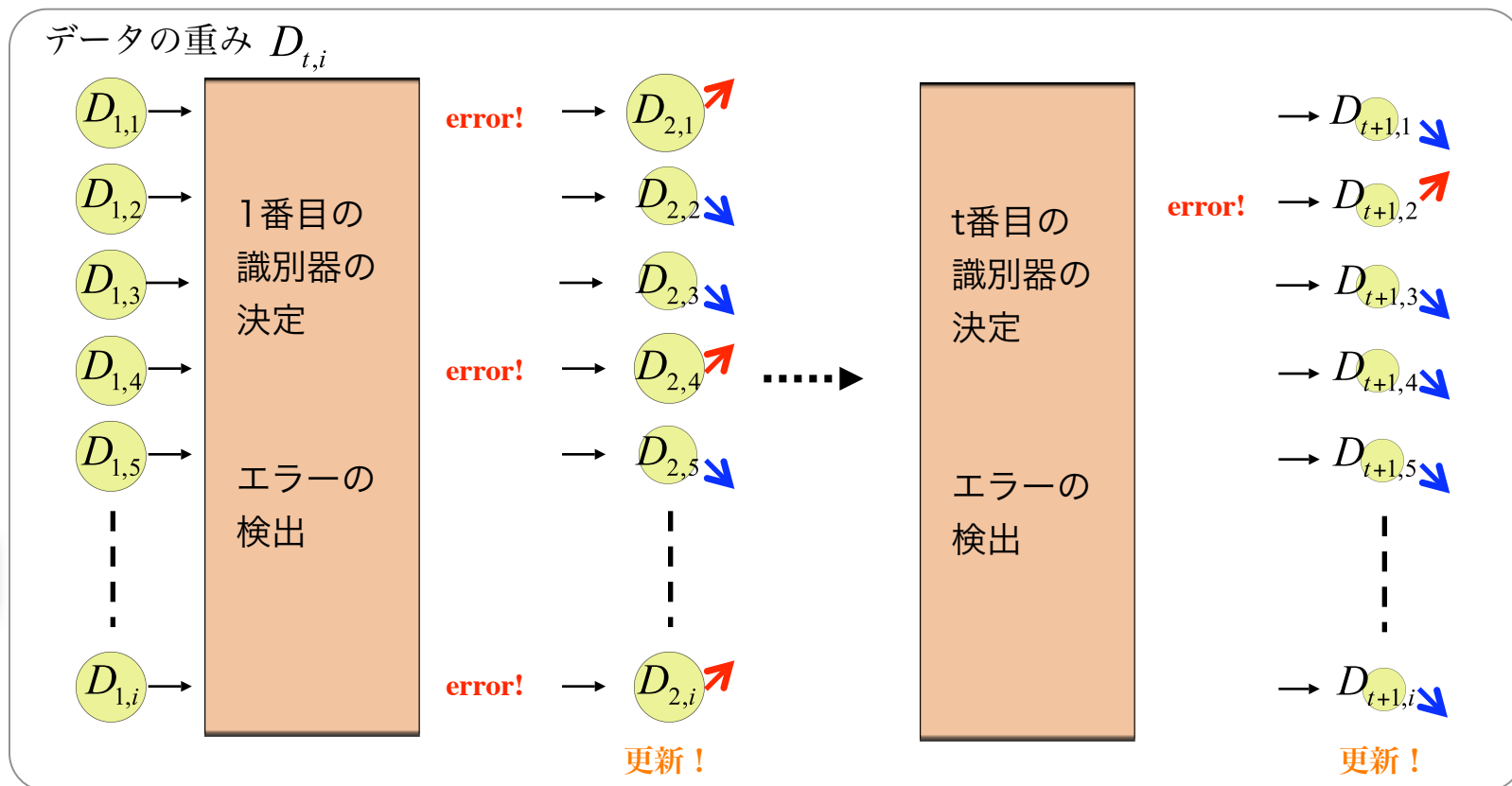
Adaboostの学習:

