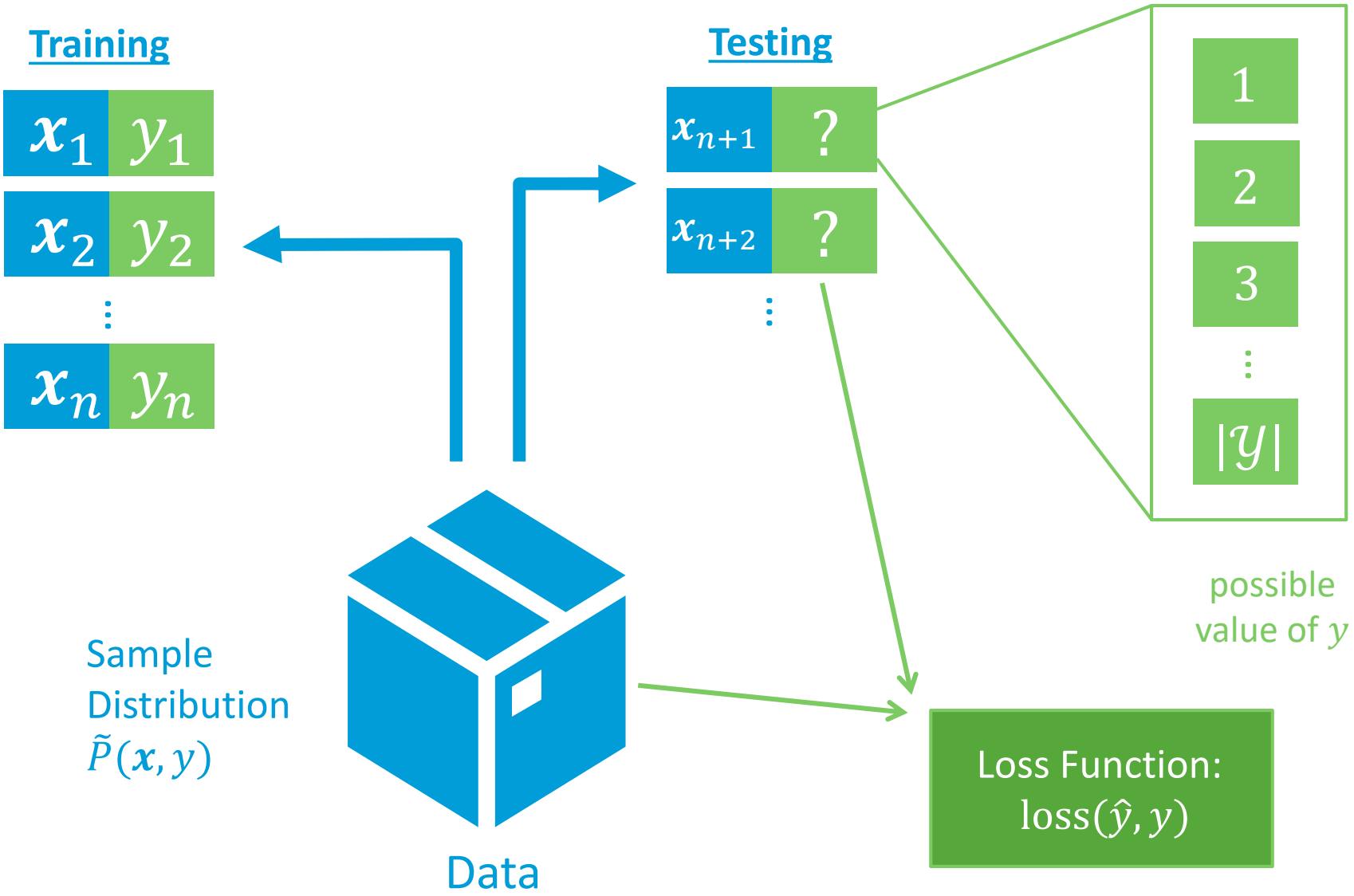
UIC COMPUTER SCIENCE

Adversarial Surrogate Losses for General Multiclass Classification

Rizal Zaini Ahmad Fathony

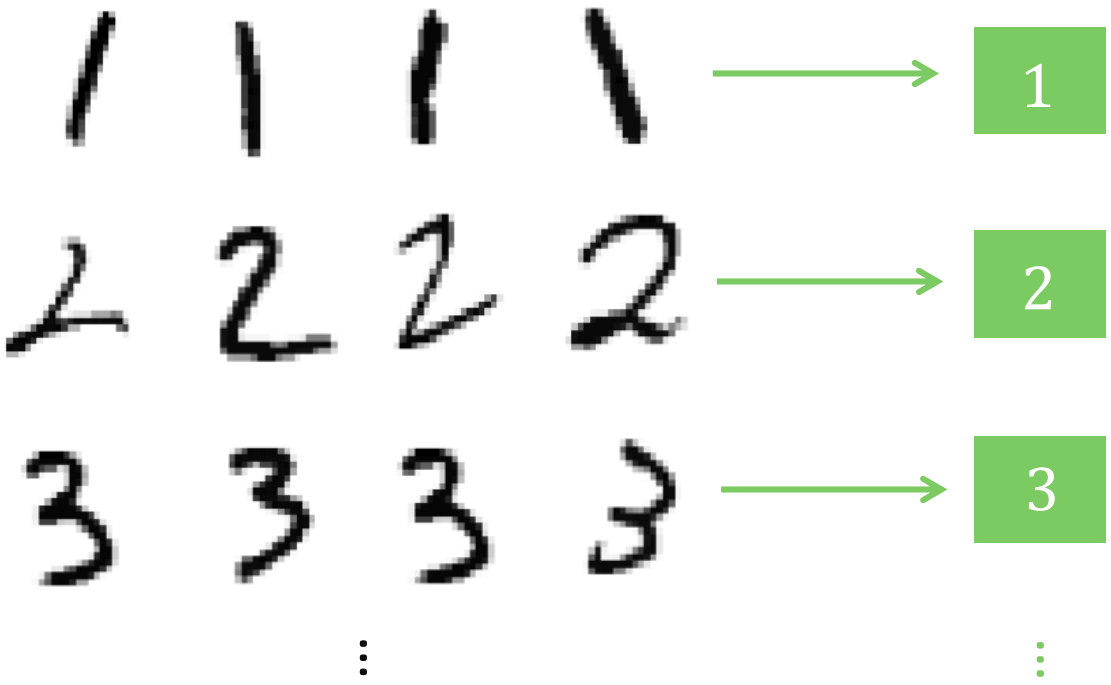
Committee: Prof. Brian Ziebart (Chair)
Prof. Bhaskar DasGupta
Prof. Lev Reyzin
Prof. Xinhua Zhang
Prof. Simon Lacoste-Julien

Supervised Learning → Multiclass Classification



Multiclass Classification → Zero-One Loss

Digit Recognition



Loss Function:

$$\text{loss}(\hat{y}, y) = I(\hat{y} \neq y)$$

General Multiclass Classification → any loss

Multiclass Classification → Ordinal Classification

Movie Rating Prediction

→ 1 ★☆☆☆☆

→ 2 ★★☆☆☆

⋮

→ 5 ★★★★★

Predicted vs Actual Label:

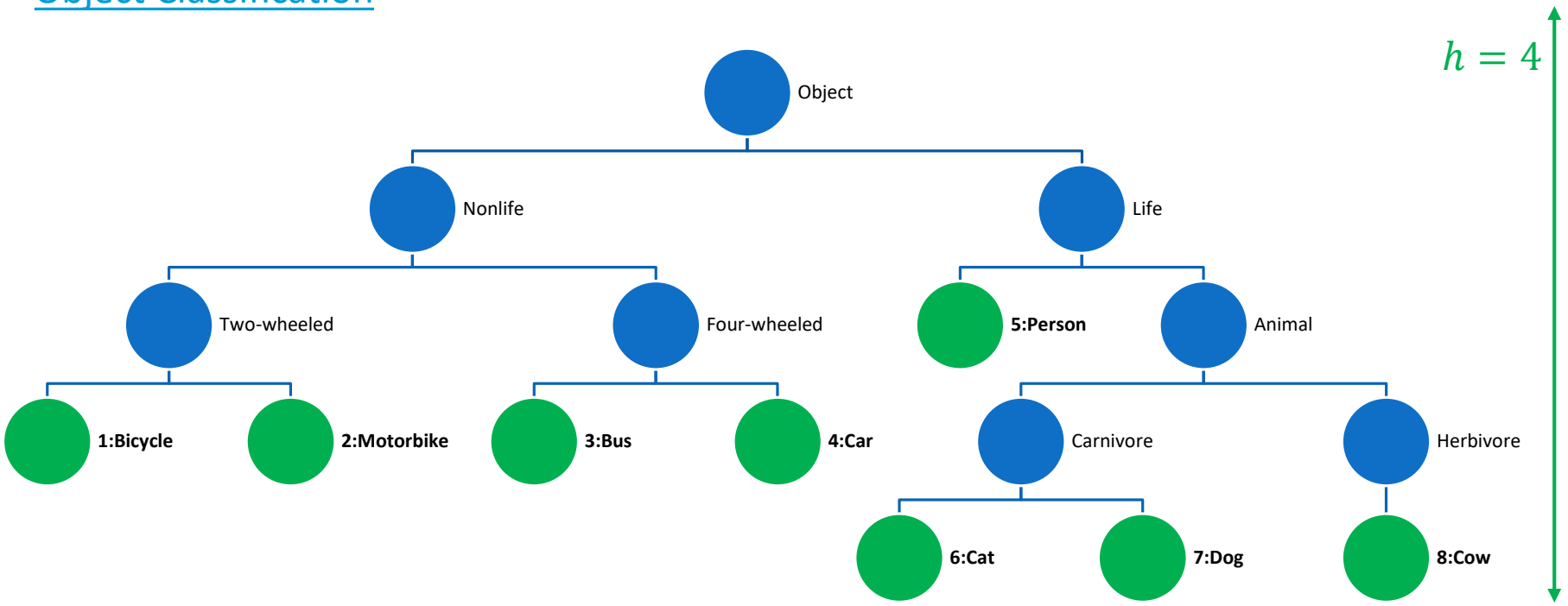
Distance ↑ → Loss ↑

Loss Function (example):

