

ECOCに基づく多値判別法とその拡張

竹之内 高志

奈良先端科学技術大学院大学

石井 信

京都大学

概要

- 多クラス問題の2値判別問題への分解
- 複数の2値判別器の情報統合による多クラスラベル予測(復号)法
 - ①確率モデルによる復号法
 - ②確率モデルによる復号法の局所化
 - ③Bradley-Terryモデルに基づく復号法の改良

多値判別の設定

- $\mathbf{x}$:入力, $y \in \{1, L, G\}$: ラベル
 - データセット: $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$
- $F(\mathbf{x}, y)$: 判別関数
 - ラベル予測 $\Rightarrow \arg \max_y F(\mathbf{x}, y)$
 - どうやって $F(\mathbf{x}, y)$ を構成するか？
- 多値判別問題を複数の2値判別問題に分割
 - 各々構成した2値判別器の出力を統合

多値判別問題の分解の枠組み

- $F(x,y)$ を2値判別関数の組み合わせで構成する
 - ErrorCorrectingOutputCoding(ECOC)に基づく手法
(Loss-based Decoding by Dietterich&Bakiri, Allwein et al.)
 - 単純なHamming decodingを含む
 - 実装が簡単
 - クラス所属確率に関しては言及しない
 - Bradley-Terryモデルに基づく手法
(Hastie&Tibshirani, Yukinawa et al.)
 - クラス所属確率 ($p(y|x)$) を推定できる
 - 確率出力を持つ2値判別器が必要
 - 例題毎に事後確率を計算する必要あり

多クラス問題の2値判別問題への分解

-符号表

- 符号表 $W: p \times G$ 行列
 - $W_{jk} \in \{1, -1, 0\}$: W の (j, k) 要素
 - W の k 列: 多値ラベル y の符号語, $(z_1(y), \dots, z_p(y))'$
 - W の j 行: 2値判別問題
 - W の j 列において
 - $W_{jk}=1$ ならば多値ラベル k ($k=1, \dots, G$)に2値ラベル1を割り当てる
 - $W_{jk}=-1$ ならば多値ラベル k ($k=1, \dots, G$)に2値ラベル-1を割り当てる
 - $W_{jk}=0$ ならば多値ラベル k ($k=1, \dots, G$)に0を割り当て2値判別には用いない
- 多値ラベル y^i を W の y^i 列 $\mathbf{z}(y^i) = (z_1(y^i), \dots, z_p(y^i))'$ で表現
 - $\{(\mathbf{x}^i, z_j(y^i)) \mid i = 1, \dots, n, z_j(y^i) \neq 0\}$ を j 番目の2値判別問題のデータセットとして, 2値判別器を訓練
 - $\Rightarrow$ 任意の2値判別機(AdaBoost, SVM等)が適用可能

多クラス問題の2値判別問題への分解

-3クラスの例

クラス 1 2 3

$$W = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

2値判別器(クラス1 vs クラス2, 3)
 2値判別器(クラス2 vs クラス1, 3)
 2値判別器(クラス3 vs クラス1, 2)
 2値判別器(クラス1 vs クラス2)
 ⋮

• j番目の2値判別器による判別結果を $\tilde{z}_j(\mathbf{x}) \in \{1, -1\}$ として, 全ての判別器の結果を纏めて $\tilde{\mathbf{z}}(\mathbf{x}) = (\tilde{z}_1(\mathbf{x}), \dots, \tilde{z}_p(\mathbf{x}))'$ あるいは単に $\tilde{\mathbf{z}}$ と記す.

多値ラベル予測

-提案法の基本アイデア

- データセット $(\mathbf{x}^i, y^i), i = 1, \dots, n$
 - ① y^i の符号語 $\mathbf{z}(y^i) = (z_1(y^i), \dots, z_p(y^i))' \in \{1, 0, -1\}^p$
 - ② 2値判別器の出力 $\tilde{\mathbf{z}}(\mathbf{x}^i) = (\tilde{z}_1(\mathbf{x}^i), \dots, \tilde{z}_p(\mathbf{x}^i))' \in \{1, -1\}^p$
- ①と②の間の関係を確率モデル $p(\tilde{\mathbf{z}}(\mathbf{x}) | y)$ (or $p(\tilde{\mathbf{z}}(\mathbf{x}) | \mathbf{z}(y))$) でモデル化して $\tilde{\mathbf{z}}_j(\mathbf{x})$ から多値ラベル y を推定 (復号) する

$$\operatorname{argmax}_y p(\mathbf{z}(y) | \tilde{\mathbf{z}}(\mathbf{x}))$$

$$p(y | \tilde{\mathbf{z}}(\mathbf{x})) \propto p(\tilde{\mathbf{z}}(\mathbf{x}) | y) p(y)$$

多値ラベル予測(復号)の例

-ハミング復号

- データセット $(\mathbf{x}^i, y^i), i = 1, \dots, n$
 - $\mathbf{z}(y^i) = (z_1(y^i), \dots, z_p(y^i))'$: y^i の符号語
 - $\tilde{\mathbf{z}}(\mathbf{x}^i) = (\tilde{z}_1(\mathbf{x}^i), \dots, \tilde{z}_p(\mathbf{x}^i))'$: 2値判別器の出力
- $\tilde{\mathbf{z}}(\mathbf{x}^i)$ と $\mathbf{z}(y)$ ($y = 1, \dots, G$) との間の拡張ハミング距離が最も近い y を予測(復号)値とする

$$\hat{y}(\mathbf{x}^i) = \arg \min_y d_{\text{ham}}(\mathbf{z}(y), \tilde{\mathbf{z}}(\mathbf{x}^i))$$

$$d_{\text{ham}}(\mathbf{z}(y), \tilde{\mathbf{z}}(\mathbf{x})) = \sum_{j=1}^p \frac{1 - z_j(y) \tilde{z}_j(\mathbf{x})}{2}$$

3つの提案モデル

- モデル1

- ラベル y とは無関係に, 符号化された値 $z_j(y) \in \{1, -1, 0\}$ と $\tilde{z}_j(\mathbf{x})$ の関係を表現
- 各2値判別器は独立と仮定する

- モデル2

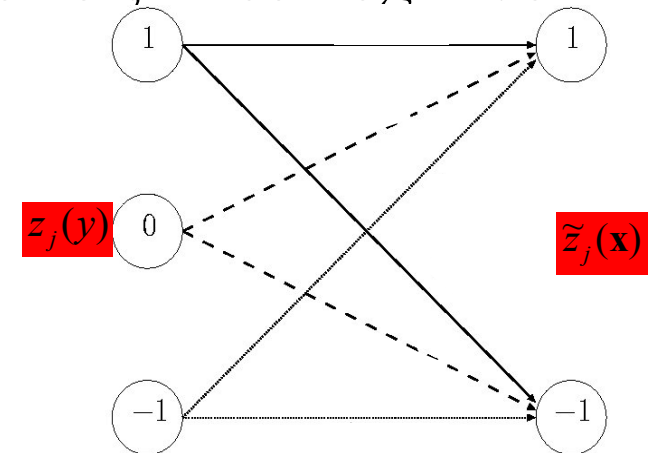
- 元のラベルが y である符号 $z_j(y) \in \{1, -1, 0\}$ と $\tilde{z}_j(\mathbf{x})$ の関係を表現
- 各2値判別器は独立と仮定する

- モデル3

- 元のラベルが y である符号 $\mathbf{z}(y) \in \{1, -1, 0\}^p$ と $\tilde{\mathbf{z}}(\mathbf{x})$ の関係を表現
- 各2値判別器間の関係性を考慮する

モデル1

- ラベル y に関係なく $z_j(y) \in \{1, -1, 0\}$ と $\tilde{z}_j(\mathbf{x})$ の関係を表現
 - 非対称な2種類の誤り率 (false positive error, false negative error) を表現可能
 - 各々の2値判別器に独立性を仮定



- j番目の判別器のモデル

$$p_1(\tilde{z}_j(\mathbf{x}) | z_j(y); \mathbf{B}_j) = \prod_{k \in \{1, -1, 0\}} \left(\frac{\exp(B_{k,j} \tilde{z}_j(\mathbf{x}))}{\exp(B_{k,j}) + \exp(-B_{k,j})} \right)^{I(z_j(y)=k)}$$

$\mathbf{B}_j = (B_{1,j}, B_{-1,j}, B_{0,j})$ はパラメーターベクトル

- 全体のモデル

$$p_1(\tilde{\mathbf{z}}(\mathbf{x}) | \mathbf{z}(y); \mathbf{B}) = \prod_{j=1}^p p_1(\tilde{z}_j(\mathbf{x}) | z_j(y); \mathbf{B}_j)$$

モデル1

- パラメーターは最尤推定法で陽に解が求まる

$$(\hat{\mathbf{B}}, \hat{\pi}) = \operatorname{argmax}_{\mathbf{B}, \pi} L(\mathbf{B}, \pi)$$

$$L(\mathbf{B}, \pi) = \sum_{i=1}^n \log p_1(\tilde{\mathbf{z}}(\mathbf{x}^i) \mid \mathbf{z}(y^i); \mathbf{B}_j) \pi_{y^i}$$

π_y : ラベルの事前分布パラメーター

$$p(y) = \pi_y, \sum_{y=1}^G \pi_y = 1$$

- ラベル予測 (MAP復号)

$$\hat{y} = \operatorname{argmax}_y p_1(\mathbf{z}(y) \mid \tilde{\mathbf{z}}(\mathbf{x}) : \hat{\mathbf{B}})$$

$$p_1(\mathbf{z}(y) \mid \tilde{\mathbf{z}}(\mathbf{x}) : \hat{\mathbf{B}}) \propto p_1(\tilde{\mathbf{z}}(\mathbf{x}) \mid \mathbf{z}(y) : \hat{\mathbf{B}}) \hat{\pi}_y$$

モデル2

- 元のラベルが y である $z_j(y) \in \{1, -1, 0\}$ と $\tilde{z}_j(\mathbf{x})$ の関係を表現
 - ラベル毎にパラメーターを持つため, モデル1と比較すると表現力は向上
 - モデル1のパラメーター数: 3×2 値判別器の数
 - モデル2のパラメーター数: クラス数 $\times$ 2値判別器の数
 - 各々の2値判別器に独立性を仮定

$$p_2(\tilde{z}_j(\mathbf{x}) | y; B_{y,j}) = \frac{\exp(B_{y,j} \tilde{z}_j(\mathbf{x}))}{\exp(B_{y,j}) + \exp(-B_{y,j})}$$

$$p_2(\tilde{\mathbf{z}}(\mathbf{x}) | y; \mathbf{B}_y) = \prod_{j=1}^p p_2(\tilde{z}_j(\mathbf{x}) | y; B_{y,j})$$

$\mathbf{B}_y = (B_{y,1}, \dots, B_{y,p})'$ はパラメーターベクトル

- 最尤推定で陽にパラメーターが求まる
- 復号はモデル1と同様

モデル3

- 元のラベルが y である符号 $\mathbf{z}(y) \in \{1, -1, 0\}^p$ と $\tilde{\mathbf{z}}(\mathbf{x})$ の関係を表現
 - 判別器間の関係性を表現

$$p_3(\tilde{\mathbf{z}}(\mathbf{x}) | y; \mathbf{B}_y, C_y) = \frac{1}{Z(y)} \exp \left(\mathbf{B}_y' \tilde{\mathbf{z}}(\mathbf{x}) + \frac{1}{2} \tilde{\mathbf{z}}(\mathbf{x})' C_y \tilde{\mathbf{z}}(\mathbf{x}) \right)$$

$\mathbf{B}_y = (B_{y,1}, \dots, B_{y,p})'$ はパラメーターベクトル
 C_y は対角成分が0の $p \times p$ 対称行列
 $Z(y)$ は正規化項

- $Z(y)$ の計算は 2^p 個のパターンの和が必要なため計算はしばしば困難
- 最尤推定では陽に解は求まらない
 - 勾配法: L を対数尤度として

$$\frac{\partial L}{\partial B_{y,j}} = E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})] - E_{p_3(\cdot|y)}[\tilde{z}_j(\mathbf{x})]$$

$$\frac{\partial L}{\partial C_{y,jk}} = E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})] - E_{p_3(\cdot|y)}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})]$$

$\tilde{p}(\tilde{\mathbf{z}} | y)$ は y が与えられた下での $\tilde{\mathbf{z}}$ の経験分布,
 $E_p[\cdot]$ は分布 $p(\tilde{\mathbf{z}})$ による期待値

モデル3

- 元のラベルが y である符号 $\mathbf{z}(y) \in \{1, -1, 0\}^p$ と $\tilde{\mathbf{z}}(\mathbf{x})$ の関係を表現
 - 判別器間の関係性を表現

$$p_3(\tilde{\mathbf{z}}(\mathbf{x}) | y; \mathbf{B}_y, C_y) = \frac{1}{Z(y)} \exp \left(\mathbf{B}_y' \tilde{\mathbf{z}}(\mathbf{x}) + \frac{1}{2} \tilde{\mathbf{z}}(\mathbf{x})' C_y \tilde{\mathbf{z}}(\mathbf{x}) \right)$$

$\mathbf{B}_y = (B_{y,1}, \dots, B_{y,p})'$ はパラメーターベクトル
 C_y は対角成分が0の $p \times p$ 対称行列

勾配中の $\tilde{p}_3(\tilde{\mathbf{z}} | y; \mathbf{B}_y, C_y)$ に関する期待値は計算困難

– $Z(y)$ の計算は 2^p 個のパターンの和が必要のため計算はしばしば困難

⇒コントラストティブ・ダイバージェンスによる近似

- 最尤推定では陽に解は求まらない
 - 勾配法: L を対数尤度として

$$\frac{\partial L}{\partial B_{y,j}} = E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})] - E_{p_3(\cdot|y)}[\tilde{z}_j(\mathbf{x})]$$

$$\frac{\partial L}{\partial C_{y,jk}} = E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})] - E_{p_3(\cdot|y)}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})]$$