- ・ 誤ったデータに対し比重が大きくなる
→ 次の識別器では比重の大きいデータに着目する



Adaboostの学習:

- ・ 誤ったデータに対し比重が大きくなる
→ 次の識別器では比重の大きいデータに着目する



Adaboostの学習アルゴリズム:

AdaBoost

データ数: m , 識別関数: $f(x)$

$D_t(i)$: データ i の重み (出現確率) ($\sum_{i=1}^m D_t(i) = 1, D_t(i) \geq 0$)

初期値 $D_t(i) \leftarrow \frac{1}{m}, \epsilon_t = 0.0$

for $t \leftarrow 1$ to T

– D_t を用いて f_t を学習

– 重み付き誤差 ϵ_t を計算 ($\epsilon_t < 0.5$ が成り立つとする)

$\epsilon_t \leftarrow 0$

for $i \leftarrow 1$ to m

if $f_t(x_i) \neq y_i$

$\epsilon_t \leftarrow \epsilon_t + D_t(i)$

$-\alpha_t \leftarrow \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$

$-D_{t+1}(i) \leftarrow \begin{cases} D_t(i)e^{-\alpha_t}/Z_t & \text{if } f_t(x_i) = y_i \\ D_t(i)e^{+\alpha_t}/Z_t & \text{if } f_t(x_i) \neq y_i \end{cases}$

$Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i f_t(x_i))$

Adaboostにおける識別境界:

VSAM

- ・ 逐次的に識別境界が変化

Object
Detection

Segmentation

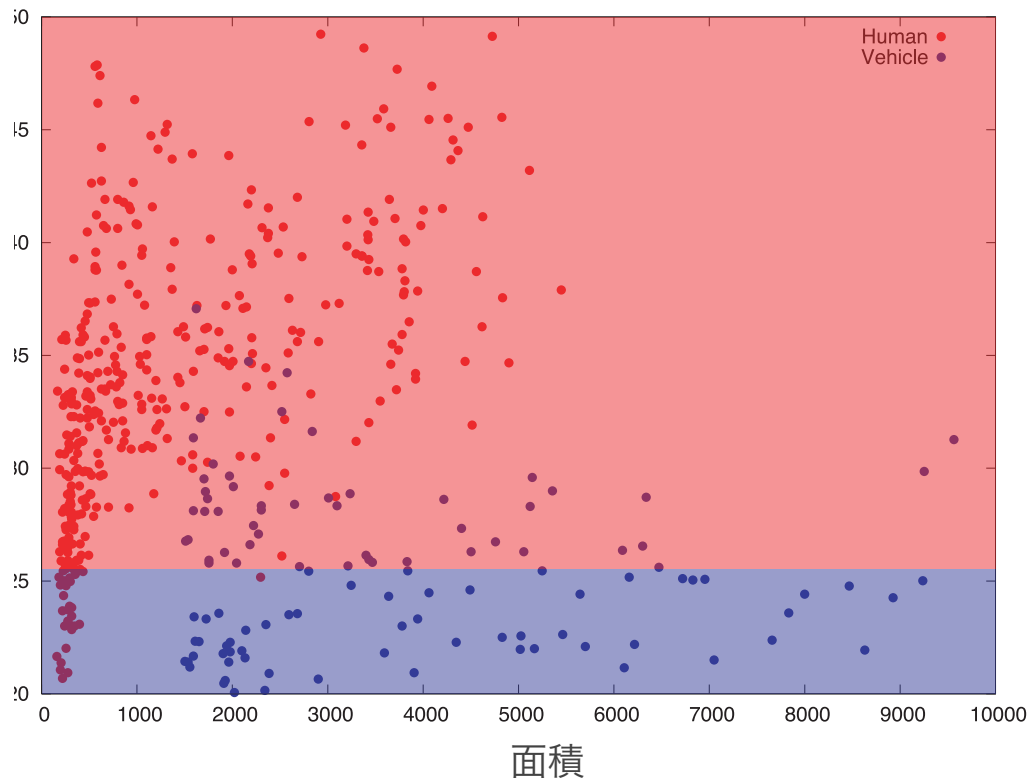
Object
Tracking

Mapping

Feature
Extraction

Object
Classification

複雑度



学習回数 : 1回 識別率 : 84.3%

Adaboostにおける識別境界:

VSAM

- 逐次的に識別境界が変化

Object Detection

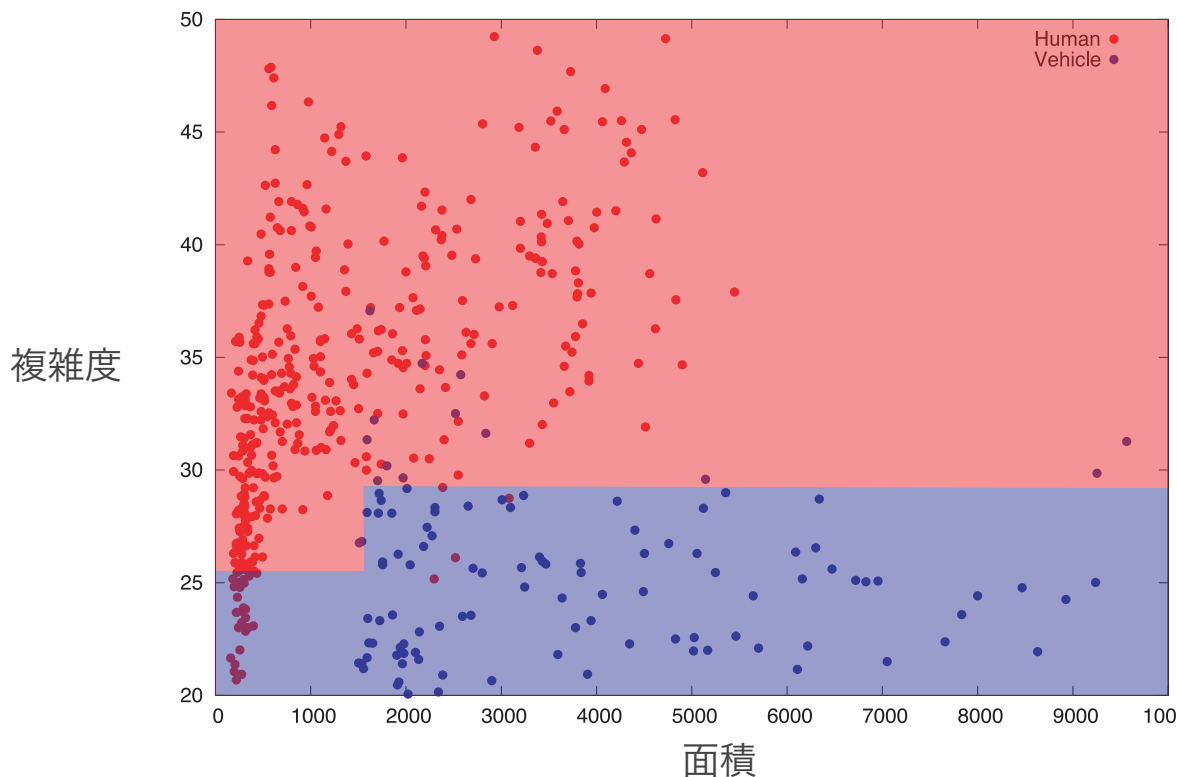
Segmentation

Object Tracking

Mapping

Feature Extraction

Object Classification



学習回数 : 3回 識別率 : 92.6%

Adaboostにおける識別境界:

VSAM

- 逐次的に識別境界が変化

Object Detection

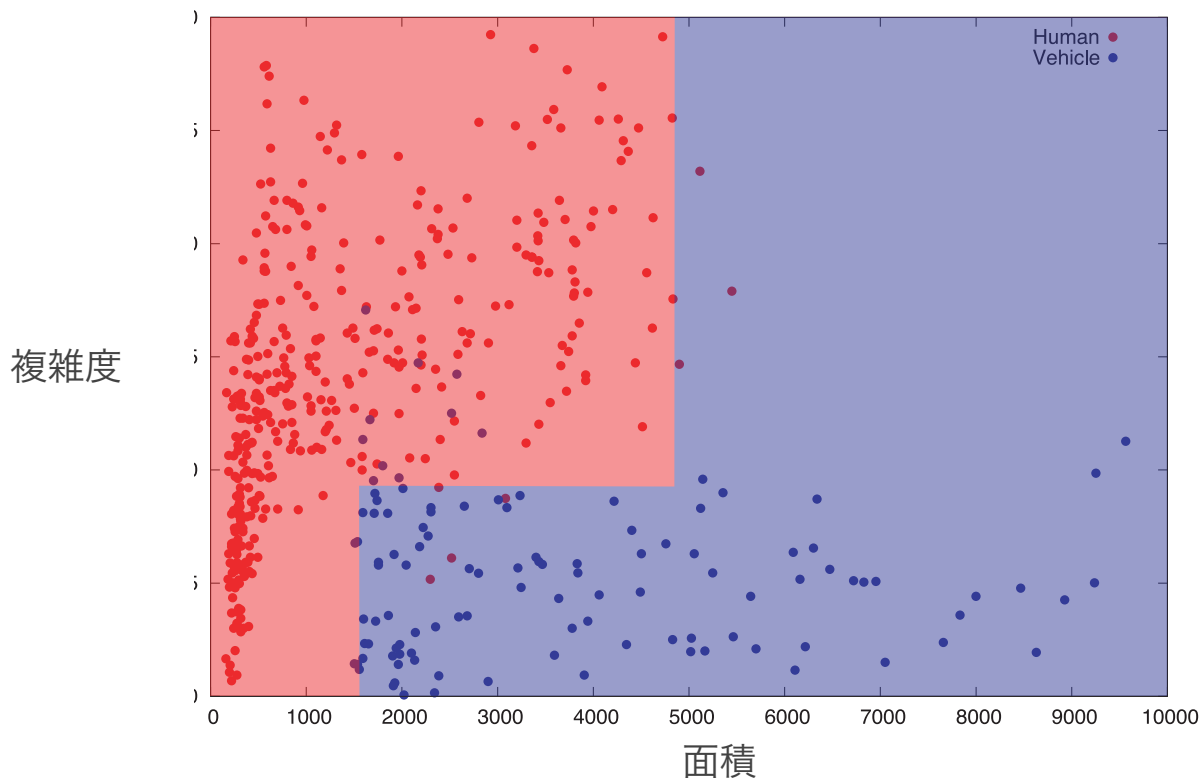
Segmentation

Object Tracking

Mapping

Feature Extraction

Object Classification



学習回数：4回 識別率：97.1%

Adaboostにおける識別境界:

VSAM

- ・ 逐次的に識別境界が変化

Object
Detection

Segmentation

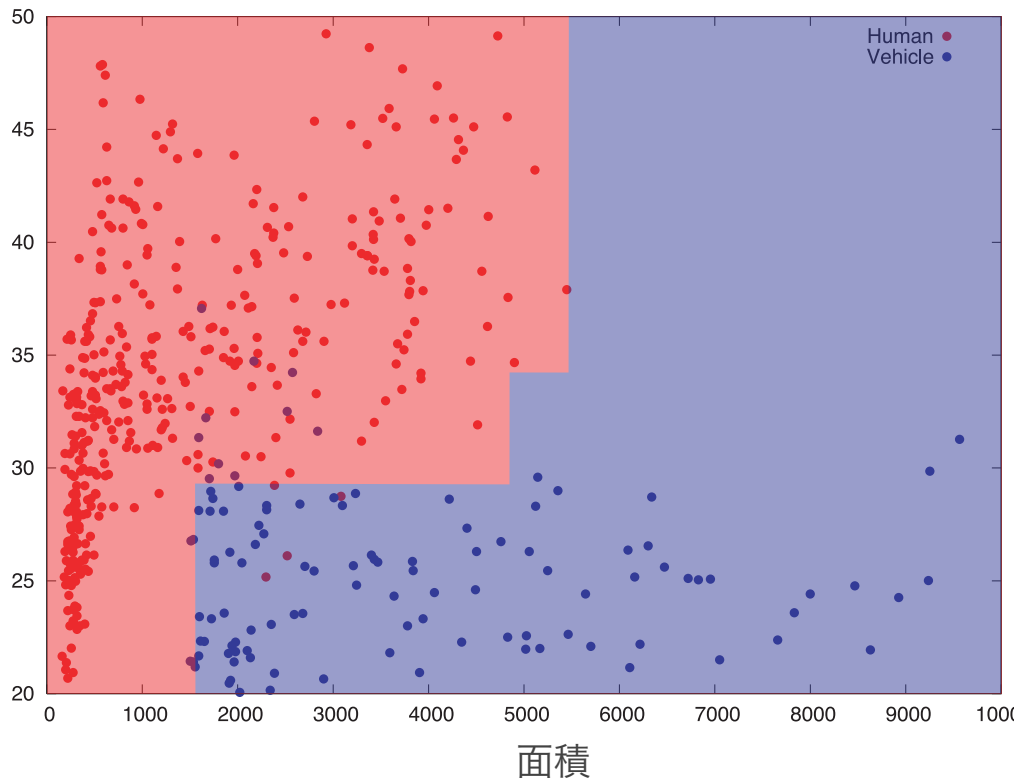
Object
Tracking

Mapping

Feature
Extraction

Object
Classification

複雑度



学習回数 : 22回

識別率 : 97.5%

Adaboostにおける識別境界:

VSAM

- 逐次的に識別境界が変化

Object
Detection

Segmentation

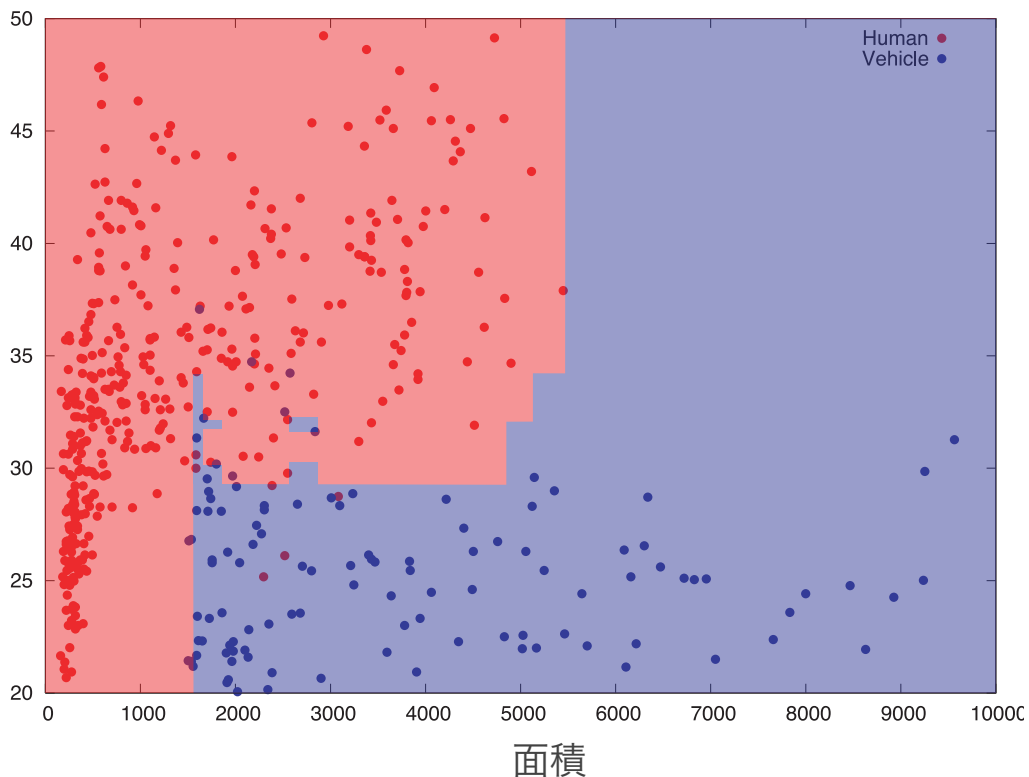
Object
Tracking

Mapping

Feature
Extraction

Object
Classification

複雑度



学習回数 : 99回 識別率 : 98.3%

Adaboostにおける識別境界:

VSAM

- ・ 逐次的に識別境界が変化

Object
Detection

Segmentation

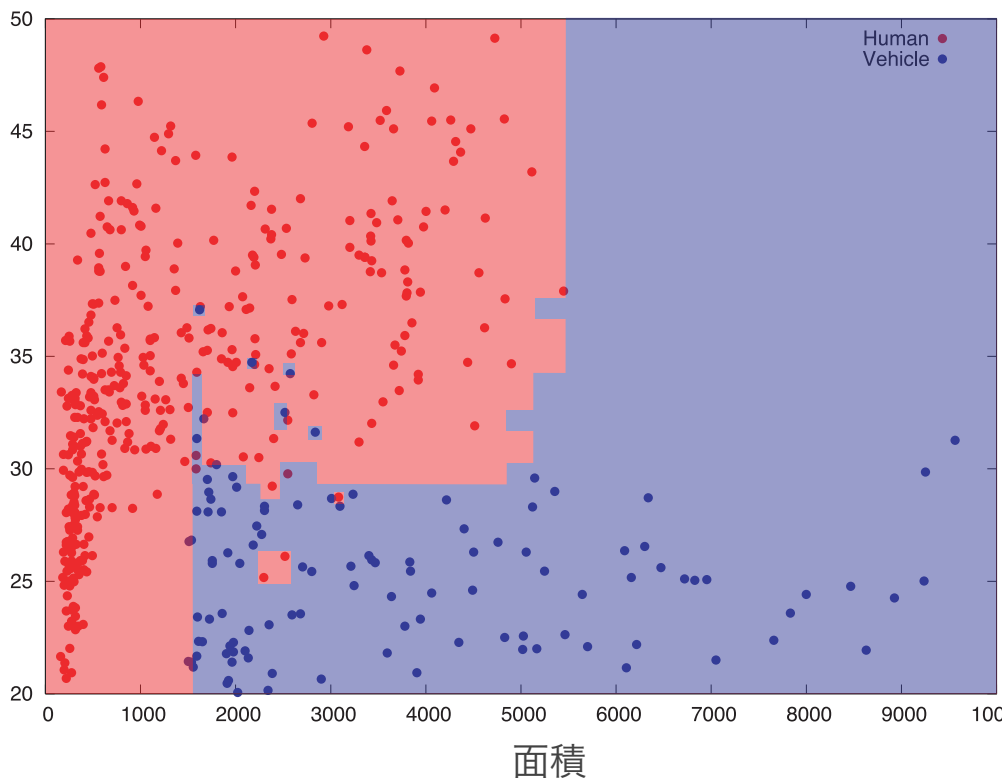
Object
Tracking

Mapping

Feature
Extraction

Object
Classification

複雑度



学習回数 : 255回 識別率 : 100%

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(\mathbf{x}) = \sum_{t=1}^T h_t(\mathbf{x}) \quad h(\mathbf{x}) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(\mathbf{x}) = \sum_{t=1}^T h_t(\mathbf{x}) \quad h(\mathbf{x}) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

Desecrate AdaBoost
$$F(\mathbf{x}) = \sum_t \alpha_t f_t(\mathbf{x})$$

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(\mathbf{x}) = \sum_{t=1}^T h_t(\mathbf{x}) \quad h(\mathbf{x}) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

Desecrate AdaBoost $F(\mathbf{x}) = \sum_t \alpha_t f_t(\mathbf{x})$

Real Adaboost: Adaboost

- ・ 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(\mathbf{x}) = \sum_{t=1}^T h_t(\mathbf{x}) \quad h(\mathbf{x}) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