$$\text{loss}(\hat{y}, y) = |\hat{y} - y|$$

Multiclass Classification → Taxonomy Classification

Object Classification



Loss Function (example):

$$\text{loss}(\hat{y}, y) = h - v(\hat{y}, y) + 1$$

h : tree height

$v(\hat{y}, y)$: level of the common ancestor

loss(Cat,Dog) = 1
loss(Cat,Cow) = 2
loss(Cow,Person) = 3
Loss(Cow,Motorbike)= 4

loss(Bus,Car) = 2
loss(Bus,Bicycle) = 3
loss(Car,Cow) = 4
loss(Bus,Person) = 4

Multiclass Classification → Loss Matrix

loss function: $\text{loss}(\hat{y}, y)$ → loss matrix: $\boldsymbol{L}$

Zero One Loss

$$\text{loss}(\hat{y}, y) = I(\hat{y} \neq y)$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

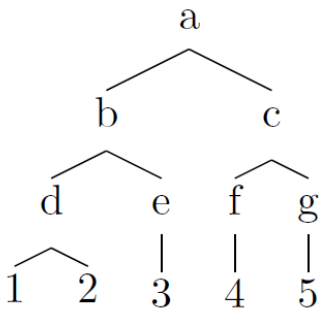
Ordinal Classification Loss

$$\text{loss}(\hat{y}, y) = |\hat{y} - y|$$

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

Taxonomy-based loss

$$\text{loss}(\hat{y}, y) = h - v(\hat{y}, y) + 1$$



$$\begin{bmatrix} 0 & 1 & 2 & 3 & 3 \\ 1 & 0 & 2 & 3 & 3 \\ 2 & 2 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 2 \\ 3 & 3 & 3 & 2 & 0 \end{bmatrix}$$

Empirical Risk Minimization (ERM)

- Assume a family of parametric hypothesis function f (e.g. linear discriminator)
- Find the hypothesis f^* that minimize the empirical risk:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \text{loss}(f(\mathbf{x}_i), y_i) \quad = \min_f \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{loss}(f(\mathbf{X}), Y)\right]$$

Intractable optimization, non-convex, non-continuous

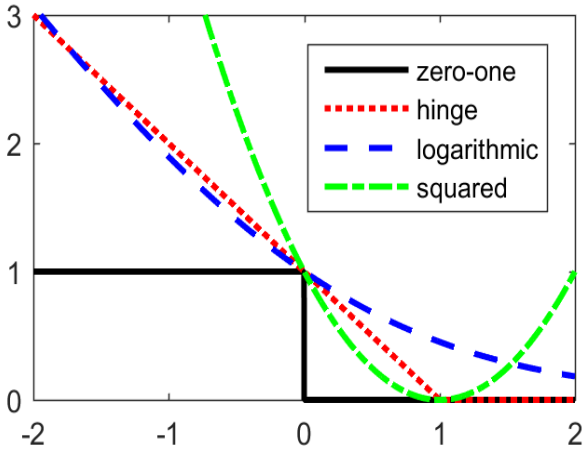
Convex surrogate loss need to be employed

Example:

Binary zero-one loss

Surrogate Loss:

- Hinge loss (used by SVM)
- Log loss (used by Logistic Regression)
- Exponential loss (used by AdaBoost)



ERM under Hinge Loss and Log Loss

SVM (hinge loss):

$$\min_f \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{HingeLoss}(f(\mathbf{X}), Y)\right]$$

Logistic regression (log loss):

Probabilistic prediction $\hat{P}_f(y|\boldsymbol{x})$

$$\min_f \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{LogLoss}(\hat{P}_f(Y|\mathbf{X}), Y)\right]$$

Binary SVM and Binary Logistic Regression:

Fisher consistent: produce Bayes optimal decision in the limit

Binary SVM only:

Dual parameter sparsity

Surrogate loss for multiclass cases:

Extend binary surrogate loss like hinge-loss and log-loss to multiclass

Adversarial Prediction (Asif et. al., 2015)

Original Loss

Non-convex, non-continuous

$$\min_f \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{loss}(f(\mathbf{X}), Y)\right]$$

Approximate the loss

with convex surrogates

Empirical Risk Minimization

$$\min_f \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{surrogate}(f(\mathbf{X}), Y)\right]$$

$$\hat{y} = \operatorname*{argmax}_j f_j(\mathbf{x})$$

Probabilistic prediction

$$\hat{P}(\hat{y}|\mathbf{x})$$

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\tilde{P}(\tilde{y}|\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})} \left[\text{loss}(\hat{Y}, \tilde{Y})\right]$$

Evaluate against an adversary,
instead of using empirical data

Adversary's probabilistic prediction

$$\check{P}(\check{y}|\mathbf{x})$$

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \max_{\check{P}(\check{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})\check{P}(\check{y}|\mathbf{x})} \left[\text{loss}(\hat{Y}, \check{Y})\right]$$

Adversarial Prediction

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \max_{\check{P}(\check{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})\check{P}(\check{y}|\mathbf{x})} \left[\text{loss}(\hat{Y}, \check{Y})\right]$$

$$\text{s.t. } \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X}, \check{Y})] = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$$

Constraint the **statistics** of
the **adversary**'s distribution
to match the **empirical** statistics

Empirical Risk Minimization

Approximate loss

Exact training data

Adversarial Prediction

Exact loss

Approximate training data

(by only using the statistics)

Adversarial Prediction → Optimization

Adversarial Prediction

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \max_{\check{P}(\check{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})\check{P}(\check{y}|\mathbf{x})} \left[\text{loss}(\hat{Y}, \check{Y})\right]$$

$$\text{s.t. } \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X}, \check{Y})] = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$$

Minimax and Lagrangian duality

Minimization over many zero-sum games

$$\min_{\theta} \mathbb{E}_{P(\mathbf{x})} \left[\max_{\check{\mathbf{p}}_\mathbf{X}} \min_{\hat{\mathbf{p}}_\mathbf{X}} \hat{\mathbf{p}}_\mathbf{X}^\mathrm{T} \mathbf{L}'_{\mathbf{X},\theta} \check{\mathbf{p}}_\mathbf{X}\right]$$

where:

$$(\mathbf{L}'_{\mathbf{x},\theta})_{\hat{y},\check{y}} = \text{loss}(\hat{y}, \check{y}) + \theta^\mathrm{T}(\phi(\mathbf{x}, \check{y}) - \phi(\mathbf{x}, \tilde{y}))$$

$$\hat{\mathbf{p}}_\mathbf{x} = [\hat{P}(\hat{Y} = 1|\mathbf{x}) \ \hat{P}(\hat{Y} = 2|\mathbf{x}) \ \ldots]^\mathrm{T}$$

$$\check{\mathbf{p}}_\mathbf{x} = [\check{P}(\check{Y} = 1|\mathbf{x}) \ \check{P}(\check{Y} = 2|\mathbf{x}) \ \ldots]^\mathrm{T}$$

Inner optimization

Can be solved using Linear Programming

Complexity: $O(|\mathcal{Y}|^{3.5})$

Example of game matrix $\mathbf{L}'_{\mathbf{x},\theta}$ for zero-one loss

$$\begin{bmatrix} \psi_{1,y}(\mathbf{x}) & \psi_{2,y}(\mathbf{x}) + 1 & \psi_{3,y}(\mathbf{x}) + 1 & \psi_{4,y}(\mathbf{x}) + 1 \\ \psi_{1,y}(\mathbf{x}) + 1 & \psi_{2,y}(\mathbf{x}) & \psi_{3,y}(\mathbf{x}) + 1 & \psi_{4,y}(\mathbf{x}) + 1 \\ \psi_{1,y}(\mathbf{x}) + 1 & \psi_{2,y}(\mathbf{x}) + 1 & \psi_{3,y}(\mathbf{x}) & \psi_{4,y}(\mathbf{x}) + 1 \\ \psi_{1,y}(\mathbf{x}) + 1 & \psi_{2,y}(\mathbf{x}) + 1 & \psi_{3,y}(\mathbf{x}) + 1 & \psi_{4,y}(\mathbf{x}) \end{bmatrix}$$

$$\psi_{j,\tilde{y}}(\mathbf{x}) = f_j(\mathbf{x}) - f_{\tilde{y}}(\mathbf{x}) = \theta^\mathrm{T}(\phi(\mathbf{x}, j) - \phi(\mathbf{x}, \tilde{y}))$$

Adversarial Prediction → ERM perspective

Adversarial Prediction (optimization)

$$\min_{\theta} \mathbb{E}_{P(\mathbf{x})} \left[\max_{\check{\mathbf{p}}_{\mathbf{X}}} \min_{\hat{\mathbf{p}}_{\mathbf{X}}} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}} \mathbf{L}'_{\mathbf{X},\theta} \check{\mathbf{p}}_{\mathbf{X}} \right]$$