$\tilde{p}(\tilde{\mathbf{z}} | y)$ は y が与えられた下での $\tilde{\mathbf{z}}$ の経験分布,

$E_p[\cdot]$ は分布 $p(\tilde{\mathbf{z}})$ による期待値

モデル3

-コントラストティブ・ダイバージェンスによるパラメーター推定

- 勾配中の $\tilde{p}_3(\tilde{\mathbf{z}} | y : \mathbf{B}_y, C_y)$ に関する期待値
 - コントラストティブ・ダイバージェンスによる近似
 - $\tilde{p}_3(\tilde{\mathbf{z}} | y : \mathbf{B}_y, C_y)$ からMCMCサンプリングした例題を用いて近似する
- $T(\mathbf{B}_y, C_y) : \tilde{p}_3(\tilde{\mathbf{z}} | y : \mathbf{B}_y, C_y)$ の遷移行列
 - 通常はMCMCサンプリングによる近似
 - マルコフ連鎖を十分に長くとり定常分布による期待値を計算
 - コントラストティブ・ダイバージェンス
 - 経験分布を初期分布としてマルコフ連鎖を少数回(典型的には1回)行って得られた分布で近似
 - $\bar{p}^t(\tilde{\mathbf{z}} | y : \mathbf{B}_y, C_y) : \text{経験分布を } T(\mathbf{B}_y, C_y) \text{ で } t \text{ 回遷移させた分布}$

$$B_{y,j}^{(r)} \leftarrow B_{y,j}^{(r-1)} + \eta \left(E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})] - E_{\bar{p}^t(\cdot|y; \mathbf{B}_y^{(r-1)}, C_y^{(r-1)})}[\tilde{z}_j(\mathbf{x})] \right)$$

$$C_{y,jk}^{(r)} \leftarrow C_{y,jk}^{(r-1)} + \eta \left(E_{\tilde{p}(\cdot|y)}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})] - E_{\bar{p}^t(\cdot|y; \mathbf{B}_y^{(r-1)}, C_y^{(r-1)})}[\tilde{z}_j(\mathbf{x})\tilde{z}_k(\mathbf{x})] \right)$$

rはパラメーターの更新回数, η は学習係数

モデル3

-ラベルの復号

- $\hat{\mathbf{B}}_y, \hat{C}_y (y=1, \dots, G)$: 推定されたパラメーター
 - 事後確率による復号

$$\hat{y} = \operatorname{argmax}_y p_3(y \mid \tilde{\mathbf{z}}(\mathbf{x}); \{\hat{\mathbf{B}}_y, \hat{C}_y\}_{y=1}^G)$$

$$p_3(y \mid \tilde{\mathbf{z}}(\mathbf{x}); \{\hat{\mathbf{B}}_y, \hat{C}_y\}_{y=1}^G) \propto p_3(\tilde{\mathbf{z}}(\mathbf{x}) \mid y; \{\hat{\mathbf{B}}_y, \hat{C}_y\}_{y=1}^G) \hat{\pi}_y$$

- 事後確率の計算も正規化項 $Z(y)$ の計算を含むため計算困難！！
 - コントラスト・ダイバージェンスによる近似でパラメーターは求まるが、正規化項の計算はできない
 - 正規化項 $Z(y)$ をサンプリングにより近似
 - $p_3(\tilde{\mathbf{z}}(\mathbf{x}) \mid y; \hat{\mathbf{B}}_y, \hat{C}_y)$ を独立分解仮定をおいた平均場近似で近似

実験

-UCIデータセット

- 各2値判別器はSVM(Rbfカーネル, 線形カーネル)
- 符号表WはFull code matrix
- 比較手法
 - multiclass-SVM (Cramer & Singer)
 - C-svm (Watson&Watkins)
 - ハミング復号
 - Loss based decoding(Allwein)
 - 指数ロス, ロジスティックロス
 - 提案モデル1
 - 提案モデル2
 - 提案モデル3 (パラメーター推定: 平均場近似, 復号: 平均場近似)
 - 提案モデル3 (パラメーター推定: コントラストティブ・ダイバージェンス, 復号: サンプリング)
 - 提案モデル3 (パラメーター推定: コントラストティブ・ダイバージェンス, 復号: 平均場近似)
- (データの70%で訓練, 残り30%で評価) × 50トライアル

実験

-UCIデータセット

•Wilcoxon matched-pair signed rank one-sided test

*:5%有意, **:1%有意

	Iris	Wine	Thyroid-disease	Balance	Car eva.
SpocSvm:rbf	4.356 ± 2.331**	2.509 ± 2.138**	4.615 ± 1.086**	8.825 ± 0.671**	4.981 ± 0.964**
SpocSvm:poly	3.511 ± 2.421*	3.455 ± 2.020**	8.636 ± 15.875**	14.254 ± 4.146**	21.785 ± 6.663**
C-Svm:rbf	4.800 ± 2.670**	1.782 ± 1.580	5.082 ± 0.959**	9.831 ± 0.889**	4.542 ± 0.807**
C-Svm:poly	3.200 ± 2.016	3.236 ± 1.919**	4.587 ± 1.706**	8.508 ± 1.832**	15.012 ± 1.045**
Ham.:rbf	4.800 ± 2.474**	1.673 ± 1.551**	5.065 ± 0.964**	9.806 ± 0.921**	5.073 ± 0.883**
Ham.:poly	3.200 ± 2.065	3.225 ± 2.133**	4.949 ± 1.468**	8.418 ± 1.942**	17.434 ± 1.123**
Allwein 1:rbf	4.356 ± 2.416**	1.673 ± 1.507**	4.600 ± 1.319**	9.524 ± 0.814**	5.692 ± 0.906**
Allwein 1:poly	5.200 ± 3.264**	2.618 ± 2.017**	7.066 ± 1.944**	8.392 ± 1.983**	17.317 ± 1.156**
Allwein 2:rbf	4.356 ± 2.416**	1.891 ± 1.641**	4.650 ± 1.285**	9.534 ± 0.789**	5.937 ± 0.895**
Allwein 2:poly	4.622 ± 3.073**	2.691 ± 2.020**	5.437 ± 1.146**	8.392 ± 1.983**	16.565 ± 1.062**
Model 1:rbf	4.756 ± 2.459**	1.673 ± 1.551**	3.833 ± 1.776	6.222 ± 1.702	4.335 ± 0.833**
Model 1:poly	3.156 ± 2.062	3.273 ± 2.173**	4.681 ± 1.614**	8.392 ± 1.983**	15.458 ± 1.110**
Model 2:rbf	4.756 ± 2.500**	1.709 ± 1.576**	3.844 ± 1.787	6.212 ± 1.703	4.062 ± 0.828**
Model 2:poly	3.156 ± 2.062	3.164 ± 2.104**	4.681 ± 1.604**	8.392 ± 1.983**	15.812 ± 1.156**
Model 3(MF):rbf	4.756 ± 2.500**	1.709 ± 1.576**	3.845 ± 1.788	6.233 ± 1.711	4.108 ± 0.848**
Model 3(MF):poly	3.156 ± 2.062	3.164 ± 2.104**	4.681 ± 1.604**	8.392 ± 1.983**	15.808 ± 1.108**
Model 3(CD+MCMC):rbf	4.711 ± 2.444**	1.709 ± 1.576	3.825 ± 1.802	6.296 ± 1.671*	3.792 ± 0.932
Model 3(CD+MCMC):poly	3.911 ± 2.441**	3.927 ± 2.184**	4.646 ± 1.641**	8.423 ± 1.924**	13.581 ± 1.062**
Model 3(CD+MF):rbf	4.756 ± 2.418**	1.636 ± 1.569	3.895 ± 1.772**	6.254 ± 1.655	4.173 ± 0.820**
Model 3(CD+MF):poly	3.111 ± 2.008	3.164 ± 2.104**	4.725 ± 1.781**	8.392 ± 1.983**	17.312 ± 1.326**

実験-バイオインフォマティックデータ

- 2値判別器はSVM(線形カーネル)
- 肺がんデータ(Harvard University)
 - 5クラス, 203サンプル, 2883特徴次元
- GCMデータ
 - 14クラス, 190サンプル, 16063特徴次元
 - Full code matrixは数が多過ぎるので, スパースランダムコードを使用

Validation error	Lung Cancer	GCM
SpocSvm:poly	$6.921 \pm 3.885^{**}$	29.937 ± 13.081
C-Svm:poly	$4.889 \pm 2.460^{**}$	$33.206 \pm 4.893^{**}$
Ham.:poly	$4.950 \pm 2.303^{**}$	$39.714 \pm 4.450^{**}$
Allwein 1:poly	$4.984 \pm 2.268^{**}$	$28.032 \pm 4.286^{**}$
Allwein 2:poly	$5.651 \pm 2.338^{**}$	$30.730 \pm 4.382^{**}$
Model 1:poly	$4.476 \pm 2.046^*$	27.048 ± 4.193
Model 2:poly	$4.921 \pm 2.362^{**}$	26.730 ± 4.331
Model 3(MF):poly	$4.921 \pm 2.362^{**}$	26.762 ± 4.326
Model 3(CD+MCMC):poly	3.873 ± 2.294	26.825 ± 4.939
Model 3(CD+MF):poly	$4.698 \pm 2.198^{**}$	$30.190 \pm 4.284^{**}$

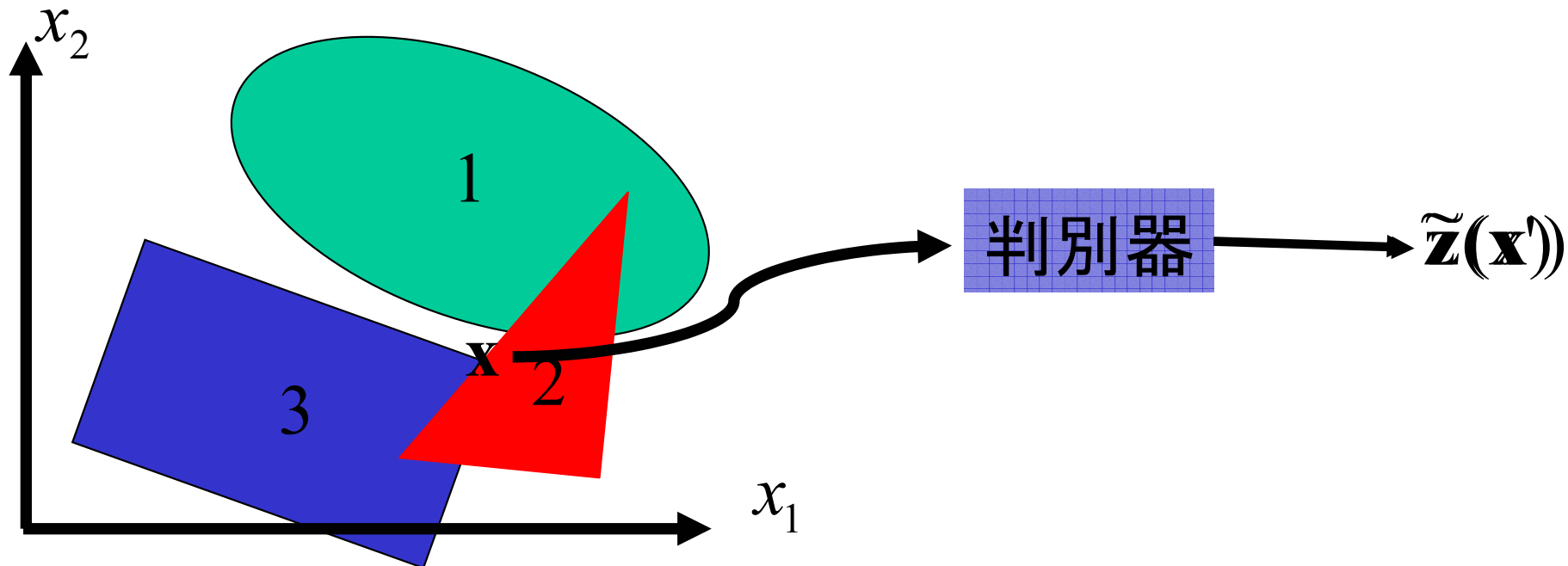
結論

- 誤り訂正符号 (ECOC) に基づく多値分類器の構成
 - 確率モデルを用いた複数の2値判別器出力の統合による多値ラベルの復号
- 判別器間の関係性を考慮したモデル
 - ボルツマンマシン
 - コントラストティブ・ダイバージェンスによる学習により高精度の分類を実現

確率モデルを持ちいた復号器の局所化

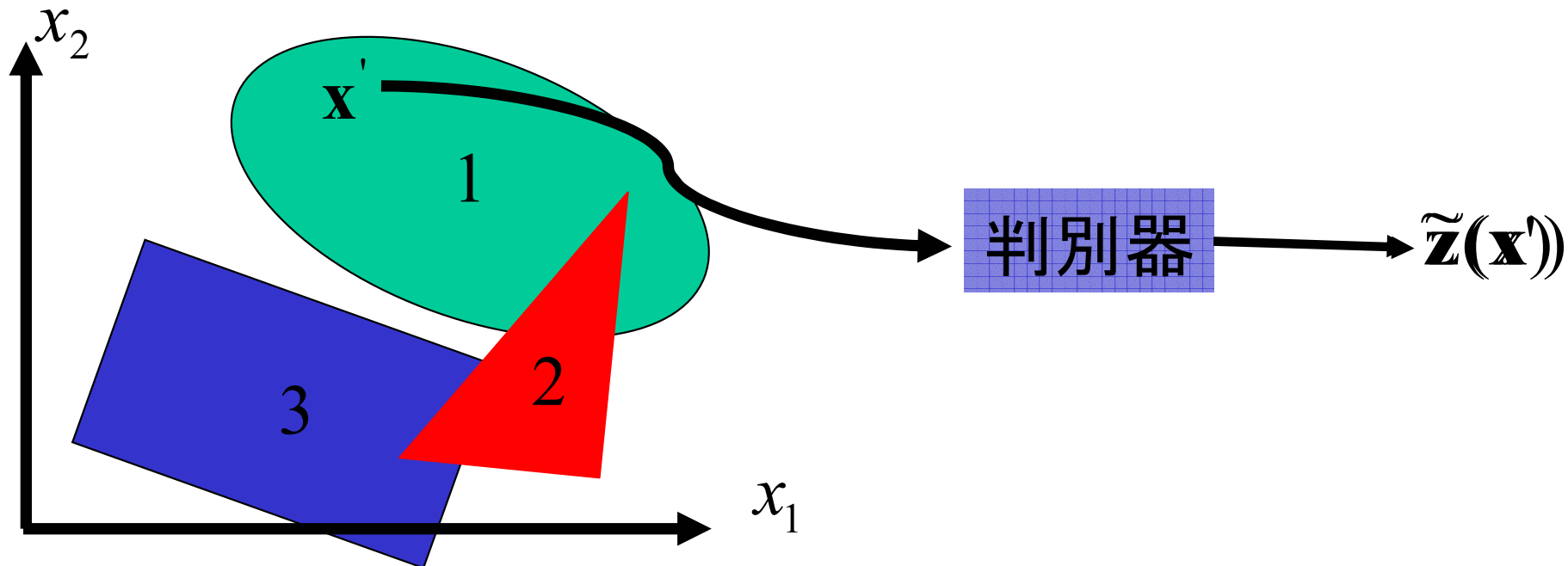
Problem of probabilistic decoder

- 前述の復号法: $p(\tilde{\mathbf{z}} | y; B)$ をモデル化
 - 入力 $\mathbf{x} \rightarrow$ 判別器 $\rightarrow \tilde{\mathbf{z}} \Leftrightarrow y$
 - $\mathbf{x}$ の情報は判別器を通して劣化している
- 最適な復号法は, 入力 $\mathbf{x}$ に依存するべき?