Desecrate AdaBoost $F(\mathbf{x}) = \sum_t \alpha_t f_t(\mathbf{x})$

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(x) = \sum_{t=1}^T h_t(x) \quad h(x) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

$$\frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

Desecrate AdaBoost $F(x) = \sum_t \alpha_t f_t(x)$

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(x) = \sum_{t=1}^T h_t(x) \quad h(x) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

$$\frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \quad \begin{cases} +1 \\ -1 \end{cases}$$

Desecrate AdaBoost $F(x) = \sum_t \alpha_t f_t(x)$

Real Adaboost: Adaboost

- 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(x) = \sum_{t=1}^T h_t(x)$$

$$h(x) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon}$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

$$\frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \begin{cases} +1 \\ -1 \end{cases}$$

Desecrate AdaBoost $F(x) = \sum_t \alpha_t f_t(x)$

Real Adaboost: Adaboost

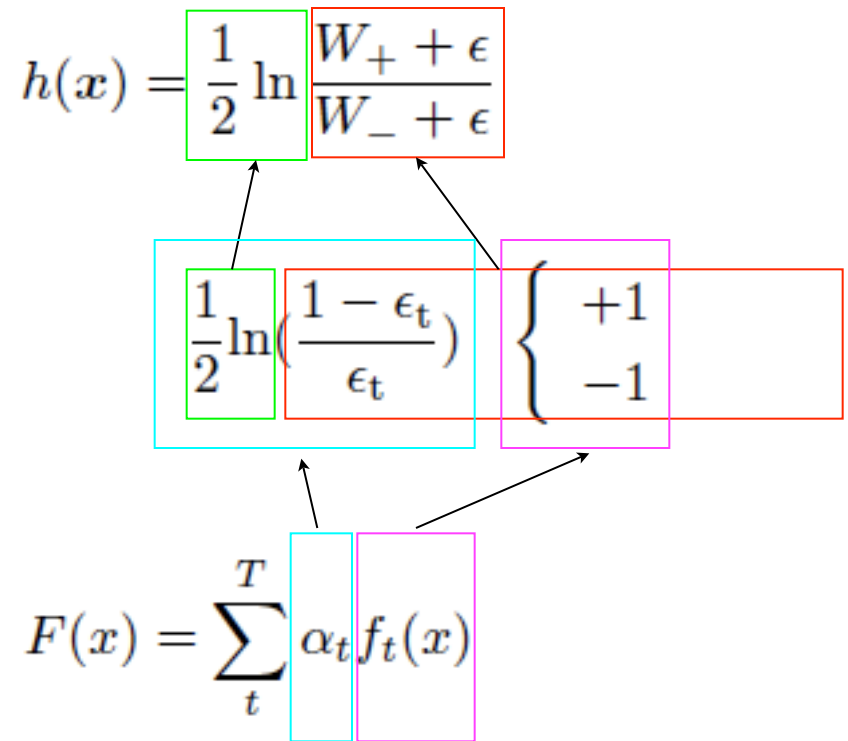
- ・ 実数による信頼度を弱識別器に
 - どの程度対象クラスらしいか
 - 弱識別器の表現力が向上

$$H(x) = \sum_{t=1}^T h_t(x)$$

- 確率密度

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$



Desecrate AdaBoost

$$F(x) = \sum_t \alpha_t f_t(x)$$

Real Adaboostの学習アルゴリズム:

Real AdaBoost

M 個の学習サンプル: $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$, ラベル: $y = \{+1, -1\}$

$D_t(i)$: データ i の重み ($\sum_{i=1}^m D_t(i) = 1, D_t(i) \geq 0$)