Empirical Risk Minimization with surrogate loss:

$$\mathrm{AdversarialSurrogate}(f_\theta(\mathbf{x}), y) = \max_{\check{\mathbf{p}}_{\mathbf{x}}} \min_{\hat{\mathbf{p}}_{\mathbf{x}}} \hat{\mathbf{p}}_{\mathbf{x}}^{\mathrm{T}} \mathbf{L}'_{\mathbf{x},\theta} \check{\mathbf{p}}_{\mathbf{x}}$$

where: $(\mathbf{L}'_{\mathbf{x},\theta})_{\hat{y},\check{y}} = \mathrm{loss}(\hat{y}, \check{y}) + (f_\theta)_{\check{y}}(\mathbf{x}) - (f_\theta)_{(\tilde{y})}(\mathbf{x})$

Adversarial Surrogate Loss

=

The **Nash equilibrium** value of
the **zero-sum game** characterized by matrix $\boldsymbol{L}'_{\boldsymbol{x},\theta}$

Outline

1. The Adversarial Surrogate Loss for Multiclass Zero-One Classification

2. Adversarial Surrogate Losses for Multiclass Ordinal Classification

3. Ongoing and Future Works

The Adversarial Surrogate Loss for Multiclass Zero-One Classification

Based on:

Rizal Fathony, Anqi Liu, Kaiser Asif, Brian D. Ziebart. "*Adversarial Multiclass Classification: A Risk Minimization Perspective*". Advances in Neural Information Processing Systems 29 (NIPS), 2016.

Multiclass Zero-One: Related Works

Multiclass Support Vector Machine

	Fisher Consistent? (Tewari and Bartlett, 2007) (Liu, 2007)	**Perform well** in low feature spaces? (Dogan, 2016)
1. The WW Model (Weston et.al., 2002)	✗	✓
2. The CS Model (Crammer and Singer, 1999)	✗	✓
3. The LLW Model (Lee et.al., 2004)	✓	✗

1. The WW Model (Weston et.al., 2002)

$$\text{loss}_{\text{WW}}(\mathbf{x}_i, y_i) = \sum_{j \neq y_i} [1 - (f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i))]_+$$

Relative Margin Model

2. The CS Model (Crammer and Singer, 1999)

$$\text{loss}_{\text{CS}}(\mathbf{x}_i, y_i) = \max_{j \neq y_i} [1 - (f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i))]_+$$

Relative Margin Model

3. The LLW Model (Lee et.al., 2004)

$$\text{loss}_{\text{LLW}}(\mathbf{x}_i, y_i) = \sum_{j \neq y_i} [1 + f_j(\mathbf{x}_i)]_+$$

with: $\sum_j f_j(\mathbf{x}_i) = 0$

Absolute Margin Model

Adversarial Prediction : Multiclass Zero-One Loss

Adversarial Game

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \max_{\check{P}(\check{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})\check{P}(\check{y}|\mathbf{x})} \left[I(\hat{Y} \neq \check{Y}) \right]$$

$$\text{s.t. } \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X}, \check{Y})] = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$$

$$\min_{\theta} \mathbb{E}_{P(\mathbf{x})} \left[\max_{\check{\mathbf{p}}_{\mathbf{X}}} \min_{\hat{\mathbf{p}}_{\mathbf{X}}} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}} \mathbf{L}'_{\mathbf{X},\theta} \check{\mathbf{p}}_{\mathbf{X}} \right]$$

$$\mathbf{L}'_{\mathbf{x},\theta} = \begin{bmatrix} \psi_{1,\tilde{y}}(\mathbf{x}) & \psi_{2,\tilde{y}}(\mathbf{x}) + 1 & \psi_{3,\tilde{y}}(\mathbf{x}) + 1 & \psi_{4,\tilde{y}}(\mathbf{x}) + 1 \\ \psi_{1,\tilde{y}}(\mathbf{x}) + 1 & \psi_{2,\tilde{y}}(\mathbf{x}) & \psi_{3,\tilde{y}}(\mathbf{x}) + 1 & \psi_{4,\tilde{y}}(\mathbf{x}) + 1 \\ \psi_{1,\tilde{y}}(\mathbf{x}) + 1 & \psi_{2,\tilde{y}}(\mathbf{x}) + 1 & \psi_{3,\tilde{y}}(\mathbf{x}) & \psi_{4,\tilde{y}}(\mathbf{x}) + 1 \\ \psi_{1,\tilde{y}}(\mathbf{x}) + 1 & \psi_{2,\tilde{y}}(\mathbf{x}) + 1 & \psi_{3,\tilde{y}}(\mathbf{x}) + 1 & \psi_{4,\tilde{y}}(\mathbf{x}) \end{bmatrix}$$

$$\psi_{j,\tilde{y}}(\mathbf{x}) = f_j(\mathbf{x}) - f_{\tilde{y}}(\mathbf{x}) = \theta^{\mathrm{T}} \left(\phi(\mathbf{x}, j) - \phi(\mathbf{x}, \tilde{y}) \right)$$

shorter notation

$$\mathbf{L}'_{\mathbf{x},\theta} = \begin{bmatrix} \psi_1 & \psi_2 + 1 & \psi_3 + 1 & \psi_4 + 1 \\ \psi_1 + 1 & \psi_2 & \psi_3 + 1 & \psi_4 + 1 \\ \psi_1 + 1 & \psi_2 + 1 & \psi_3 & \psi_4 + 1 \\ \psi_1 + 1 & \psi_2 + 1 & \psi_3 + 1 & \psi_4 \end{bmatrix}$$

Nash Equilibrium

Adversarial Zero-Sum Game (Zero-One Loss)

The augmented game for 4 classes

$$\begin{array}{c} \\ \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{array} \begin{array}{c} \begin{array}{cccc} \check{p}_1 & \check{p}_2 & \check{p}_3 & \check{p}_4 \end{array} \\ \begin{bmatrix} \psi_1 & \psi_2+1 & \psi_3+1 & \psi_4+1 \\ \psi_1+1 & \psi_2 & \psi_3+1 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_3 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_3+1 & \psi_4 \end{bmatrix} \end{array}$$

if completely mixed

$$v = \frac{\sum_{j=1}^{|\mathcal{Y}|} \psi_j + |\mathcal{Y}| - 1}{|\mathcal{Y}|} = \frac{\psi_1 + \psi_2 + \psi_3 + \psi_4 + 3}{4}$$

when $\check{p}_3 = 0$

$$\begin{array}{c} \\ \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{array} \begin{array}{c} \begin{array}{cccc} \check{p}_1 & \check{p}_2 & 0 & \check{p}_4 \end{array} \\ \begin{bmatrix} \psi_1 & \psi_2+1 & \psi_3+1 & \psi_4+1 \\ \psi_1+1 & \psi_2 & \psi_3+1 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_3 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_3+1 & \psi_4 \end{bmatrix} \end{array}$$

Considering all possible set of adversary's non-zero probability:

$$v = \mathrm{AL}_{\mathbf{f}}^{0\text{-}1}(\mathbf{x}, y) = \max_{\substack{\mathcal{S} \subseteq \{1,\ldots,|\mathcal{Y}|\} \\ \mathcal{S} \neq \emptyset}} \frac{\sum_{j \in \mathcal{S}} \psi_{j,y}(\mathbf{x}) + |\mathcal{S}| - 1}{|\mathcal{S}|}$$

$\mathrm{AL}^{0\text{-}1}$: maximization over $2^{|\mathcal{Y}|} - 1$ hyperplanes

$$\begin{array}{c} \\ \hat{p}_1 \\ \hat{p}_2 \\ 0 \\ \hat{p}_4 \end{array} \begin{array}{c} \begin{array}{ccc} \check{p}_1 & \check{p}_2 & \check{p}_4 \end{array} \\ \begin{bmatrix} \psi_1 & \psi_2+1 & \psi_4+1 \\ \psi_1+1 & \psi_2 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_4 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_4 \end{array} \begin{array}{c} \begin{array}{ccc} \check{p}_1 & \check{p}_2 & \check{p}_4 \end{array} \\ \begin{bmatrix} \psi_1 & \psi_2+1 & \psi_4+1 \\ \psi_1+1 & \psi_2 & \psi_4+1 \\ \psi_1+1 & \psi_2+1 & \psi_4 \end{bmatrix} \end{array}$$

if completely mixed

$$v = \frac{\psi_1 + \psi_2 + \psi_4 + 2}{3}$$

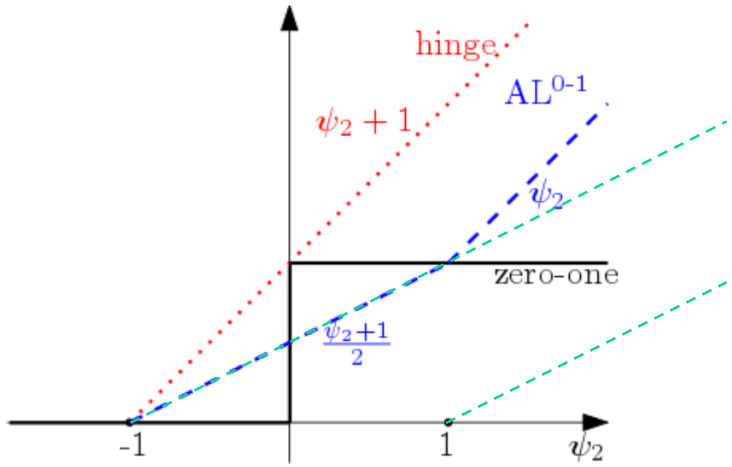
$AL^{0\text{-}1}$ (Adversarial Surrogate Loss) → Binary Classification