Problem of probabilistic decoder

- 前述の復号法: $p(\tilde{\mathbf{z}} | y; B)$ をモデル化
 - 入力 $\mathbf{x} \rightarrow$ 判別器 $\rightarrow \tilde{\mathbf{z}} \Leftrightarrow y$
 - $\mathbf{x}$ の情報は判別器を通して劣化している
- 最適な復号法は, 入力 $\mathbf{x}$ に依存するべき?



カーネル関数を用いた局所化

- Log-likelihood of probabilistic decoder

$$\sum_{i=1}^n \log p(\tilde{\mathbf{z}}^i | y^i; \mathbf{B}_{y^i}) \pi_{y^i}$$

$\Downarrow$

For an input $\mathbf{x}$, $\min_B \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^i) \log p(\tilde{\mathbf{z}}^i | y^i; \mathbf{B}_{y^i}(\mathbf{x})) \pi_{y^i}(\mathbf{x})$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-S \|\mathbf{x} - \mathbf{x}'\|^2)$$

$$B_{y,j}(\mathbf{x}) = \frac{1}{2} \log \frac{1 + J_{y,j}(\mathbf{x})}{1 - J_{y,j}(\mathbf{x})}$$
$$\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x}) I(y^i = y)$$

$$\pi_y(\mathbf{x}) = \frac{i=1}{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x})}$$

$$J_{y,j}(\mathbf{x}) = \frac{\sum_{i=1}^n I(y^i = y) k(\mathbf{x}, \mathbf{x}^i) \tilde{z}_j^i}{\sum_{i=1}^n I(y^i = y) k(\mathbf{x}, \mathbf{x}^i)}$$

- Parameters ^{$i=1$} are explicitly written, however the calculation of parameters is needed for each input $\mathbf{x}$.

カーネル関数を用いた局所化

- Log-likelihood of probabilistic decoder

$$\sum_{i=1}^n \log p(\tilde{\mathbf{z}}^i | y^i; \mathbf{B}_{y^i}) \pi_{y^i}$$

⇓

For an input $\mathbf{x}$ min $\sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^i) \log p(\tilde{\mathbf{z}}^i | y^i; \mathbf{B}_{y^i}(\mathbf{x})) \pi_{y^i}(\mathbf{x})$

For a new input $\mathbf{x}$,

Calculate $P(y|z; \mathbf{B}(\mathbf{x}), \pi(\mathbf{x}))$ for decoding

$$B_{y,j}(\mathbf{x}) = \frac{1}{2} \log \frac{1 + J_{y,j}(\mathbf{x})}{1 - J_{y,j}(\mathbf{x})}$$

$$\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x}) I(y^i = y)$$

$$J_{y,j}(\mathbf{x}) = \frac{\sum_{i=1}^n I(y^i = y) k(\mathbf{x}, \mathbf{x}^i) z_j}{\sum_{i=1}^n I(y^i = y) k(\mathbf{x}, \mathbf{x}^i)}$$

$$\pi_y(\mathbf{x}) = \frac{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x}) I(y^i = y)}{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x})}$$

- Parameters are explicitly written, however the calculation of parameters is needed for each input $\mathbf{x}$.

推定方程式の不偏性

Estimating equation of θ : $\sum_{i=1}^n \Psi(\theta; \mathbf{x}^i) = 0 \Rightarrow \hat{\theta}$

Ex. 最尤推定法

$$L(\theta) = \sum_{i=1}^n \log p(\mathbf{x}^i; \theta) \Rightarrow 0 = \frac{\partial L(\theta)}{\partial \theta} = \sum_{i=1}^n \Psi(\theta; \mathbf{x}^i)$$

- 不偏な推定方程式

$$\Leftrightarrow 0 = \int \Psi(\theta_0; \mathbf{x}) p(\mathbf{x}; \theta_0) d\mathbf{x} \quad p(\mathbf{x}; \theta_0) : \mathbf{x} \text{ の密度関数}$$

Ex. 最尤推定法の推定方程式は不偏

- 不偏性が満たされていないと推定量の性能が低下する事が知られている

局所化復号器の推定方程式

$$L(\mathbf{B}, \pi | \mathbf{x}) = \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^i) \log p(\tilde{\mathbf{z}}^i | y^i; \mathbf{B}_{y^i}(\mathbf{x})) \pi_{y^i}(\mathbf{x})$$

$$0 = \frac{\partial L(\mathbf{B}, \pi | \mathbf{x})}{\partial B_{y,j}(\mathbf{x})} = \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^i) I(y^i = y) \left\{ \tilde{z}_j(\mathbf{x}^i) - \frac{e^{B_{y,j}(\mathbf{x})} - e^{-B_{y,j}(\mathbf{x})}}{e^{B_{y,j}(\mathbf{x})} + e^{-B_{y,j}(\mathbf{x})}} \right\}$$

$$= \sum_{i: y^i = y} \Psi_1(B_{y,j}; \mathbf{x}^i) \quad (y = 1, \dots, G)$$

$$0 = \frac{\partial L(\mathbf{B}, \pi | \mathbf{x})}{\partial \pi_y} = \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}^i) \left\{ \frac{I(y^i = y)}{\pi_y(\mathbf{x})} - \frac{I(y^i = G)}{\pi_G(\mathbf{x})} \right\}$$

$$= \sum_{i=1}^n \Psi_2(\pi; \mathbf{x}^i, y^i) \quad (y = 1, \dots, G-1)$$

- In general, estimating equation of localized method is biased, which leads to degradation of performance.

バイアス補正

- For parameter $B_{y,j}$, the estimating equation is

$$\sum_{i:y^i=y} \Psi_1(B_{y,j}; \mathbf{x}^i) = 0$$
$$\Psi_1(B_{y,j}; \mathbf{x}^i) = k(\mathbf{x}, \mathbf{x}^i) \left\{ \tilde{z}_j(\mathbf{x}^i) - \frac{e^{B_{y,j}(\mathbf{x})} - e^{-B_{y,j}(\mathbf{x})}}{e^{B_{y,j}(\mathbf{x})} + e^{-B_{y,j}(\mathbf{x})}} \right\}$$
$$\int \Psi_1(B_{y,j}; \mathbf{x}^i) p(\tilde{z}_j(\mathbf{x}^i) | y; B_{y,j}(\mathbf{x})) d\mathbf{x}$$
$$= \frac{2n_y}{(e^{B_{y,j}(\mathbf{x})} + e^{-B_{y,j}(\mathbf{x})})^2} m_j(\mathbf{x}), m_j(\mathbf{x}) = \int \tilde{z}_j(\mathbf{x}') k(\mathbf{x}', \mathbf{x}) d\mathbf{x}'$$

- Corrected estimating equation

$$\sum_{i:y^i=y} \Psi_1(B_{y,j}; \mathbf{x}^i) = \frac{2n_y m_j(\mathbf{x})}{(e^{B_{y,j}(\mathbf{x})} + e^{-B_{y,j}(\mathbf{x})})^2}$$

Unbiased localized decoder

$$B_{y,j}(\mathbf{x}) = \frac{1}{2} \log \frac{1 + J_{y,j}(\mathbf{x})}{1 - J_{y,j}(\mathbf{x})} + \frac{1}{2} \log \frac{1 - r_{y,j}(\mathbf{x}) + \sqrt{(1 + r_{y,j}(\mathbf{x}))^2 - 2r_{y,j}(\mathbf{x})(J_{y,j}(\mathbf{x}) + 1)}}{1 + r_{y,j}(\mathbf{x}) + \sqrt{(1 + r_{y,j}(\mathbf{x}))^2 - 2r_{y,j}(\mathbf{x})(J_{y,j}(\mathbf{x}) + 1)}}$$

$$r_{y,j}(\mathbf{x}) = \frac{n_y m_{y,j}(\mathbf{x})}{\sum_i k(\mathbf{x}^i, \mathbf{x})}$$

• For parameter π_y , the estimating equation is written as

$$0 = \sum_{i=1}^n \Psi_2(\pi_y; \mathbf{x}^i, y^i), \Psi_2(\pi_y; \mathbf{x}^i, y^i) = k(\mathbf{x}, \mathbf{x}^i) \left\{ \frac{I(y^i = y)}{\pi_y(\mathbf{x})} - \frac{I(y^i = G)}{\pi_G(\mathbf{x})} \right\}$$

$$\sum_{y'} k(\mathbf{x}, \mathbf{x}^i) \Psi_2(\pi_{y'}; \mathbf{x}^i, y') \pi_{y'} = 0 \Rightarrow \text{Unbias} \quad \pi_y(\mathbf{x}) = \frac{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x}) I(y^i = y)}{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x})}$$

Unbiased localized decoder

$$B_{y,j}(\mathbf{x}) = \frac{1}{2} \log \frac{1 + J_{y,j}(\mathbf{x})}{1 - J_{y,j}(\mathbf{x})} + \frac{1}{2} \log \frac{1 - r_{y,j}(\mathbf{x}) + \sqrt{(1 + r_{y,j}(\mathbf{x}))^2 - 2r_{y,j}(\mathbf{x})(J_{y,j}(\mathbf{x}) + 1)}}{(1 + r_{y,j}(\mathbf{x}))^2 - 2r_{y,j}(\mathbf{x})(J_{y,j}(\mathbf{x}) + 1)}$$

For a new input $\mathbf{x}$,

by plugging parameters into model

• For 1 Calculate $P(y|z; B(\mathbf{x}), \pi(\mathbf{x}))$ for decoding

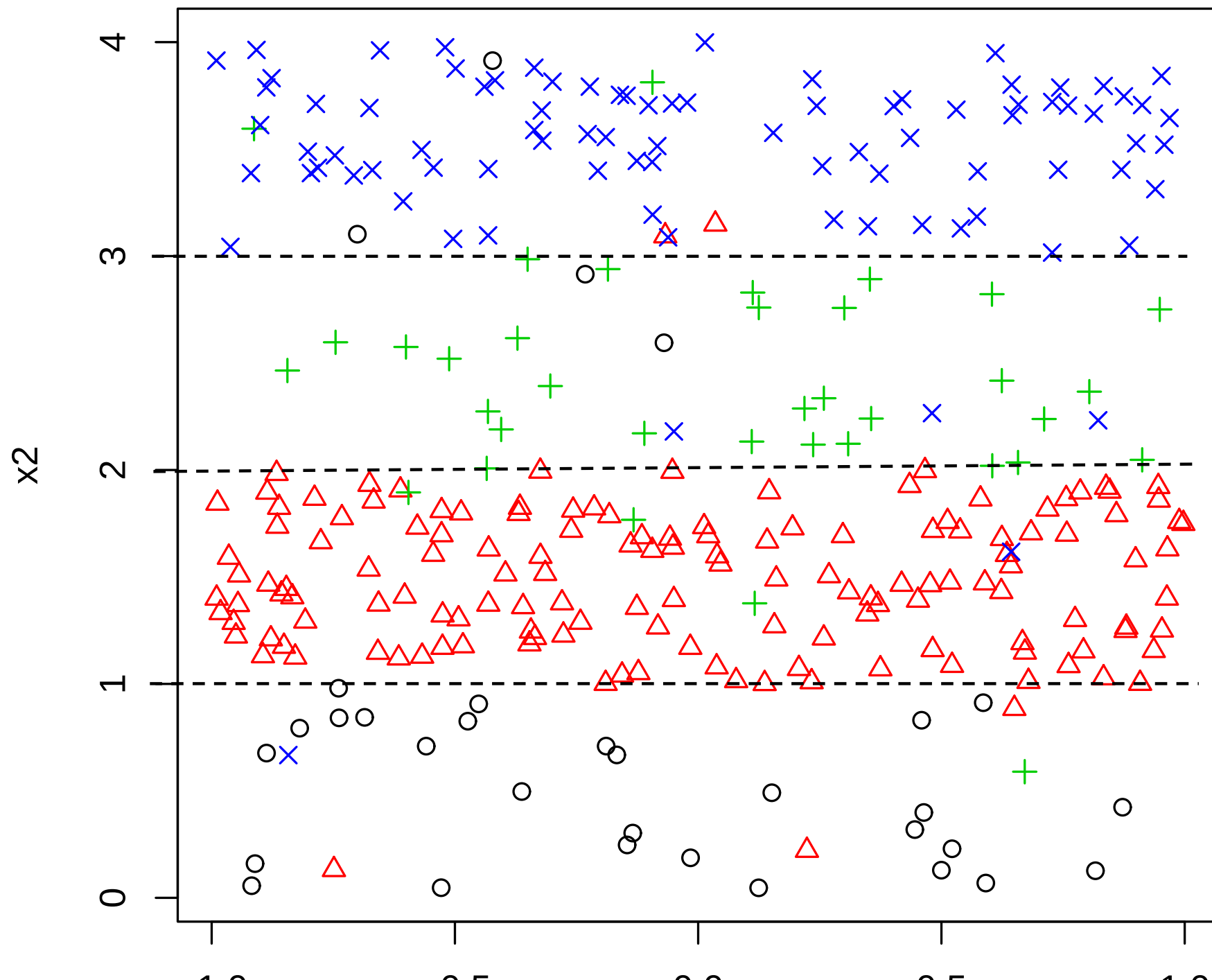
$$0 = \sum_{i=1}^n \Psi_2(\pi_y; \mathbf{x}^i, y^i), \Psi_2(\pi_y; \mathbf{x}^i, y^i) = k(\mathbf{x}, \mathbf{x}^i) \left\{ \frac{I(y^i = y)}{\pi_y(\mathbf{x})} - \frac{I(y^i = G)}{\pi_G(\mathbf{x})} \right\}$$

$$\sum_{y'} k(\mathbf{x}, \mathbf{x}^i) \Psi_2(\pi_{y'}; \mathbf{x}^i, y') \pi_{y'} = 0 \Rightarrow \text{Unbias} \quad \pi_y(\mathbf{x}) = \frac{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x}) I(y^i = y)}{\sum_{i=1}^n k(\mathbf{x}^i, \mathbf{x})}$$