初期値 $D_t(i) \leftarrow \frac{1}{m}$

for $t \leftarrow 1$ to T

- 確率密度関数 W の作成

$$W_+^j = \sum_{i: j \in J \wedge y_i = +1} D_t(i)$$

$$W_-^j = \sum_{i: j \in J \wedge y_i = -1} D_t(i)$$

- 評価値 Z の算出

$$Z = 2 \sum_j \sqrt{W_+^j W_-^j}$$

- 弱識別器の選択

$$h_t(\mathbf{x}) = \operatorname{argmax}_{h \in H} Z$$

- 学習サンプルの重みの更新

$$D_{t+1}(i) = \frac{D_t(i) e^{-h_t(\mathbf{x}) y_i}}{\sum_{m=1}^M D_t(m) e^{-h_t(\mathbf{x}) y_m}}$$

end for

VSAM

Object
Detection

Segmentation

Object
Tracking

Mapping

Feature
Extraction

Object
Classification

SVMとAdaBoost: SVM & AdaBoost

VSAM

Object
Detection

Segmentation

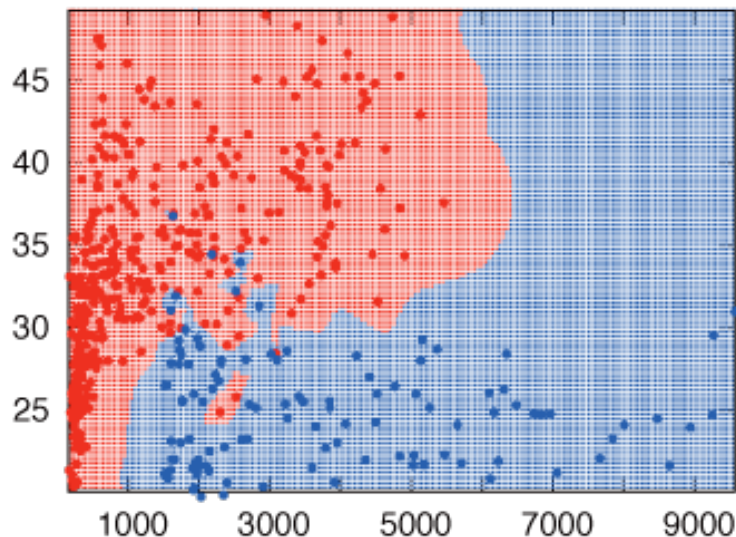
Object
Tracking

Mapping

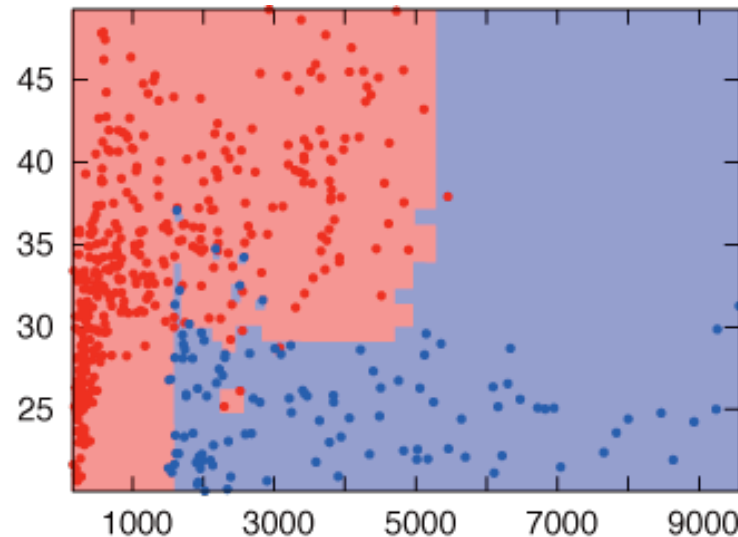
Feature
Extraction

Object
Classification

- ・ 共にマージン最大化による非線形識別が可能
 - SVM：直接マージン最大化
カーネル, カーネルパラメータ
 - AdaBoost：マージン関数最大化
弱識別器の素性



(a)SVM



(b)AdaBoost

物体識別のまとめ:

VSAM

- ・ より高性能な識別を実現するには

Object
Detection

- パターンの観察

Segmentation

- 有効な入力特徴量の選別

Object
Tracking

- バリエーションが多く、かつ大量の学習サンプルの採取

Mapping

- 識別器の選択

Feature
Extraction

- ・ 多次元空間における非線形な識別面

Object
Classification

- 学習時の工夫

- ・ bootstrap (誤検出パターンの学習)