$AL^{0\text{-}1}$ for binary zero-one classification:

$$\mathrm{AL}_{\mathbf{f}}^{0\text{-}1}(\mathbf{x}, y) = \max\left\{\psi_{1,y}(\mathbf{x}), \psi_{2,y}(\mathbf{x}), \frac{\psi_{1,y}(\mathbf{x}) + \psi_{2,y}(\mathbf{x}) + 1}{2}\right\}$$

If the true label $y = 1$

$$\mathrm{AL}_{\mathbf{f}}^{0\text{-}1}(\mathbf{x}, y = 1) = \max\left\{0, \psi_2(\mathbf{x}), \frac{\psi_2(\mathbf{x}) + 1}{2}\right\}$$



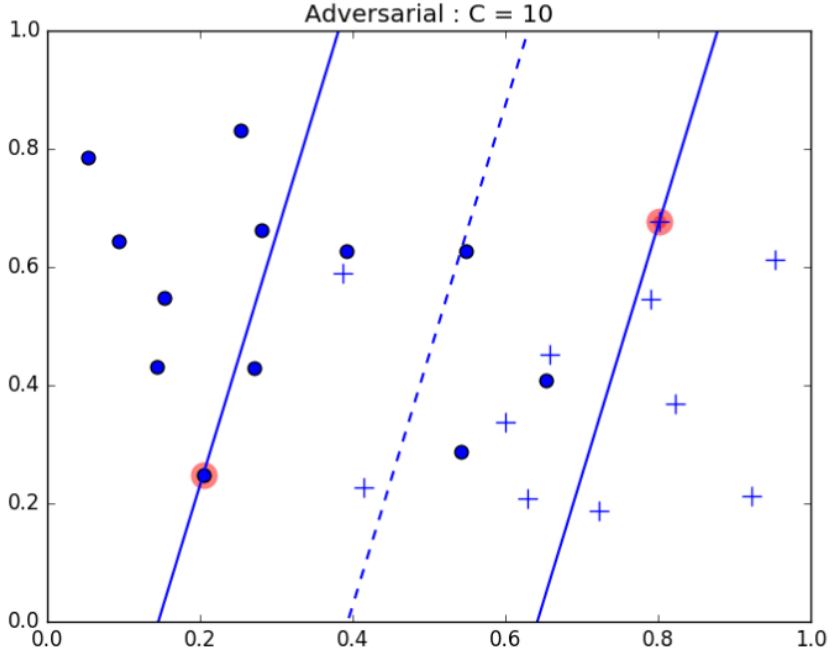
Change classification notation to $y \in \{+1, -1\}$, parameter to $\boldsymbol{w}$ and b, add L2 regularization

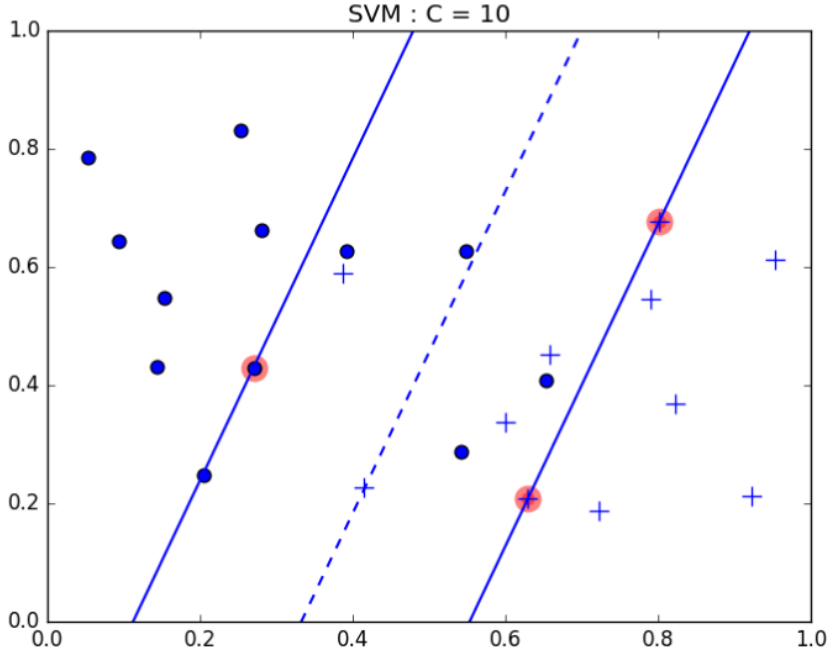
$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\delta}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\xi_i + \delta_i) \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq -1 - \delta_i \\ & \xi_i \geq 0,\ \delta_i \geq 0,\ i \in \{1,\ldots,n\} \end{aligned}$$

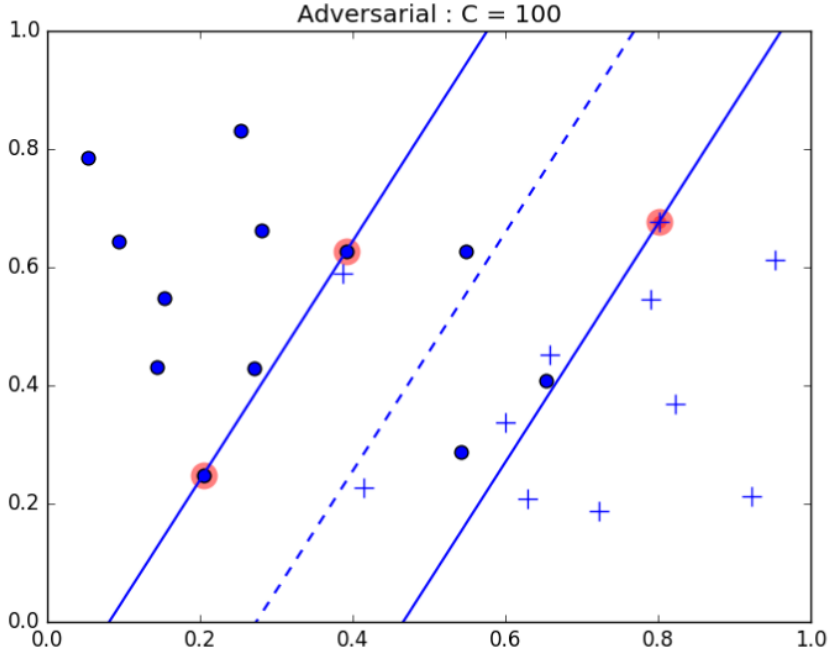
Binary $AL^{0\text{-}1}$

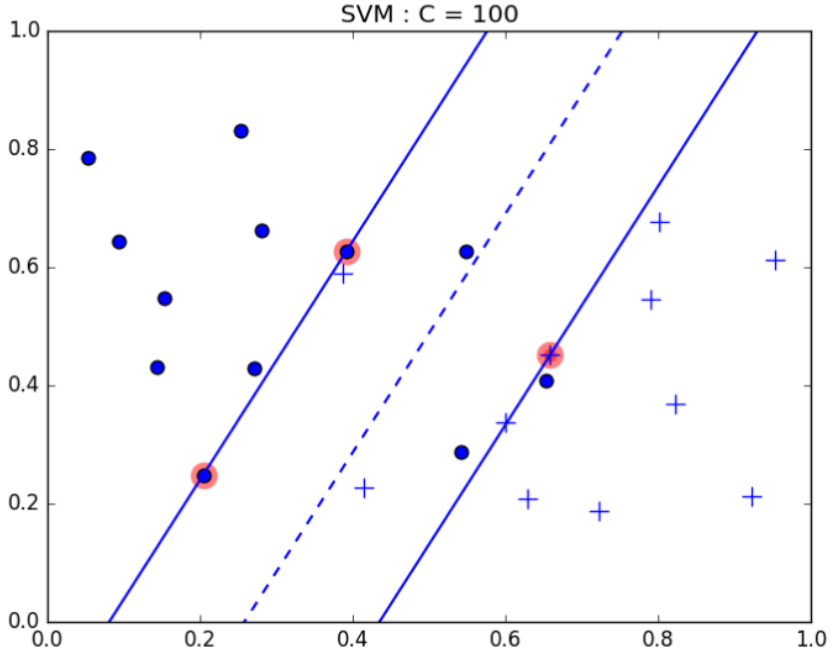
$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0,\ i \in \{1,\ldots,n\} \end{aligned}$$