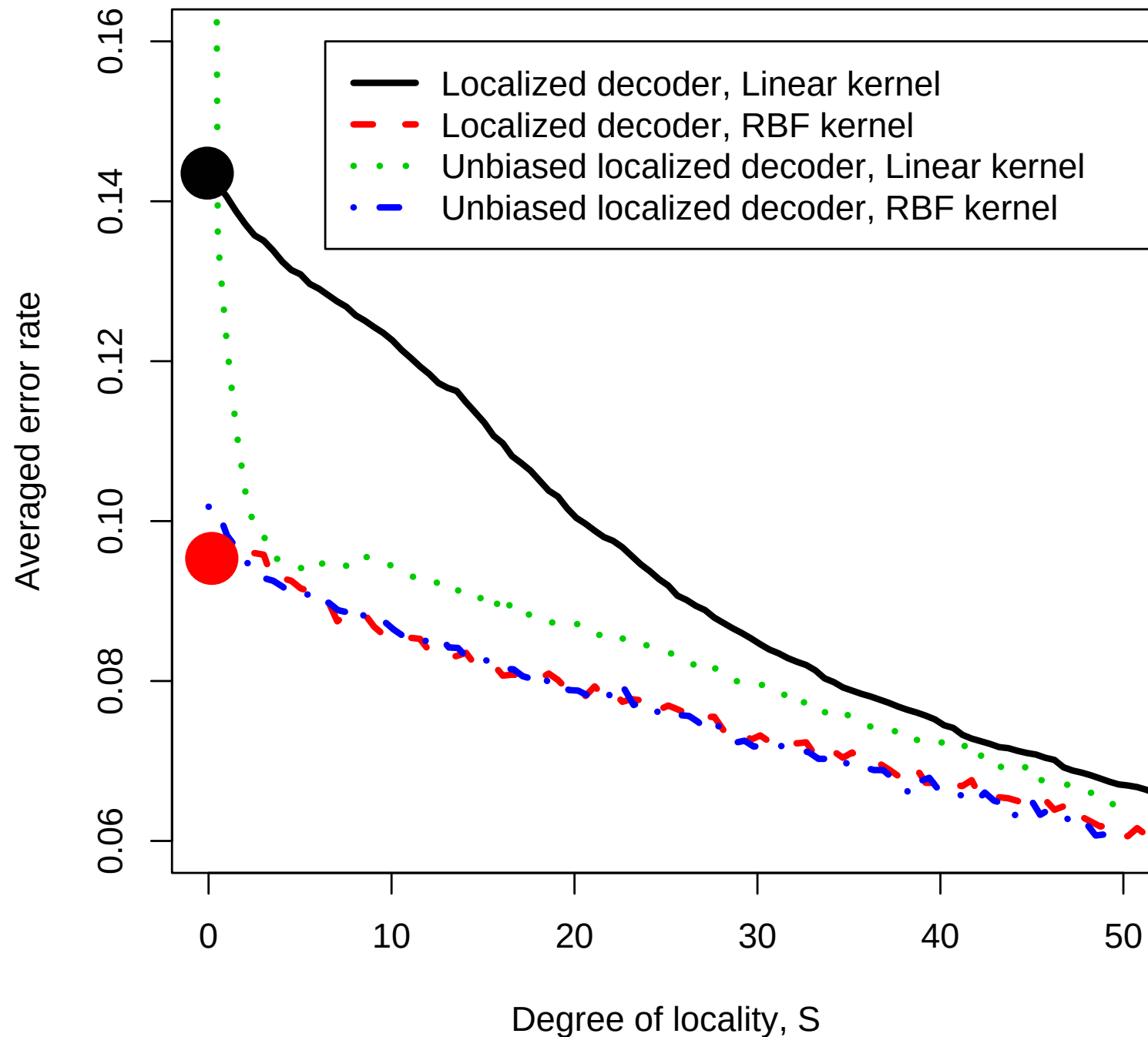
数値実験：人工データ

- データセット：訓練300, テスト3000 × 50セット
- 各2値判別器はSVM
 - カーネルは線形とRBF
- 比較対象：多値版SVM (C-svm, Spoc-svm)
 - カーネルは線形とRBF
- Hamming復号

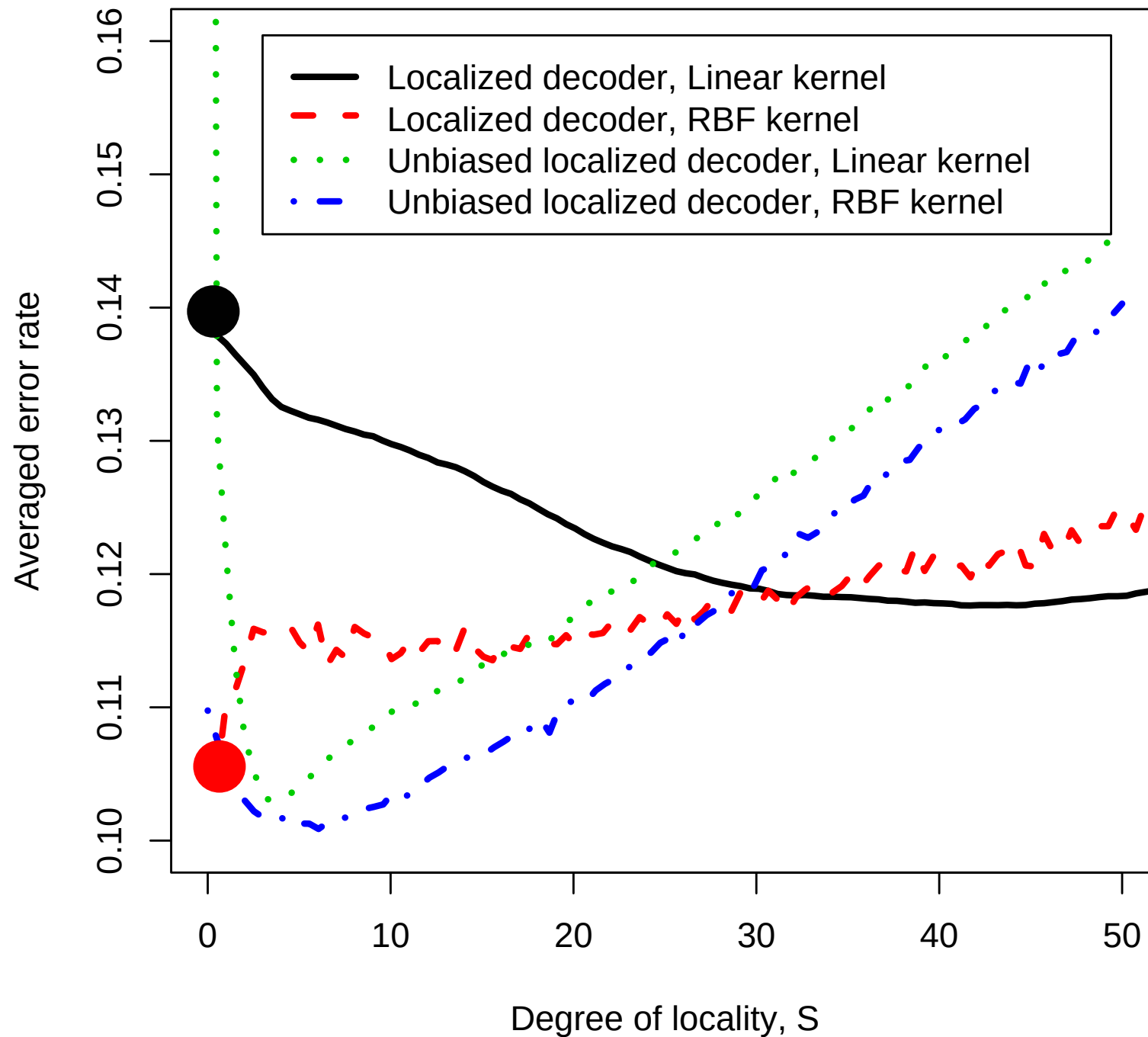
typical dataset



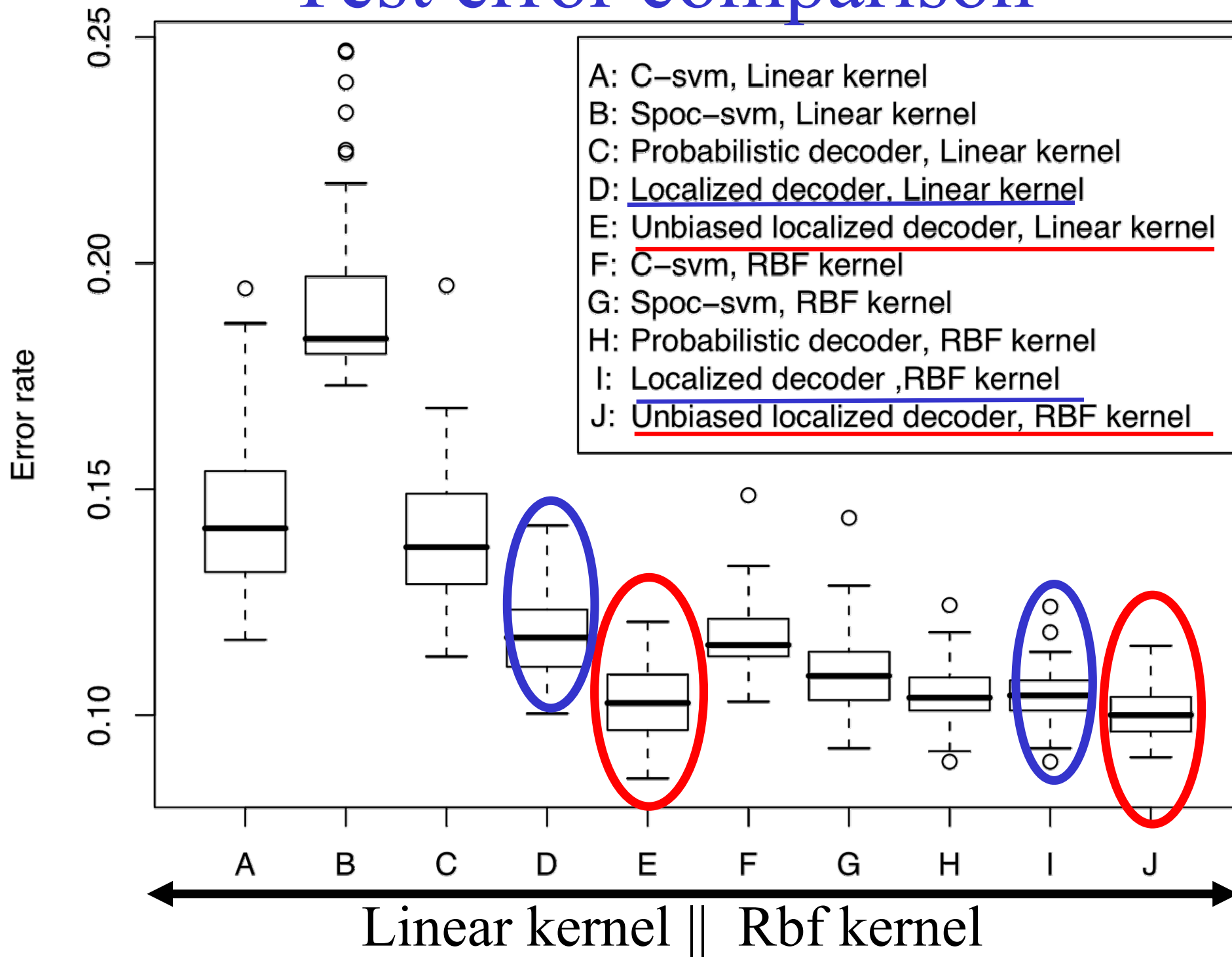
Training error against degree of localization



Test error against degree of localization S



Test error comparison



局所化復号器の結論

- 確率モデルによる復号法の局所化
 - カーネル関数の導入
- バイアス補正
 - 不偏局所化復号器 (Unbiased localized decoder)
 - Unbiased localized decoderはカーネル関数の形に関わらず他の方法を優越
- Future work
 - 局所化の度合いを決めるハイパーパラメーターSの決定
 - 実データへの適用