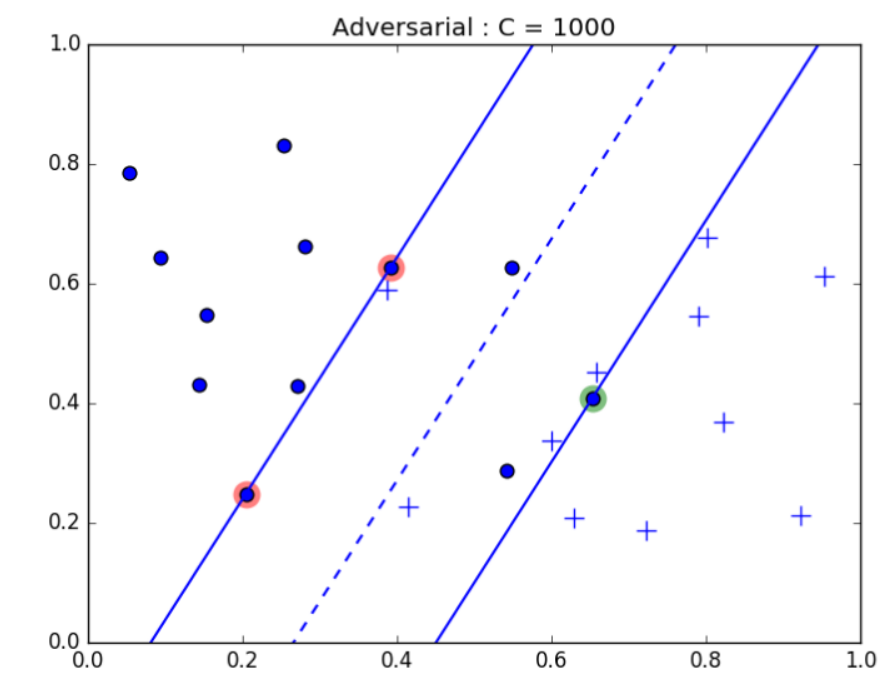
Soft Margin SVM

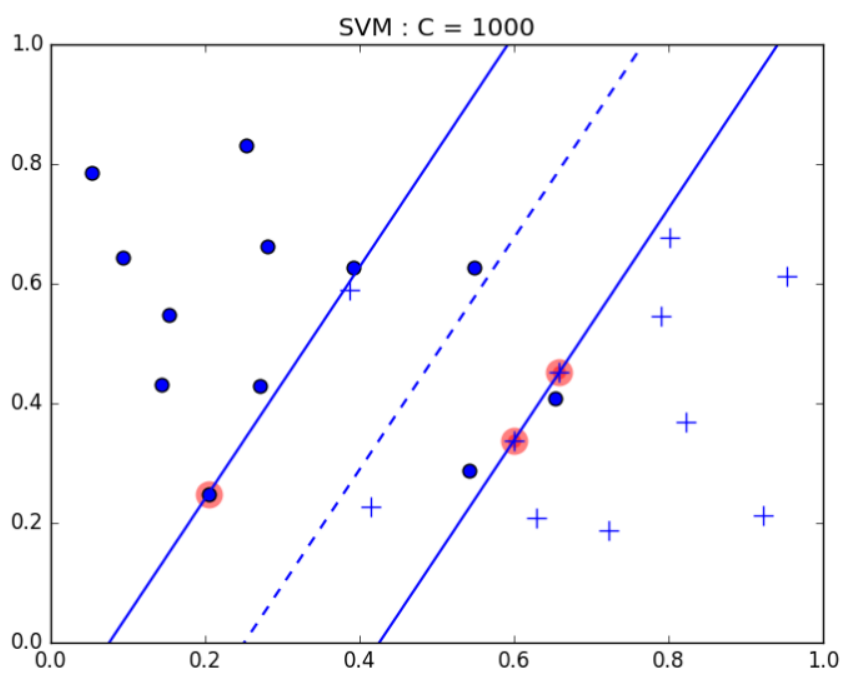

Adversarial : C = 10
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

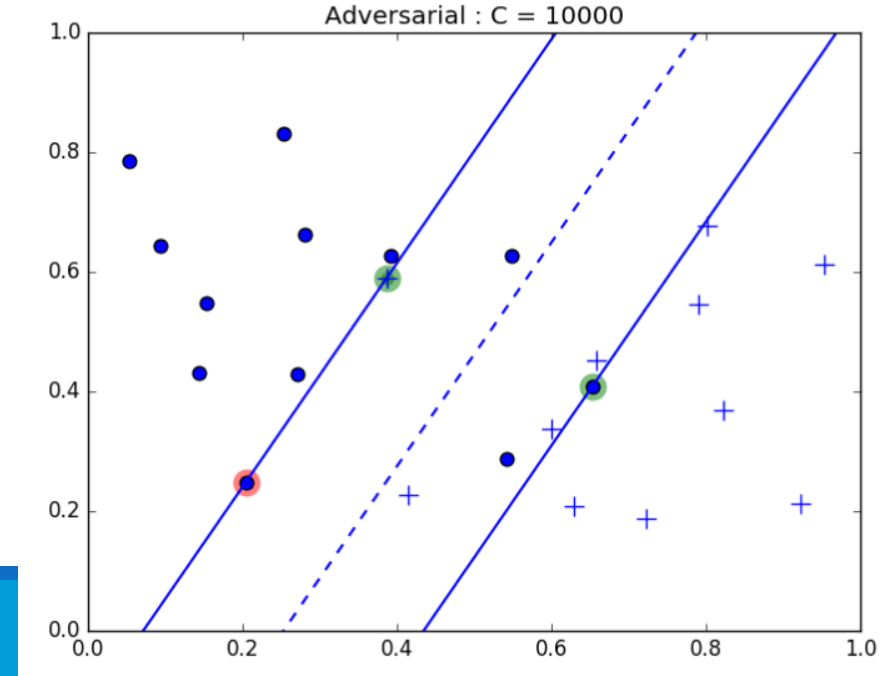

SVM : C = 10
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

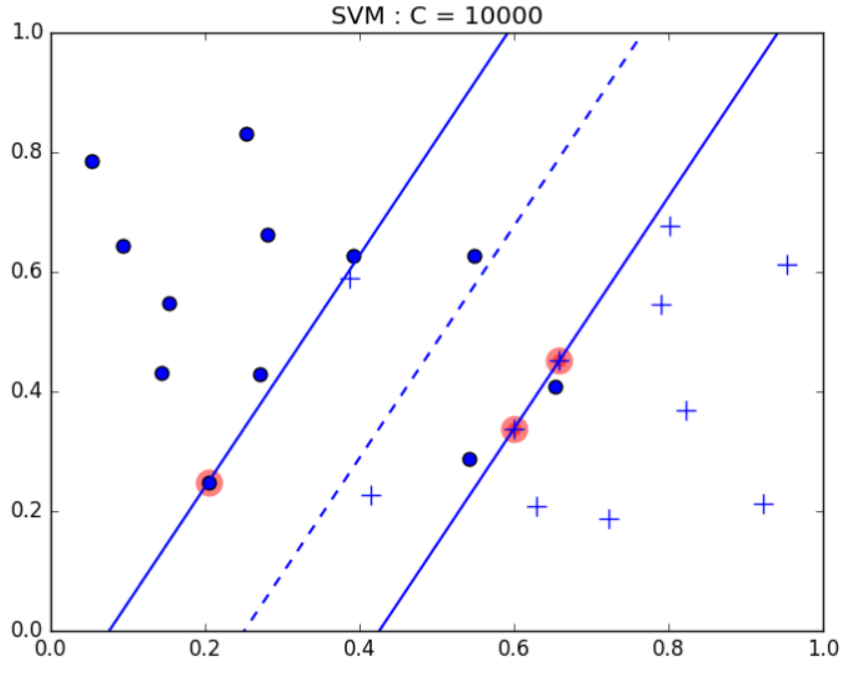

Adversarial : C = 100
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


SVM : C = 100
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Adversarial : C = 1000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


SVM : C = 1000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

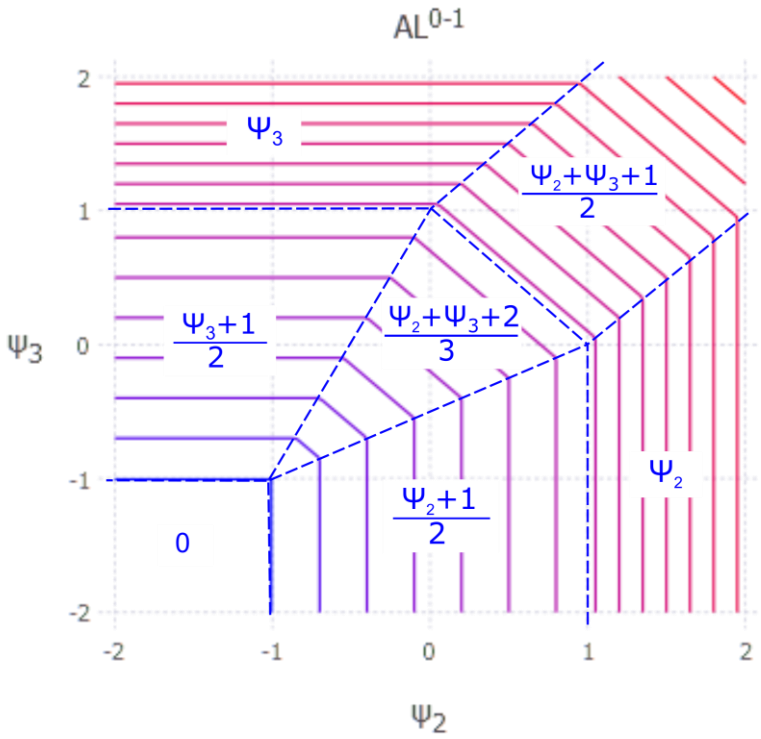

Adversarial : C = 10000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


SVM : C = 10000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

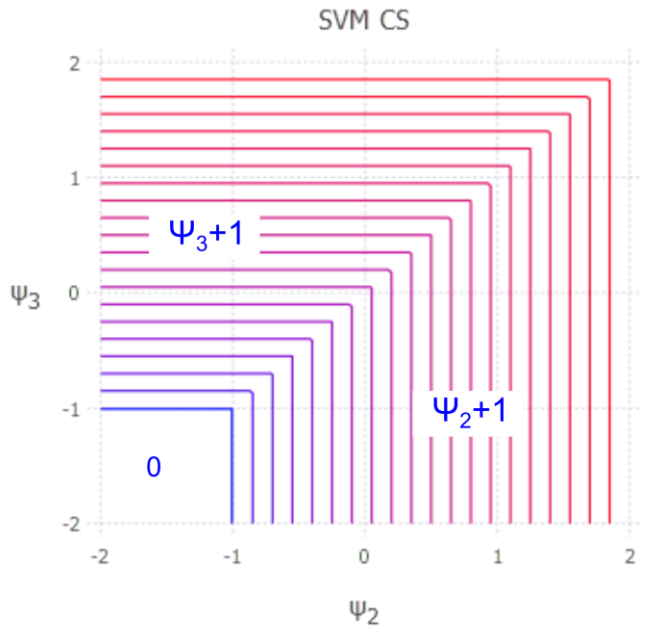
AL^{0-1} → 3 Class Classification

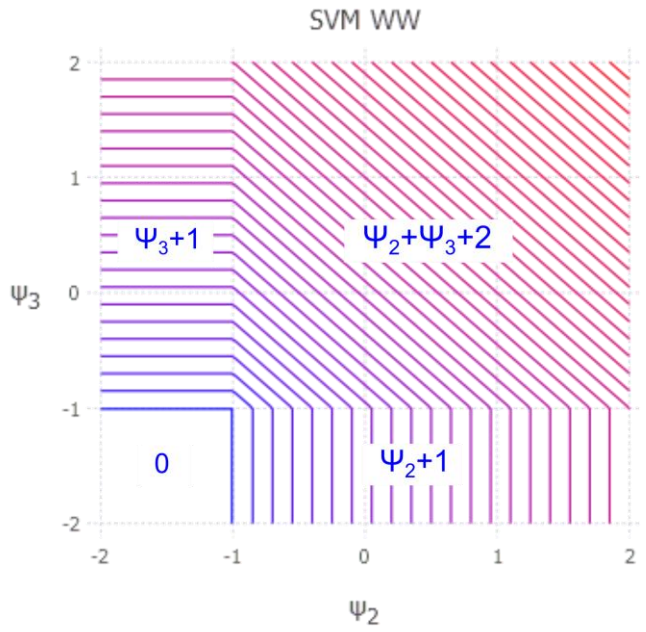
AL^{0-1} for 3-class zero-one classification:

Maximization over 7 hyperplanes:



$y = 1$





$AL^{0\text{-}1}$ → Fisher Consistency → Property of the Minimizer

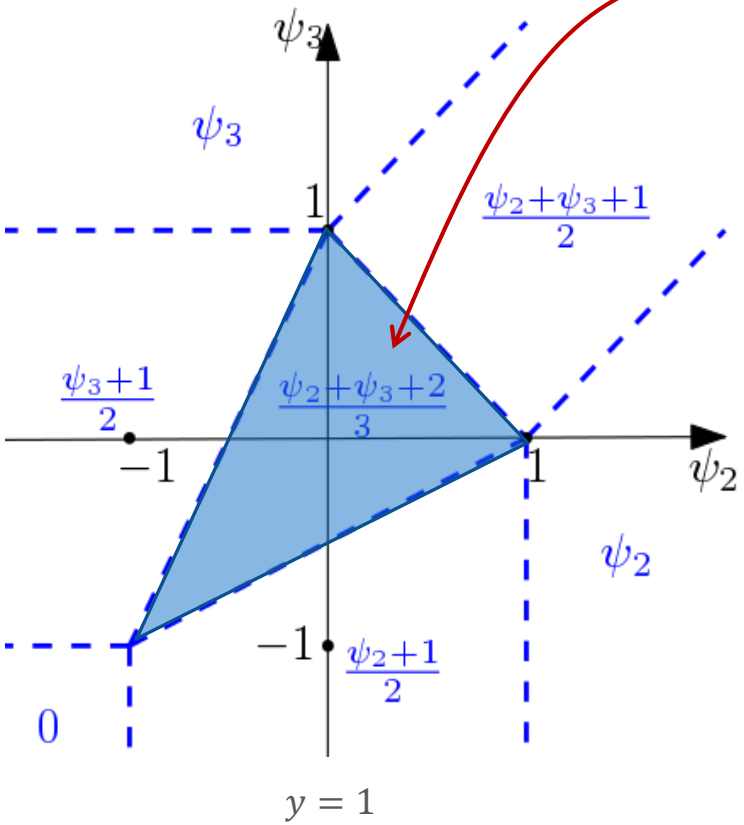
Fisher Consistency in Multiclass Zero-One Classification

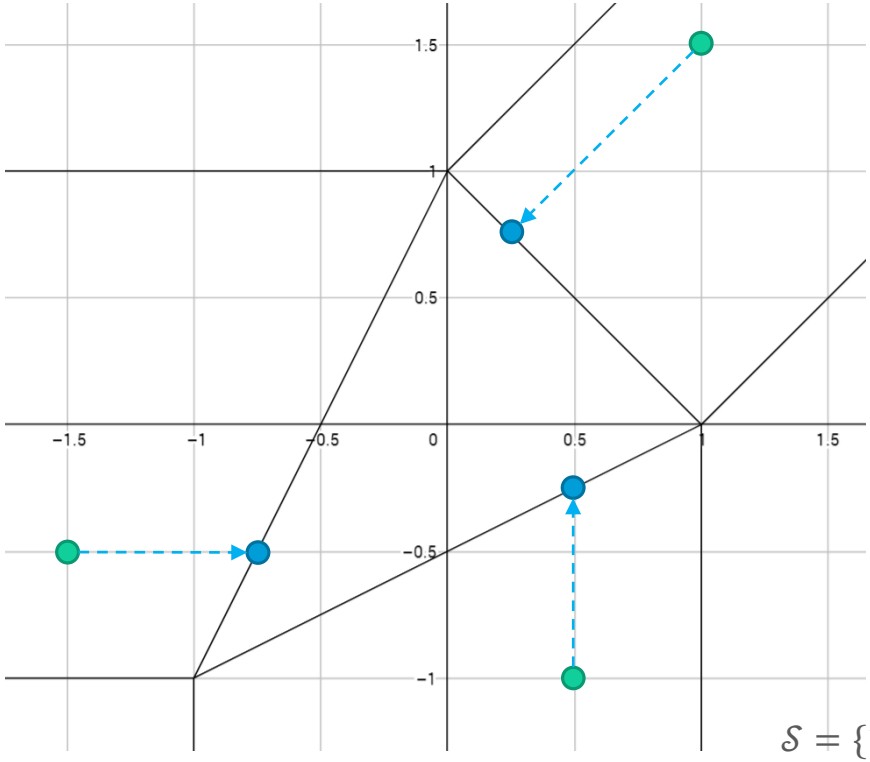
$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}}\, \mathbb{E}_{P(y|\mathbf{x})} \left[\text{surrogate}_{\mathbf{f}}(\mathbf{X}, Y)|\mathbf{X} = \mathbf{x}\right] \;\Rightarrow\; \underset{j}{\operatorname{argmax}}\, f_j^* \subseteq \underset{j}{\operatorname{argmax}}\, P(Y = j|\mathbf{x})$$

$\sum_{j=1}^{|\mathcal{Y}|} f_j(\mathbf{x}) = 0$ constraint is employed to remove redundant solution

The minimizer $\boldsymbol{f}^*$ lies
in the area defined by all class label

$$-\tfrac{1}{|\mathcal{Y}|} \leq f_j(\mathbf{x}) \leq \tfrac{|\mathcal{Y}|-1}{|\mathcal{Y}|}, \forall j \in \{1, \ldots, |\mathcal{Y}|\}$$





$AL^{0\text{-}1}$ → Fisher Consistency

Finding the minimizer f^*

$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}} \mathbb{E}_{P(y|\mathbf{x})}\left[\mathrm{AL}_{\mathbf{f}}^{0\text{-}1}(\mathbf{X}, Y)|\mathbf{X}=\mathbf{x}\right] = \operatorname*{argmin}_{\mathbf{f}} \sum_{y=1}^{|\mathcal{Y}|} P_y(\mathbf{x}) \max_{\substack{\mathcal{S}\subseteq\{1,\ldots,|\mathcal{Y}|\} \\ \mathcal{S}\neq\emptyset}} \frac{\sum_{j\in\mathcal{S}} \psi_{j,y}(\mathbf{x}) + |\mathcal{S}| - 1}{|\mathcal{S}|}$$

based on the properties of the minimizer

$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}} \sum_{y=1}^{|\mathcal{Y}|} P_y(\mathbf{x}) \left[\frac{\sum_{j=1}^{|\mathcal{Y}|}(f_j(\mathbf{x}) - f_y(\mathbf{x})) + |\mathcal{Y}| - 1}{|\mathcal{Y}|}\right]$$

$$\text{subject to } -\frac{1}{|\mathcal{Y}|} \leq f_j(\mathbf{x}) \leq \frac{|\mathcal{Y}|-1}{|\mathcal{Y}|}, \quad j \in \{1,\ldots,|\mathcal{Y}|\}; \quad \sum_{j=1}^{|\mathcal{Y}|} f_j(\mathbf{x}) = 0$$

$$\mathbf{f}^* = \max_{\mathbf{f}} \sum_{y=1}^{|\mathcal{Y}|} P_y(\mathbf{x}) f_y(\mathbf{x})$$

$$\text{subject to } -\frac{1}{|\mathcal{Y}|} \leq f_j(\mathbf{x}) \leq \frac{|\mathcal{Y}|-1}{|\mathcal{Y}|}, \quad j \in \{1,\ldots,|\mathcal{Y}|\}; \quad \sum_{j=1}^{|\mathcal{Y}|} f_j(\mathbf{x}) = 0$$