TERNARY BRADLEY-TERRY MODELに基づく多値判別

概要

1. BT-modelに基づく多値判別

1. Hasite & Tibshirani, Zadrozny, Yukinawa
2. Problematic point of conventional methods

2. Ternary BTモデルを用いた改良

1. Ternary probabilistic guess
2. Ternary AdaBoost

多値判別の設定

- $\mathbf{x}$: 入力, $y \in \{1, L, G\}$: ラベル
 - データセット: $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$
- $F(\mathbf{x}, y)$: 判別関数
 - ラベル予測
 - どうやって $F(\mathbf{x}, y)$ を構成するか? $\Rightarrow \operatorname{argmax}_y F(\mathbf{x}, y)$
 - 多値判別問題を複数の2値判別問題に分割
 - 各々構成した2値判別器の出力を統合
- 今までの設定との違い
 - 今まで: $\tilde{\mathbf{z}}(\mathbf{x}) \in \{1, -1\}^p$ から y を構成
 - これから: 2値判別関数 $f_j(\mathbf{x}) \in R (j = 1, L, p)$ から y を構成

多クラス問題の分解

- 符号表 $W \in \{+1, -1, 0\}^{p \times G}$, $p \times G$ 行列

$$C_j^k = \{g : W_{jg} = k\}$$

j番目の判別問題においてラベルkを持つクラスの集合

$$C_j^{+1} \cup C_j^{-1} \cup C_j^0 \in \{1, L, G\}$$

Class 1 2 3

$$W = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

判別関数 $f_1 \in R$

$$(C_1^{+1} = \{1\}, C_1^{-1} = \{2, 3\})$$

判別関数 $f_4 \in R$

$$(C_4^{+1} = \{1\}, C_4^{-1} = \{2\}, C_4^0 = \{3\})$$

多クラス問題の分解

- 符号表 $W \in \{+1, -1, 0\}^{p \times G}$, $p \times G$ 行列

$$C_j^k = \{g : W_{jg} = k\}$$

j番目の判別問題においてラベルkを持つクラスの集合

$$C_j^{+1} \cup C_j^{-1} \cup C_j^0 \in \{1, L, G\}$$

Class 1 2 3

$$W = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

判別関数 $f_1 \in R$

$$f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x}) \quad (C_1^{-1} = \{2, 3\})$$

判別関数 $f_4 \in R$

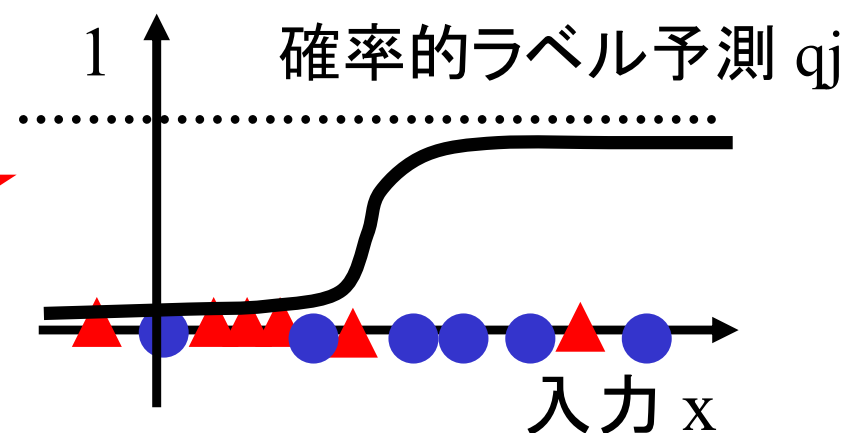
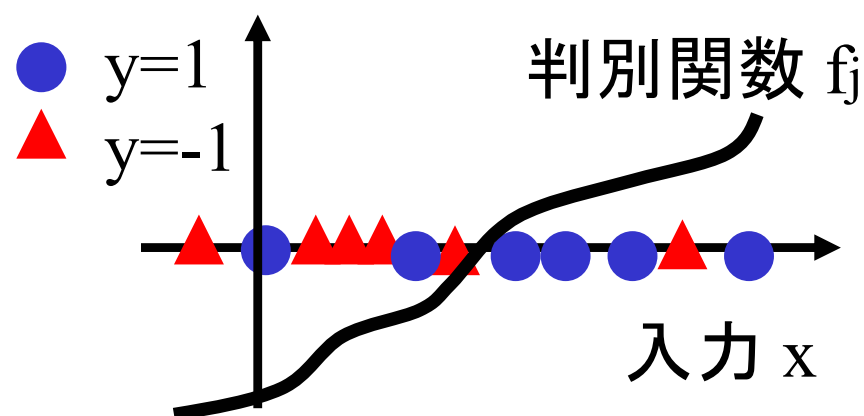
$$(C_4^{+1} = \{1\}, C_4^{-1} = \{2\}, C_4^0 = \{3\})$$

Bradley-Terry Modelに基づく多値判別

-Step 1. ラベルの確率的予測 (probabilistic guess)

• $f_j(\mathbf{x})$: j 番目の判別関数

⇒ 入力 $\mathbf{x}$ に対する2値ラベルの確率的予測



$$q_j^{+1}(\mathbf{x}) = \frac{1}{1 + \exp(a_j + b_j f_j(\mathbf{x}))}, f_j(\mathbf{x}) \in R$$

$$\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x})), q_j^{-1}(\mathbf{x}) = 1 - q_j^{+1}(\mathbf{x})$$

$0 \leq q_j(\mathbf{x}) \leq 1$: probability of event of which input $\mathbf{x}$ belongs to class set C_j^{+1}

Bradley-Terry Modelに基づく多値判別

-Step 1. ラベルの確率的予測 (probabilistic guess)

• $f_j(\mathbf{x})$: j 番目の判別関数

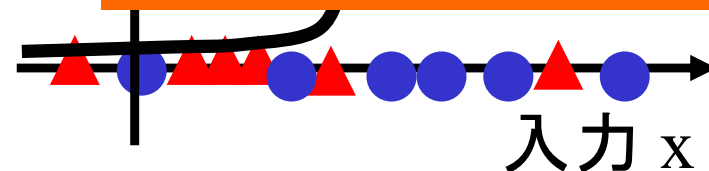
⇒ 入力 $\mathbf{x}$ に対する2値ラベルの確率的予測

パラメーター推定

$$\max_{a_j, b_j} \sum_{i=1}^n \sum_{k \in \{+1, -1\}} I(y^i \in C_j^k) \log q_j^k(\mathbf{x}^i)$$

入力 $\mathbf{x}$

入力 $\mathbf{x}$ に対する
 $(q_1(\mathbf{x}), \dots, q_j(\mathbf{x}), \dots, q_p(\mathbf{x}))$
 確率的ラベル予測



$$q_j^{+1}(\mathbf{x}) = \frac{1}{1 + \exp(a_j + b_j f_j(\mathbf{x}))}, f_j(\mathbf{x}) \in R$$

$$\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x})), q_j^{-1}(\mathbf{x}) = 1 - q_j^{+1}(\mathbf{x})$$

$0 \leq q_j(\mathbf{x}) \leq 1$: probability of event of which input $\mathbf{x}$ belongs to class set C_j^{+1}

Bradley-Terry Modelに基づく多値判別-

Step 2 復号

- 判別関数 $f_j(\mathbf{x})$

⇒ 入力 $\mathbf{x}$ に対する確率的予測 $\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x}))$

もし真のクラス事後確率 $\mathbf{p}(y | \mathbf{x}) = (p_1, \dots, p_G)$ をしていたら

$$\mathbf{p}(y | \mathbf{x}) \Rightarrow_j = (\pi_j^{+1}, \pi_j^{-1}), \pi_j^{+1} = \frac{\sum_{y \in C_j^{+1}} p_y}{\sum_{y \in C_j^{+1}} p_y + \sum_{y \in C_j^{-1}} p_y}, \pi_j^{-1} = 1 - \pi_j^{+1}$$

$\mathbf{q}_j(\mathbf{x})$ と π_j を比較することで $\mathbf{p}(y | \mathbf{x})$ を推定

- Hastie et.al. $\hat{\mathbf{p}}(\mathbf{x}) = (\hat{p}_1, \dots, \hat{p}_G) = \arg \min_{\mathbf{p}} \sum_j w_j D(\mathbf{q}_j, \pi_j)$

w_j : j番目の判別器の重み

Bradley-Terry Modelに基づく多値判別-

Step 2 復号

- 判別関数 $f_j(\mathbf{x})$

⇒ 入力 $\mathbf{x}$ に対する確率的予測 $\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x}))$

もし真のクラス事後確率 $\mathbf{p}(y | \mathbf{x}) = (p_1, \dots, p_G)$ をしていたら

$$\mathbf{p}(y | \mathbf{x}) \Rightarrow_j = (\pi_j^{+1}, \pi_j^{-1}), \pi_j^{+1} = \frac{\sum_{y \in C_j^{+1}} p_y}{\sum_{y \in C_j^{+1}} p_y + \sum_{y \in C_j^{-1}} p_y}, \pi_j^{-1} = 1 - \pi_j^{+1}$$

$\mathbf{q}_j(\mathbf{x})$ と π_j を比較することで $\mathbf{p}(y | \mathbf{x})$ を推定

- Hastie et.al. $\hat{\mathbf{p}}(\mathbf{x}) = (\hat{p}_1, \dots, \hat{p}_G) = \arg \min_{\mathbf{p}} \sum_j w_j D(\mathbf{q}_j, \pi_j)$
 $\arg \max_y \{ \hat{p}_1, \dots, \hat{p}_G \}$

w_j : j 番目の判別器の重み

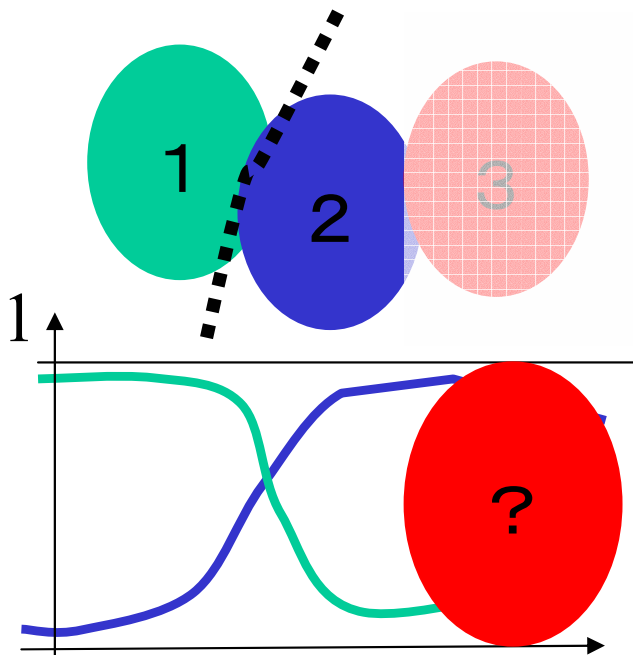
従来法の問題点

$$W = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ \hline 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

- 訓練時
 - クラス3の例題は学習に用いない
 - 2値判別器はクラス1とクラス2の例題で訓練
 - ⇒ 確率的予測 $\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x}))$

- テスト時(新たな入力 $\mathbf{x}$ に対して)
 - クラスラベルは未知.
 - もし $\mathbf{x}$ のラベルが1あるいは2ならば, 確率的予測は適切かも
 - もし $\mathbf{x}$ のラベルが3の場合, 訓練時にクラス3の情報を全く持っていないので, 確率的予測は全く役立たず. そもそも符号は1でも-1でもなく0

- **3クラス(Ternary)の確率的予測が必要**



概要

1. BT-modelに基づく多値判別

1. Hasite & Tibshirani, Zadrozny, Yukinawa
2. Problematic point of conventional methods

2. Ternary BTモデルを用いた改良

1. Ternary probabilistic guess
2. Ternary AdaBoost

Ternary BTモデルに基づく復号

-3クラスの確率的クラス予測

- 3クラスのロジスティックモデル

$$\mathbf{q}_j(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^0(\mathbf{x}), q_j^{-1}(\mathbf{x})) = \left(\frac{\exp(F_j^{+1}(\mathbf{x}))}{Z(\mathbf{x})}, \frac{1}{Z(\mathbf{x})}, \frac{\exp(F_j^{-1}(\mathbf{x}))}{Z(\mathbf{x})} \right)$$
$$Z(\mathbf{x}) = 1 + \exp(F_j^{+1}(\mathbf{x})) + \exp(F_j^{-1}(\mathbf{x}))$$
$$F_j^k(\mathbf{x}) = a_j^k + b_j^k f_j^k(\mathbf{x}), k \in \{+1, 0, -1\}$$

– Note: 符号表が0を含まない場合は従来と同様に2クラスの確率的予測 $\mathbf{q}_l(\mathbf{x}) = (q_j^{+1}(\mathbf{x}), q_j^{-1}(\mathbf{x}))$ を用いる

- パラメーター推定は最尤推定で行う

3クラスの確率的クラス予測構成における問題点

- 2値判別器 $f_j(x)$
 - 符号+1あるいは -1を持つ例題で訓練
 - $f_j(x)$ には符号0を持つクラスの情報に含まれない

⇒そのまま3クラスのロジスティックモデルを使ってもダメ

3クラス判別が必要？

⇒”多値判別問題の2値判別問題への分解“？

- 3クラスの情報を持った2値判別器
 - Ternary-AdaBoost (T-AdaBoost)

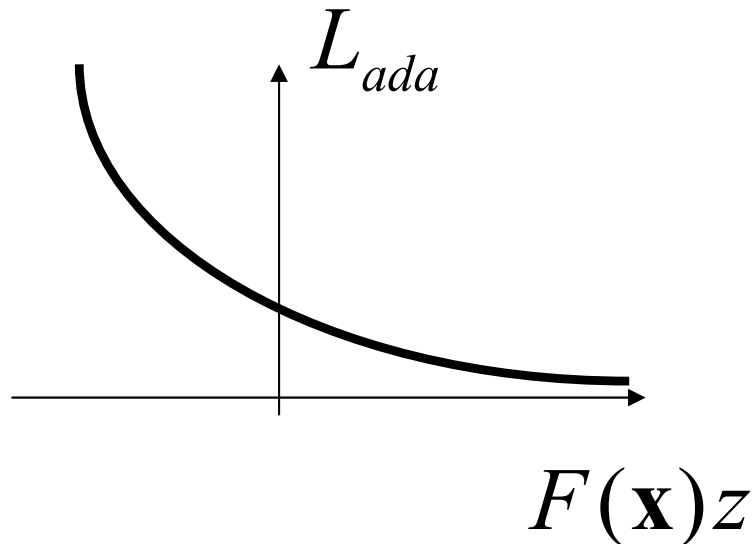
⇒従来のAdaBoostの変形

AdaBoost

- データセット: $\{\mathbf{x}^i, z^i\}_{i=1}^n, z \in \{+1, -1\}$
- 以下のロス関数の逐次最小化

$$\hat{F}(\mathbf{x}) = \arg \min_F L_{ada}(F) \Rightarrow \text{sgn}(\hat{F}(\mathbf{x}))$$

$$L_{ada}(F) = \sum_{i: z^i \neq 0} \exp(-F(\mathbf{x}^i)z^i)$$



• もし $F(\mathbf{x})$ が z を正しく判別していれば F と z は同符号

$$\Rightarrow F(\mathbf{x})z > 0$$

• もし $F(\mathbf{x})$ が z を間違って判別していれば, F と z は異符号

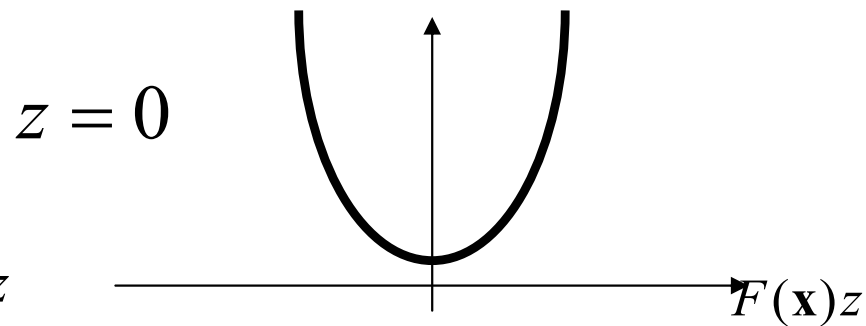
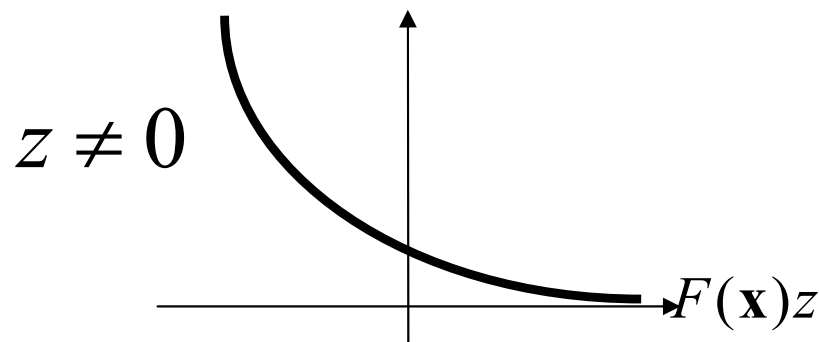
$$\Rightarrow F(\mathbf{x})z < 0$$

Ternary-AdaBoost

- 目的: Ternary AdaBoostの判別関数は
 - 符号 ± 1 を持つ入力に対しては大きな値を持つ
 - 符号0を持つ入力に対しては小さな値を持つ
- Dataset: $\{\mathbf{x}^i, z^i\}_{i=1}^n, z \in \{+1, 0, -1\}$
- Ternary-AdaBoost

$$\arg \min_F \sum_{i=1}^n U(F(\mathbf{x}^i); z^i)$$

$$U(\xi; z) = I(z \neq 0) \exp(-z\xi) + I(z = 0) \lambda(\exp(\xi) + \exp(-\xi))$$



Ternary BT モデルに基づく復号

1. 入力 $\mathbf{x}$ に対して, Ternary-AdaBoostの判別関数を用いて3クラスの確率的ラベル予測値を計算 $\mathbf{q}_j(\mathbf{x})$
2. $\mathbf{q}_j(\mathbf{x})$ と真のクラス事後確率から計算した π_j のKLダイバージェンスを最小化する事で, クラス事後確率 $\mathbf{p}(\mathbf{y}|\mathbf{x})=(p_1, \dots, p_G)$ を計算

$$\hat{\mathbf{p}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{p}} \sum_j w_j D(\mathbf{q}_j, \pi_j)$$

w_j : Weight for j - th binary problem.

$$\pi_j = \left(\sum_{y \in C_j^{+1}} p_y, \sum_{y \in C_j^0} p_y, \sum_{y \in C_j^{-1}} p_y \right) :$$

真のクラス事後確率が分かっている
として計算した確率的予測値

3. $\operatorname{argmax}_y \hat{p}_y(\mathbf{x})$ でラベルを復号

KLの最小化は訓練, テストの各サンプルに対して必要
となるため, 計算コストが大きい

計算コストの削減

- KL-ダイバージェンスを2乗誤差で代替

$$\Rightarrow \hat{\mathbf{p}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{p}} \sum_j w_j \|\pi_j - \mathbf{q}_j(\mathbf{x})\|^2$$

$\|\bullet\|$ is euclid norm.

$$\hat{\mathbf{p}} = (X^T V X)^{-1} \left(X^T V Q - \frac{\mathbf{1}^T (X^T V X)^{-1} X^T V Q}{\mathbf{1}^T (X^T V X)^{-1} \mathbf{1}} \mathbf{1} \right)$$

X : Matrix determined by code matrix W

Q : Vector in which $q_j(\mathbf{x}) (j = 1, \dots, p)$ are aligned

V : Diagonal matrix whose component is w_j .

計算コストの削減

- KL-ダイバージェンスを2乗誤差で代替

$$\Rightarrow \hat{\mathbf{p}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{p}} \sum_j w_j \|\pi_j - \mathbf{q}_j(\mathbf{x})\|^2$$

$\|\bullet\|$ is euclid norm.

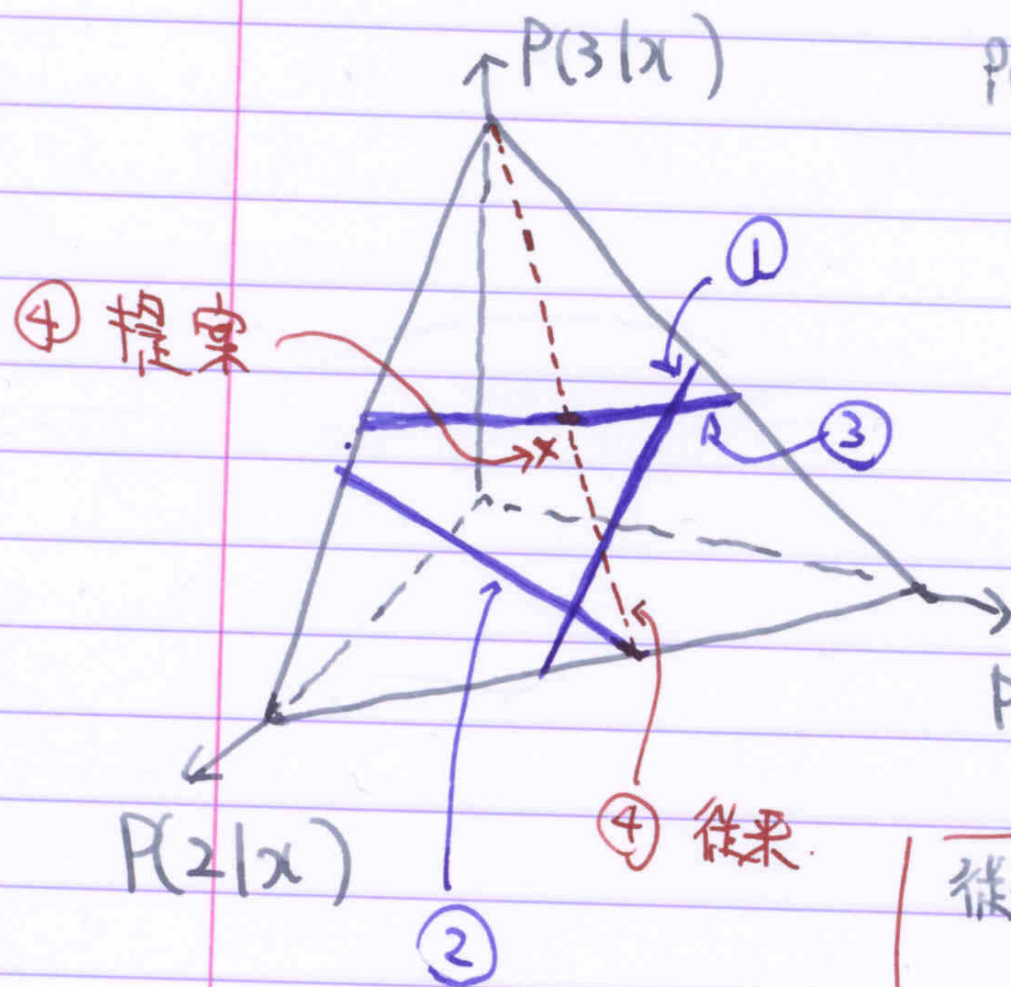
$$\hat{\mathbf{p}} = \underline{(X^T V X)^{-1} X^T V Q}$$

$$\sum_y p_y = 1 \text{ is not assumed.}$$

$$\underline{(X^T V X)^{-1} X^T V} \text{ is not depend on probabilistic guess } Q$$

従来のBTモデルに基づく復号では, 上記の最適化問題は陽には解けない.

幾何的解釋



$P(1|x)$ vs $P(2|x) + P(3|x)$

$\rightarrow \rightarrow \rightarrow$	①	②	③
①	+	-	-
②	-	+	-
③	-	-	+
$P(1 x)$	+	+	-

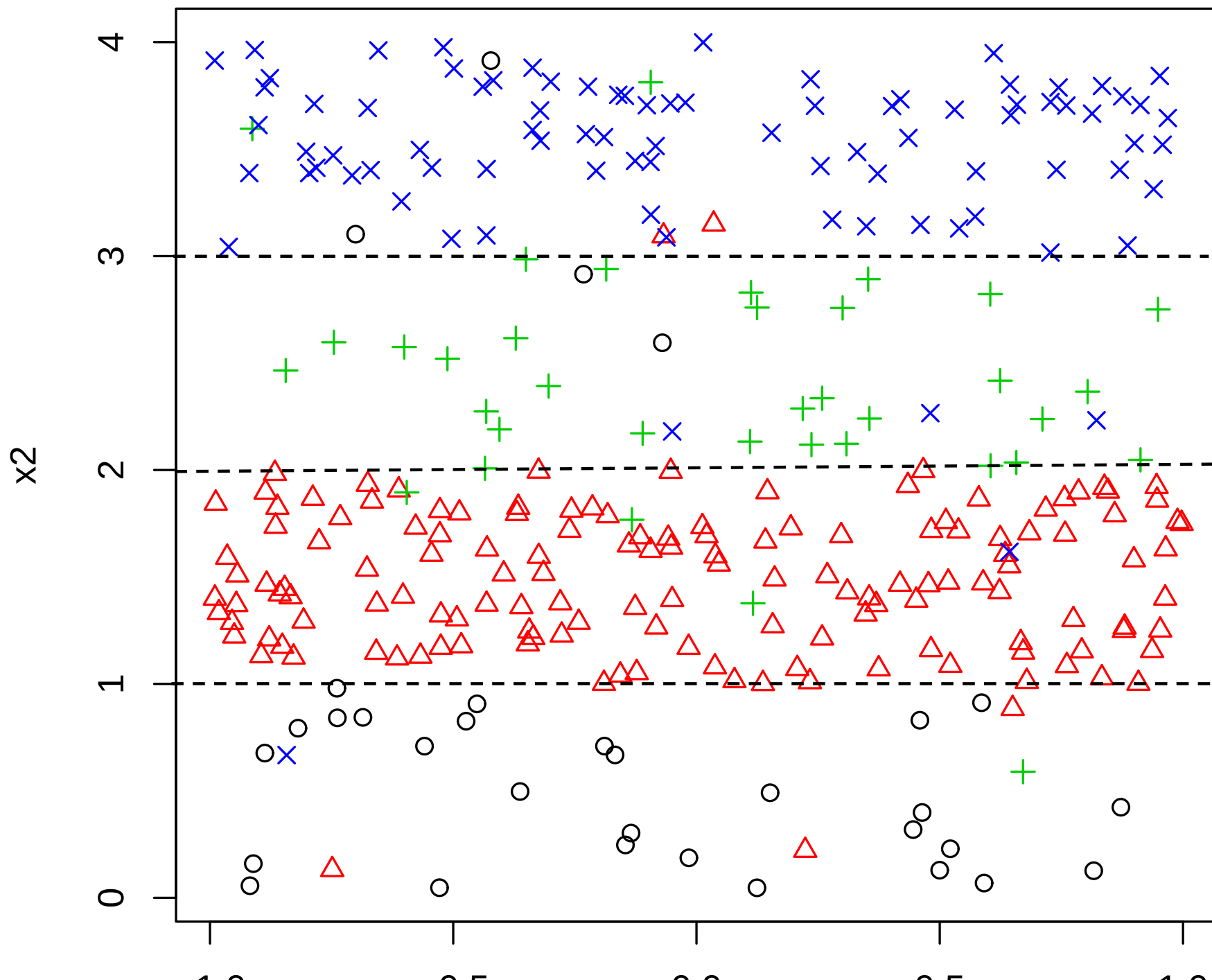
結果 $\frac{P(1|x)}{P(1|x) + P(2|x)}$ vs $\frac{P(2|x)}{P(1|x) + P(2|x)}$

結果 $P(1|x)$ vs $P(2|x)$ vs $P(3|x)$

Experiment: synthetic dataset

- 4クラス問題
 - (訓練:300, テスト:3000) × 50
- 重み $w_j \Rightarrow$ (I) $w_j = 1$, (II) $w_j = \sum_{i=1}^n I(y^i \in C_j^{+1} \text{ or } y^i \in C_j^{-1})$
- 手法:
 - BT: Bradley-Terry model-based decoding with AdaBoost (←Conventional method)
 - BT model-based decoding with Ternary-AdaBoost
 - TBT: Ternary BT with Ternary-AdaBoost (←Proposed method)
 - TBT(S): Ternary BT with Ternary-AdaBoost, squared loss
 - TBT(A): Approximated version of TBT(S),
without constraint $\sum_y p_y = 1$