Solution:

$$f_j^*(\mathbf{x}) = \begin{cases} \frac{|\mathcal{Y}|-1}{|\mathcal{Y}|} & \text{if } j = \operatorname{argmax}_j P_j(\mathbf{x}) \\ -\frac{1}{|\mathcal{Y}|} & \text{otherwise.} \end{cases}$$

Fisher Consistent

AL^{0-1} → Optimization → Primal

Optimization of AL^{0-1} (Empirical Risk Minimization)

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{\substack{\mathcal{S} \subseteq \{1,\ldots,|\mathcal{Y}|\} \\ \mathcal{S} \neq \emptyset}} \frac{\sum_{j \in \mathcal{S}} \theta^{\mathrm{T}} \left(\phi(\mathbf{x}_i, j) - \phi(\mathbf{x}_i, y_i)\right) + |\mathcal{S}| - 1}{|\mathcal{S}|}$$

Gradient for a single sample $\boldsymbol{x_i}$

Let R be the set that maximize AL^{0-1} for $\boldsymbol{x_i}$,

The sub-gradient for a single sample $\boldsymbol{x_i}$ includes:

$$\frac{1}{|R|} \sum_{j \in R} \left[\phi(\mathbf{x}_i, j) - \phi(\mathbf{x}_i, y_i)\right]$$

Finding the set R:

Greedy algorithm:

1. Compute all $\psi_j \triangleq \theta^T\left(\phi(\boldsymbol{x}_i, j) - \phi(\boldsymbol{x}_i, y_i)\right)$
2. Sort ψ_j in non-descending order
3. Start with empty set $R = \emptyset$
4. Repeat:
5. Incrementally add j to the set R, update the value of AL^{0-1}
6. Until adding another one decrease the value of AL^{0-1}

AL^{0-1} → Optimization → Dual

Primal Quadratic Programming Formulation of AL^{0-1} with L2 regularization

$$\min_{\theta} \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n} \max_{\substack{\mathcal{S}\subseteq\{1,\ldots,|\mathcal{Y}|\}\\ \mathcal{S}\neq\emptyset}} \frac{\sum_{j\in\mathcal{S}} \theta^{\mathrm{T}}\left(\phi(\mathbf{x}_i, j) - \phi(\mathbf{x}_i, y_i)\right) + |\mathcal{S}| - 1}{|\mathcal{S}|}$$

Constrained Primal QP

$$\min_{\theta} \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to:} \quad \xi_i \geq \Delta_{i,k} \quad \forall i \in \{1,\ldots n\} k \in \{1,\ldots,2^{|\mathcal{Y}|}-1\}$$

$$\Delta_{i,1} = \psi_{1,y_i}(\mathbf{x}_i)$$

$$\vdots$$

$$\Delta_{i,|\mathcal{Y}|} = \psi_{|\mathcal{Y}|,y_i}(\mathbf{x}_i)$$

$$\Delta_{i,|\mathcal{Y}|+1} = \frac{\psi_{1,y_i}(\mathbf{x}_i) + \psi_{2,y_i}(\mathbf{x}_i) + 1}{2}$$

$$\vdots$$

$$\Delta_{i,2^{|\mathcal{Y}|}-1} = \frac{\sum_{j\in\mathcal{Y}} \psi_{j,y_i}(\mathbf{x}_i) + |\mathcal{Y}| - 1}{|\mathcal{Y}|}$$

Dual QP Formulation

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\sum_{k=1}^{2^{|\mathcal{Y}|}-1} \nu_{i,k}\,\alpha_{i,k} - \frac{1}{2}\sum_{i,j=1}^{m}\sum_{k,l=1}^{2^{|\mathcal{Y}|}-1} \alpha_{i,k}\alpha_{j,l}\left[\Lambda_{i,k}\cdot\Lambda_{j,l}\right]$$

$$\text{subject to} \quad \alpha_{i,k} \geq 0, \quad \sum_{k=1}^{2^{|\mathcal{Y}|}-1} \alpha_{i,k} = C, \; i \in \{1,\ldots,n\}, \; k \in \{1,\ldots,2^{|\mathcal{Y}|}-1\}$$

where:

$\Lambda_{i,k} = \frac{d\Delta_{i,k}}{d\theta}$, and $\nu_{i,k}$ is the constant part of $\Delta_{i,k}$

$$\Delta_{i,k} = \frac{\psi_{1,y_i}(\mathbf{x}_i) + \psi_{3,y_i}(\mathbf{x}_i) + \psi_{4,y_i}(\mathbf{x}_i) + 2}{3}$$

$$\nu_{i,k} = \frac{2}{3}$$

AL$^{0\text{-}1}$ → Optimization → Dual → Kernel Trick

Kernel trick

input space $\boldsymbol{x}_i$ → rich feature space $\omega(\boldsymbol{x}_i)$

Compute the dot products implicitly

$$K(\mathbf{x}_i, \mathbf{x}_j) = \omega(\mathbf{x}_i) \cdot \omega(\mathbf{x}_j)$$

Dual QP Formulation

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \sum_{k=1}^{2^{|\mathcal{Y}|}-1} \nu_{i,k}\, \alpha_{i,k} - \frac{1}{2} \sum_{i,j=1}^{m} \sum_{k,l=1}^{2^{|\mathcal{Y}|}-1} \alpha_{i,k}\alpha_{j,l}\, [\Lambda_{i,k} \cdot \Lambda_{j,l}]$$

$$\text{subject to} \quad \alpha_{i,k} \geq 0, \quad \sum_{k=1}^{2^{|\mathcal{Y}|}-1} \alpha_{i,k} = C, \; i \in \{1, \ldots, n\}, \; k \in \{1, \ldots, 2^{|\mathcal{Y}|} - 1\}$$

$$\Lambda_{i,k} \cdot \Lambda_{j,l} = c_{(i,k),(j,l)}\, K(\mathbf{x}_i, \mathbf{x}_j)$$

where: $c_{(i,k),(j,l)} = \sum_{m=1}^{|\mathcal{Y}|} \left(\frac{\mathbf{1}(m \in R_{i,k})}{|R_{i,k}|} - \mathbf{1}(m = y_i) \right) \left(\frac{\mathbf{1}(m \in R_{j,l})}{|R_{j,l}|} - \mathbf{1}(m = y_j) \right)$

$R_{i,k}$ is the set of labels included in the constraint $\Delta_{i,k}$

AL$^{0\text{-}1}$ → Optimization → Dual → Constraint Generation

Primal and Dual Optimization

Exponential number of constraints (in primal) and dual variables

Constraint Generation Algorithm

Algorithm 1 Constraint generation method

Input: Training data $(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_n, y_n)$, C, ϵ
$\theta \leftarrow \mathbf{0}$
$A_i^* \leftarrow \{\Delta_{i,k} | \Delta_{i,k} = \psi_{y_i,y_i}(\mathbf{x}_i)\}\ \forall i = 1, \ldots, n$ {Actual label enforces non-negativity}
repeat
  for $i \leftarrow 1, n$ **do**
    $a \leftarrow \arg\max_{k|\Delta_{i,k} \in A_i} \Delta_{i,k}$ {Find the most violated constraint}
    $\xi_i \leftarrow \max_{k|\Delta_{i,k} \in A_i^*} \Delta_{i,k}$ {Compute the example's current loss estimate}
    if $\Delta_{i,a} > \xi_i + \epsilon$ **then**
      $A_i^* \leftarrow A_i^* \cup \{\Delta_{i,a}\}$ {Add it to the enforced constraints set}
      $\boldsymbol{\alpha} \leftarrow$ Optimize dual over $A^* = \cup_i A_i^*$
      Compute θ from $\boldsymbol{\alpha}$: $\theta = -\sum_{i=1}^{n} \sum_{k|\Delta_{i,k} \in A_i^*} \alpha_{i,k} \Lambda_{i,k}$
    end if
  end for
until no A_i^* has changed in the iteration

Polynomial time convergence guarantee is provided

Experiment shows better convergence rate

AL^{0-1} → Experiments

Dataset properties and AL^{0-1} constraints

	Dataset	Properties				SVM	AL^{0-1} constraints added and active			
		#class	#train	# test	#feat.	constraints	Linear kernel		Gauss. kernel	
(1)	iris	3	105	45	4	210	213	13	223	38
(2)	glass	6	149	65	9	745	578	125	490	252
(3)	redwine	10	1119	480	11	10071	5995	1681	3811	1783
(4)	ecoli	8	235	101	7	1645	614	117	821	130
(5)	vehicle	4	592	254	18	1776	1310	311	1201	248
(6)	segment	7	1617	693	19	9702	4410	244	4312	469
(7)	sat	7	4435	2000	36	26610	11721	1524	11860	6269
(8)	optdigits	10	3823	1797	64	34407	7932	597	10072	2315
(9)	pageblocks	5	3831	1642	10	15324	9459	427	9155	551
(10)	libras	15	252	108	90	3528	1592	389	1165	353
(11)	vertebral	3	217	93	6	434	344	78	342	86
(12)	breasttissue	6	74	32	9	370	258	65	271	145

AL^{0-1} → Experiments → Results

Results for Linear Kernel and Gaussian Kernel

The **mean (standard deviation)** of the **accuracy**
Bold numbers: **best** or **not significantly worse** than the best