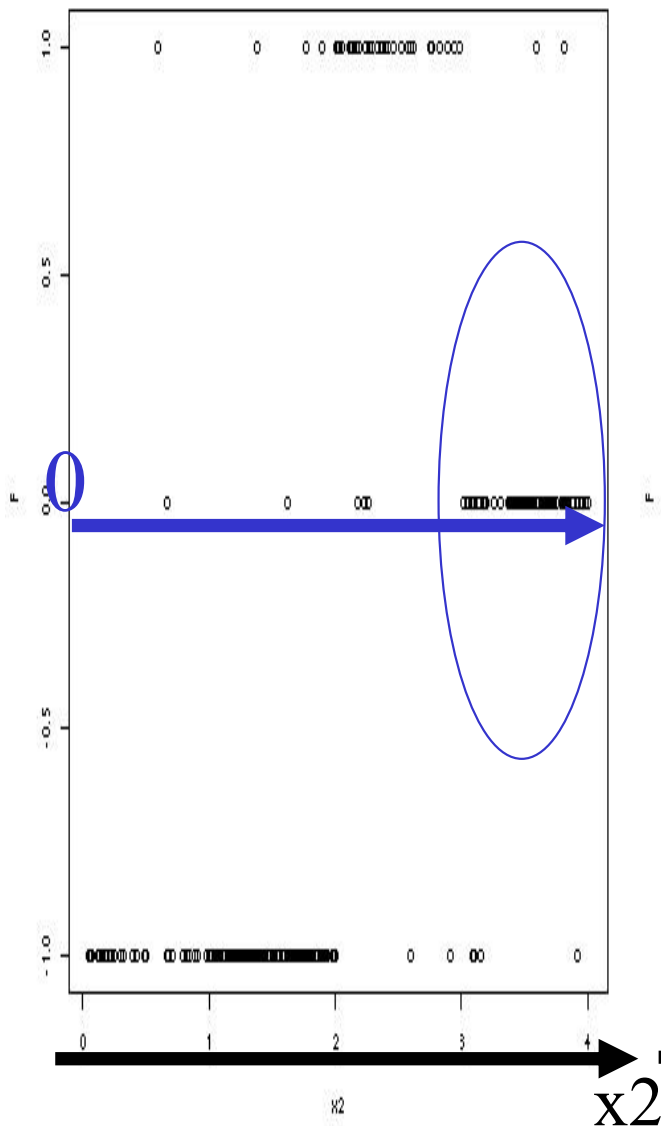
Experiment: Dataset



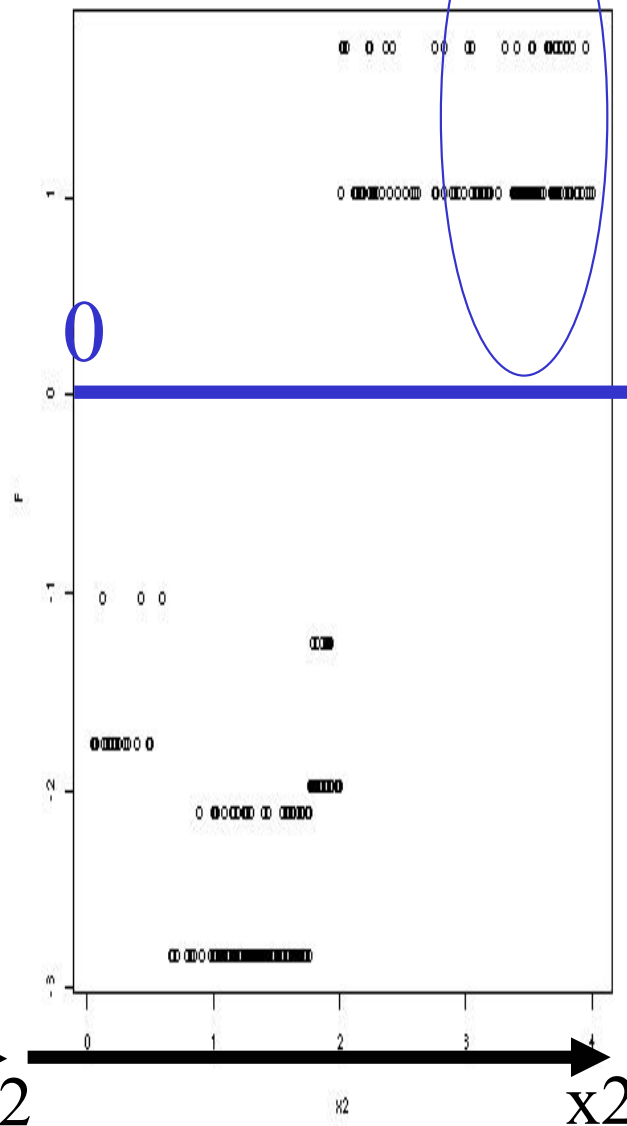
Discriminant function of Ternary-AdaBoost

-Code matrix (-1,-1,1,0)

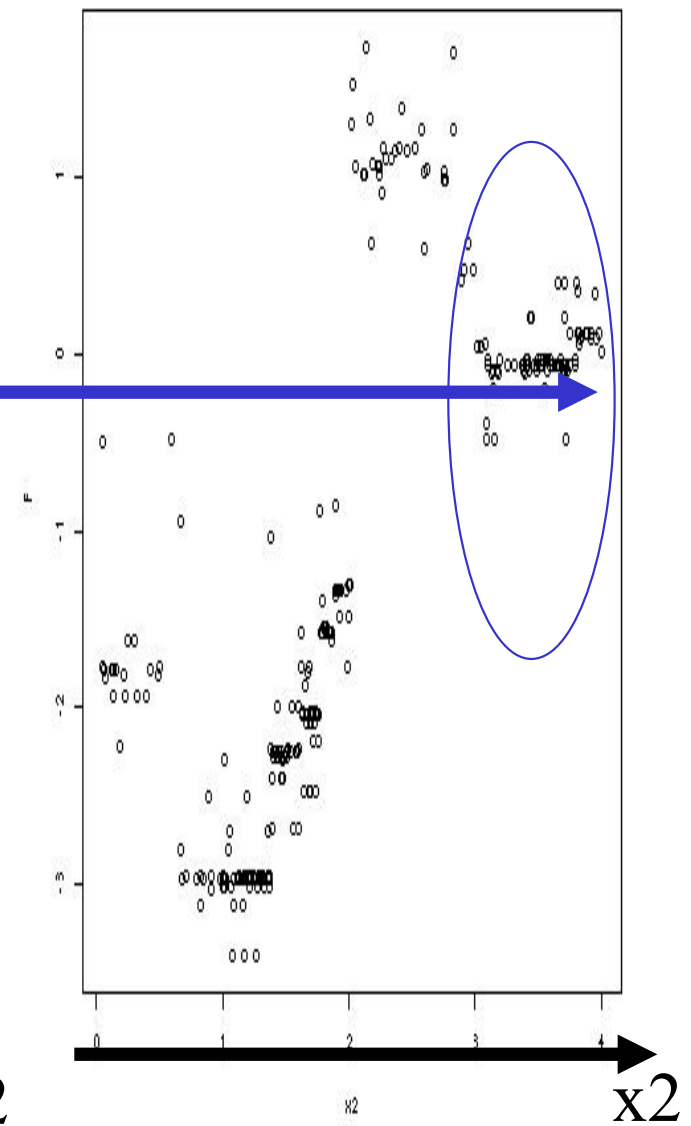
True code



AdaBoost



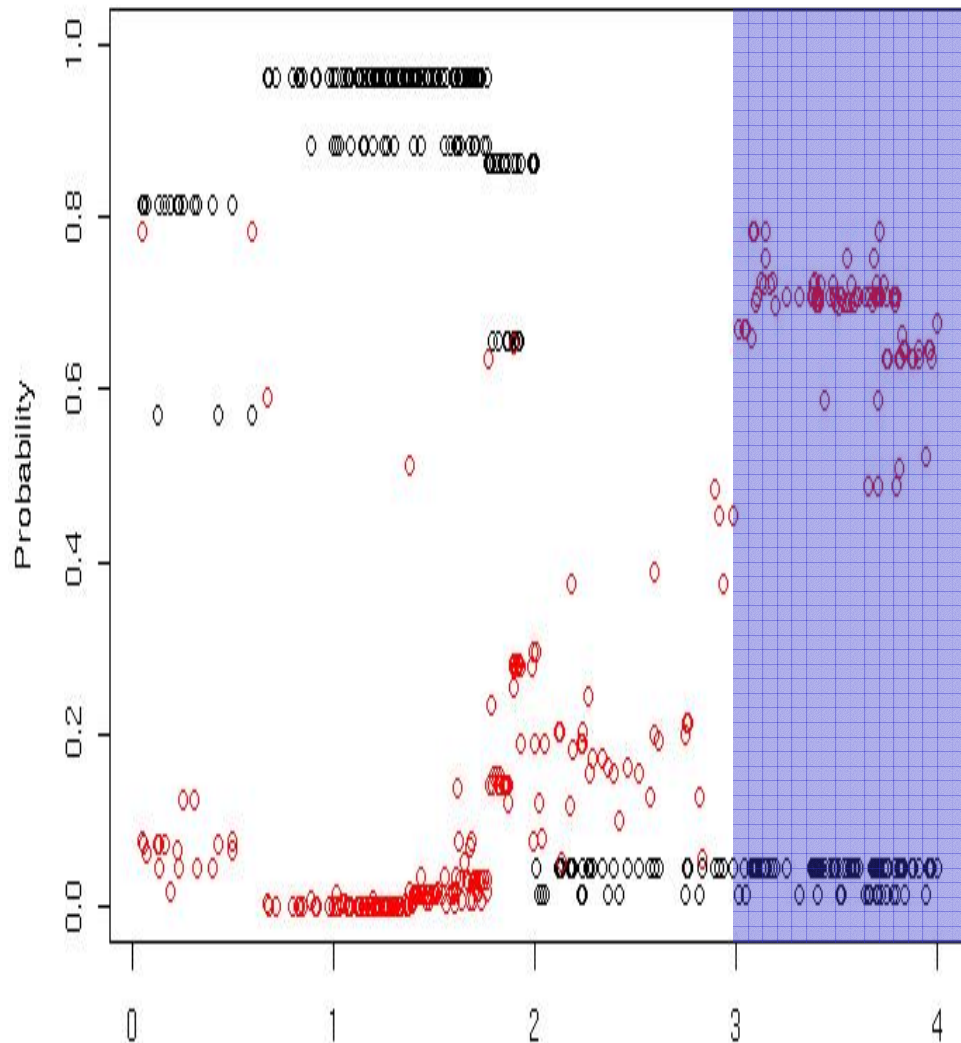
Ternary-AdaBoost



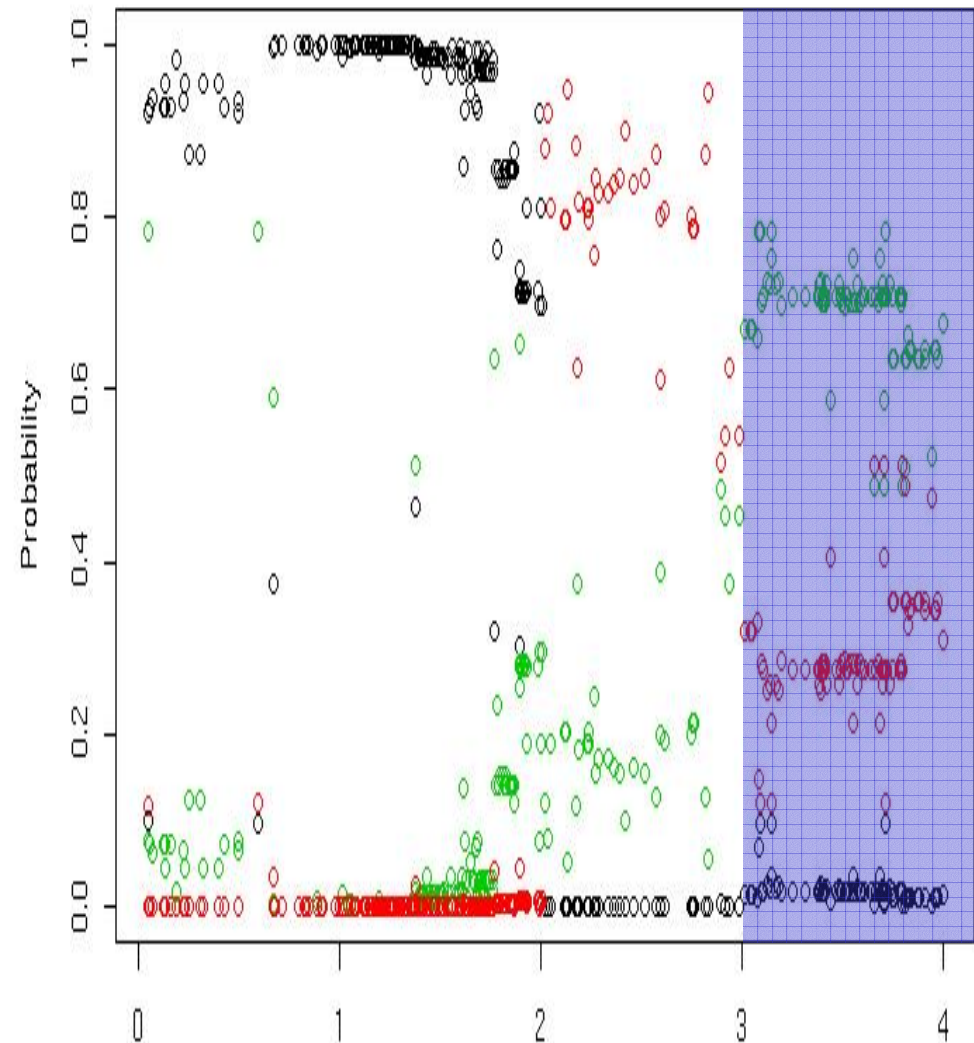
確率的ラベル予測 - Code matrix $(-1,-1,1,0)$

○ : -1, ○ : 1, ○ : 0

AdaBoost



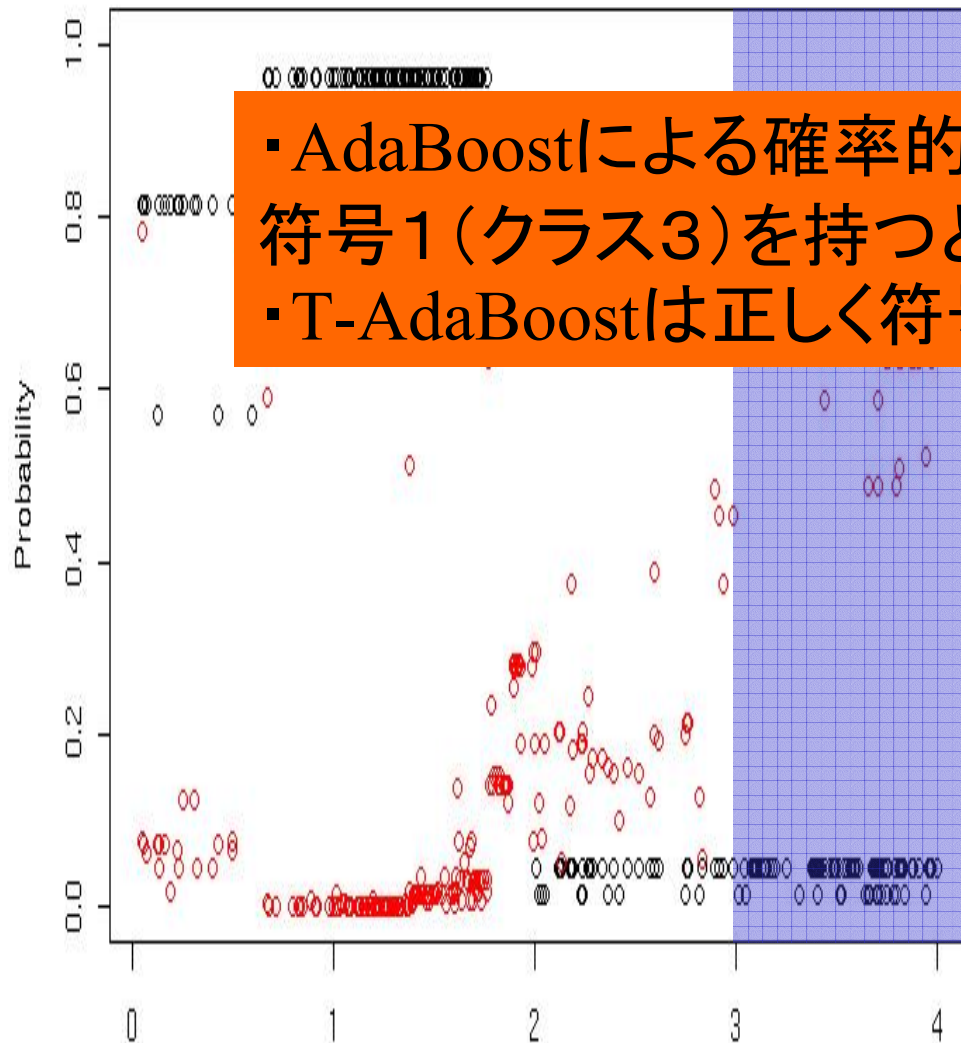
T-AdaBoost



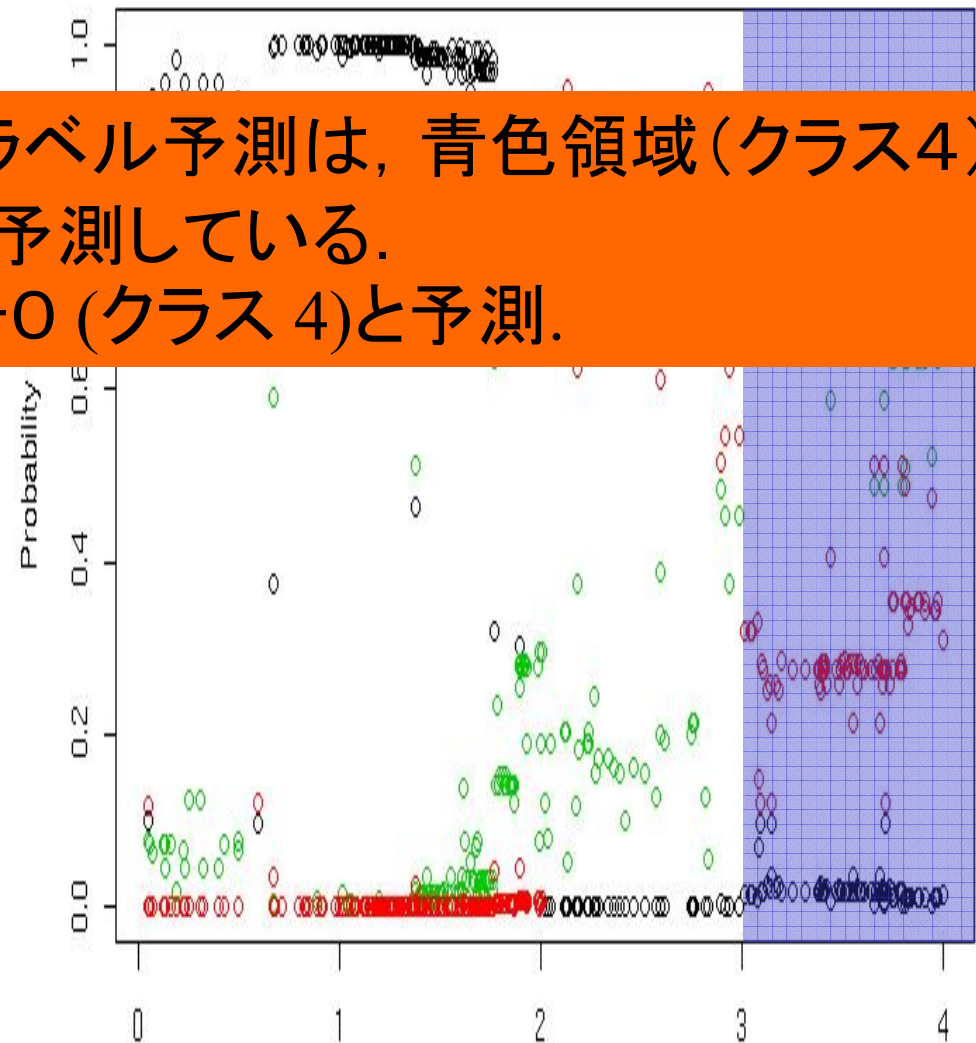
確率的ラベル予測 - Code matrix $(-1,-1,1,0)$

○ : -1, ○ : 1, ○ : 0

AdaBoost

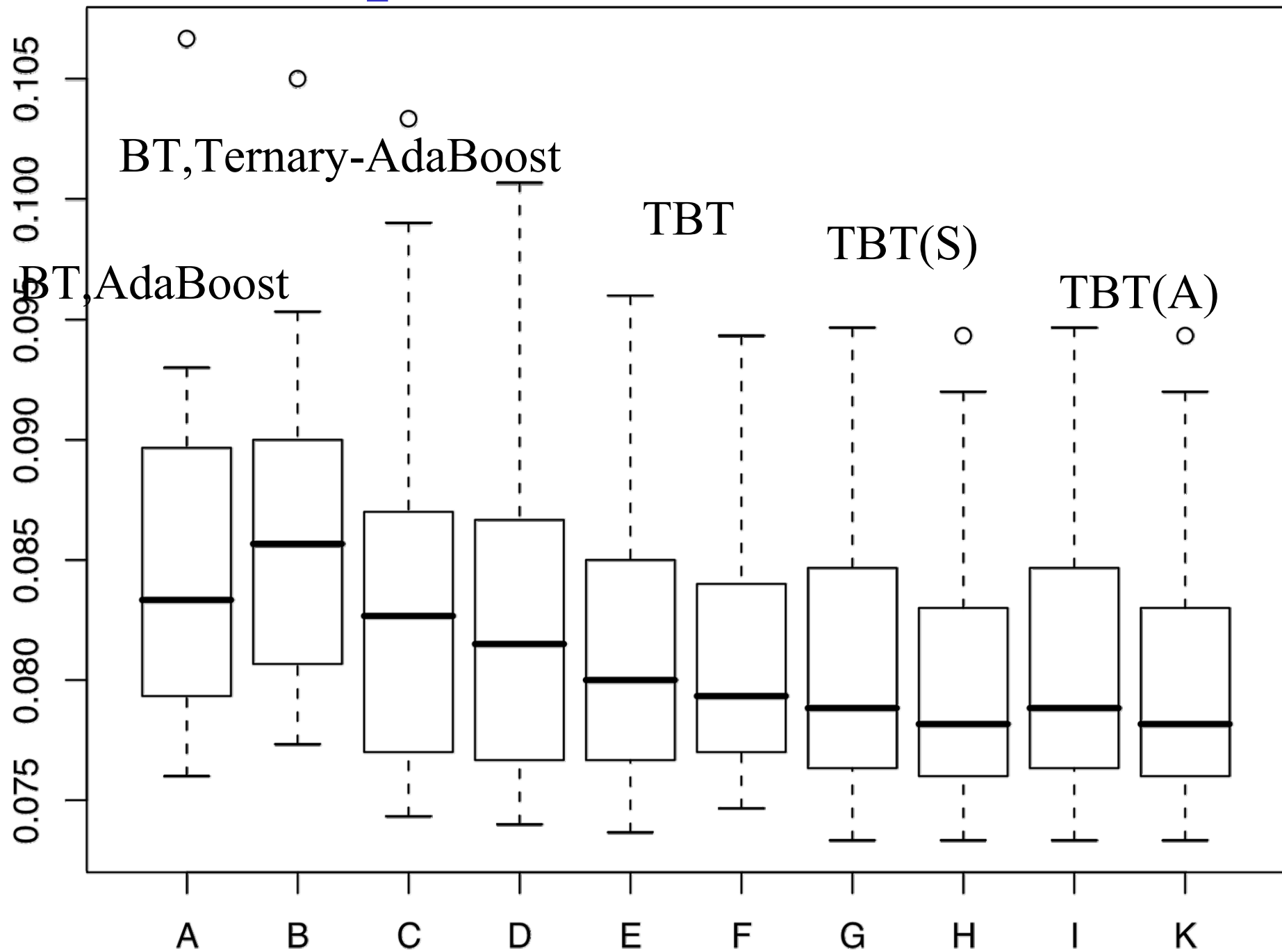


T-AdaBoost



- AdaBoostによる確率的ラベル予測は、青色領域(クラス4)が符号1(クラス3)を持つと予測している.
- T-AdaBoostは正しく符号0(クラス4)と予測.

Experiment: Test error



UCI datasets: Test error

TABLE 1: TEST AND TRAINING ERROR OF TBT ALGORITHM

Dataset	BT	TBT	TBT(S)	TBT(A)
Thyroid($w_j = n_j$)	0.569 ± 0.200	0.688 ± 0.208**	0.667 ± 0.198**	0.667 ± 0.198**
($w_j = 1$)	0.600 ± 0.208	0.838 ± 0.210**	0.800 ± 0.209**	0.800 ± 0.209**
Waveform($w_j = n_j$)	15.021 ± 1.016	14.922 ± 0.992	14.905 ± 1.011	14.905 ± 1.011
($w_j = 1$)	15.124 ± 0.993	14.976 ± 1.008	14.919 ± 1.005	14.919 ± 1.005
Iris ($w_j = n_j$)	5.400 ± 3.409	5.167 ± 3.363	5.167 ± 3.363	5.167 ± 3.363
($w_j = 1$)	5.400 ± 3.409	5.300 ± 3.452	5.300 ± 3.452	5.300 ± 3.452
Wine ($w_j = n_j$)	2.889 ± 2.790	2.972 ± 2.854	2.972 ± 2.854	2.972 ± 2.854
($w_j = 1$)	2.889 ± 2.817	3.000 ± 2.920	2.972 ± 2.881	2.972 ± 2.881
Glass ($w_j = n_j$)	27.163 ± 6.577	26.884 ± 6.395	26.884 ± 6.505	26.884 ± 6.505
($w_j = 1$)	27.488 ± 6.344	26.512 ± 6.157	26.512 ± 6.202	26.512 ± 6.202
Balance ($w_j = n_j$)	4.632 ± 1.947**	1.808 ± 1.285	2.232 ± 1.371*	2.232 ± 1.371*
($w_j = 1$)	5.144 ± 2.113**	1.968 ± 1.323	2.264 ± 1.439**	2.224 ± 1.454*
Car eva. ($w_j = n_j$)	29.298 ± 2.328**	26.942 ± 2.281**	27.379 ± 2.320**	27.379 ± 2.320**
($w_j = 1$)	31.023 ± 2.284**	24.786 ± 2.296	25.246 ± 2.254	25.243 ± 2.249
Dermatology ($w_j = n_j$)	4.000 ± 2.201**	3.431 ± 1.939	3.069 ± 1.702	3.069 ± 1.702
	3.819 ± 2.017**	3.361 ± 1.864	3.056 ± 1.743	3.056 ± 1.743

Conclusions

- Ternary Bradley-Terryモデルに基づく多値判別法の提案
 - Ternary-AdaBoost
- 2乗誤差ロスを用いた復号
 - 性能を保持したまま, 大幅な計算量の削減

全体の結論

- 複数の2値判別問題の情報を統合して多値判別を行うための手法の提案
 - 確率モデルを用いたECOC復号の拡張
 - 局所化による拡張
 - Bradley-Terryモデルに基づく手法の改良
 - 性能の向上と2乗誤差ロスの採用による高速化
- どちらを使うのがよいのか？
 - ケースバイケースだが提案手法のどちらかを使えば問題ない