D	Linear Kernel				Gaussian Kernel			
	AL^{0-1}	WW	CS	LLW	AL^{0-1}	WW	CS	LLW
(1)	**96.3** (3.1)	**96.0** (2.6)	**96.3** (2.4)	79.7 (5.5)	**96.7** (2.4)	**96.4** (2.4)	**96.2** (2.3)	95.4 (2.1)
(2)	**62.5** (6.0)	**62.2** (3.6)	**62.5** (3.9)	52.8 (4.6)	**69.5** (4.2)	66.8 (4.3)	**69.4** (4.8)	**69.2** (4.4)
(3)	**58.8** (2.0)	**59.1** (1.9)	56.6 (2.0)	57.7 (1.7)	63.3 (1.8)	64.2 (2.0)	64.2 (1.9)	**64.7** (2.1)
(4)	**86.2** (2.2)	85.7 (2.5)	**85.8** (2.3)	74.1 (3.3)	**86.0** (2.7)	84.9 (2.4)	**85.6** (2.4)	**86.0** (2.5)
(5)	**78.8** (2.2)	**78.8** (1.7)	**78.4** (2.3)	69.8 (3.7)	**84.3** (2.5)	**84.4** (2.6)	83.8 (2.3)	**84.4** (2.6)
(6)	94.9 (0.7)	94.9 (0.8)	**95.2** (0.8)	75.8 (1.5)	**96.5** (0.6)	**96.6** (0.5)	96.3 (0.6)	96.4 (0.5)
(7)	84.9 (0.7)	**85.4** (0.7)	84.7 (0.7)	74.9 (0.9)	**91.9** (0.5)	**92.0** (0.6)	**91.9** (0.5)	**91.9** (0.4)
(8)	**96.6** (0.6)	96.5 (0.7)	96.3 (0.6)	76.2 (2.2)	98.7 (0.4)	98.8 (0.4)	98.8 (0.3)	**98.9** (0.3)
(9)	96.0 (0.5)	96.1 (0.5)	**96.3** (0.5)	92.5 (0.8)	**96.8** (0.5)	96.6 (0.4)	96.7 (0.4)	96.6 (0.4)
(10)	**74.1** (3.3)	72.0 (3.8)	71.3 (4.3)	34.0 (6.4)	83.6 (3.8)	83.8 (3.4)	**85.0** (3.9)	83.2 (4.2)
(11)	**85.5** (2.9)	**85.9** (2.7)	**85.4** (3.3)	79.8 (5.6)	**86.0** (3.1)	**85.3** (2.9)	85.5 (3.3)	84.4 (2.7)
(12)	**64.4** (7.1)	59.7 (7.8)	**66.3** (6.9)	58.3 (8.1)	**68.4** (8.6)	**68.1** (6.5)	**66.6** (8.9)	**68.0** (7.2)
avg	81.59	81.02	81.25	68.80	85.14	84.82	85.00	84.93
#b	9	6	8	0	9	6	6	7

Multiclass Zero-One Classification

	Fisher Consistent?	**Perform well** in low feature spaces?
1. The SVM WW Model (Weston et.al., 2002) Relative Margin Model	✗	✓
2. The SVM CS Model (Crammer and Singer, 1999) Relative Margin Model	✗	✓
3. The SVM LLW Model (Lee et.al., 2004) Absolute Margin Model	✓	✗
4. The AL^{0-1} (Adversarial Surrogate Loss) Relative Margin Model	✓	✓

Adversarial Surrogate Losses for Multiclass Ordinal Classification

Based on:

Rizal Fathony, Mohammad Bashiri, Brian D. Ziebart. "*Adversarial Surrogate Losses for Ordinal Regression*". Advances in Neural Information Processing Systems 30 (NIPS), 2017.

Ordinal Classification: Related Works

Support Vector Machine for Ordinal Classification

Extend hinge loss to ordinal classification

A. Threshold Methods (Sashua & Levin, 2003; Chu & Keerthi, 2005; Rennie & Srebro, 2005)

1. All Threshold (also called SVORIM)

$$\mathrm{AT}(\hat{f}, y) = \sum_{k=1}^{y-1} \delta(-(\eta_k - \hat{f})) + \sum_{k=y}^{|\mathcal{Y}|} \delta(\eta_k - \hat{f})$$

2. Immediate Threshold (also called SVOREX)

$$\mathrm{IT}(\hat{f}, y) = \delta(-(\eta_{y-1} - \hat{f})) + \delta(\eta_y - \hat{f})$$

δ is a surrogate for binary classification, e.g. the hinge loss

B. Reduction Framework (Li & Lin, 2007)

- Create $|\mathcal{Y}| - 1$ weighted extended samples for each training sample,
- Run binary classification with binary surrogate loss (e.g. hinge loss) on the extended samples

C. Cost Sensitive Classification Based Methods (Lin, 2008; Tu & Lin, 2010; Lin, 2014)

1. Cost Sensitive One-Versus-All (CSOVA)
2. Cost Sensitive One-Versus-One (CSOVO)
3. Cost Sensitive One-Sided-Regression (CSOSR)

Adversarial Surrogate Loss : Ordinal Classification

Adversarial Game

$$\min_{\hat{P}(\hat{y}|\mathbf{x})} \max_{\check{P}(\check{y}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{y}|\mathbf{x})\check{P}(\check{y}|\mathbf{x})} \left[|\hat{Y} - \check{Y}| \right]$$

$$\text{s.t. } \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X}, \check{Y})] = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$$

$$\min_{\theta} \mathbb{E}_{P(\mathbf{x})} \left[\max_{\check{\mathbf{p}}_{\mathbf{X}}} \min_{\hat{\mathbf{p}}_{\mathbf{X}}} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}} \mathbf{L}'_{\mathbf{X},\theta} \check{\mathbf{p}}_{\mathbf{X}} \right] \qquad \mathbf{L}'_{\mathbf{x},\theta} = \begin{bmatrix} f_1 - f_{\tilde{y}} & f_2 - f_{\tilde{y}} + 1 & f_3 - f_{\tilde{y}} + 2 & f_4 - f_{\tilde{y}} + 3 \\ f_1 - f_{\tilde{y}} + 1 & f_2 - f_{\tilde{y}} & f_3 - f_{\tilde{y}} + 1 & f_4 - f_{\tilde{y}} + 2 \\ f_1 - f_{\tilde{y}} + 2 & f_2 - f_{\tilde{y}} + 1 & f_3 - f_{\tilde{y}} & f_4 - f_{\tilde{y}} + 1 \\ f_1 - f_{\tilde{y}} + 3 & f_2 - f_{\tilde{y}} + 2 & f_3 - f_{\tilde{y}} + 1 & f_4 - f_{\tilde{y}} \end{bmatrix}$$

where $f_j(\mathbf{x}) = \theta^{\mathrm{T}} \phi(\mathbf{x}, j)$

Nash Equilibrium

$$v = \mathrm{AL}_{\theta}^{\mathrm{ord}}(\mathbf{x}, y) = \max_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{f_j + f_l + j - l}{2} - f_y$$

$$= \max_{j} \frac{f_j + j}{2} + \max_{l} \frac{f_l - l}{2} - f_y$$

$\mathrm{AL}^{\mathrm{ord}}$: maximization over pairs

Can be independently realized

AL^{ord} → Feature Representation

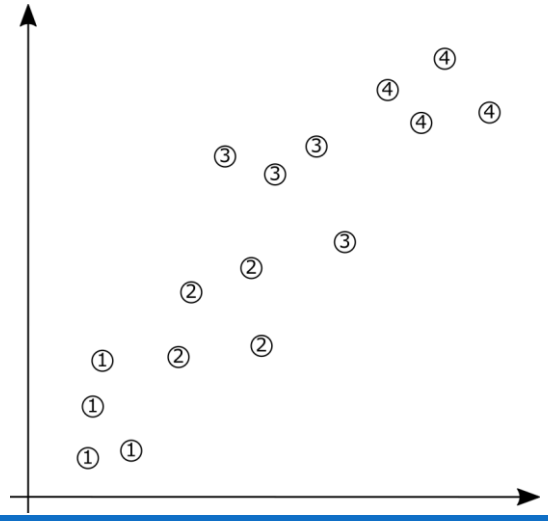
Thresholded Regression Representation

$$\phi_{th}(\mathbf{x}, y) = \begin{pmatrix} y\mathbf{x} \\ I(y \leq 1) \\ I(y \leq 2) \\ \vdots \\ I(y \leq |\mathcal{Y}| - 1) \end{pmatrix}$$

size : $m + |\mathcal{Y}| - 1$

m is the dimension of input space

- a single shared vector of feature weights
- a set of threshold



An example where thresholded regression representation are useful

Multiclass Representation

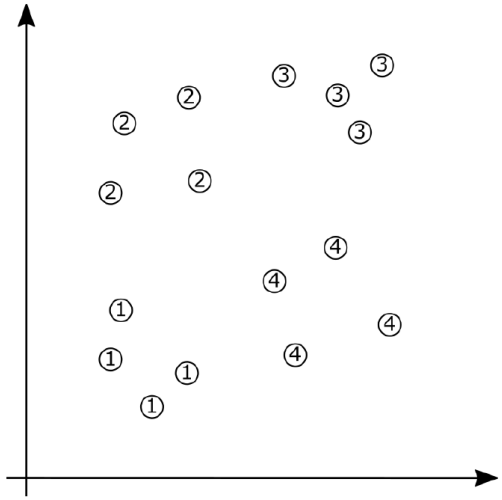
$$\phi_{mc}(\mathbf{x}, y) = \begin{pmatrix} I(y = 1)\mathbf{x} \\ I(y = 2)\mathbf{x} \\ I(y = 3)\mathbf{x} \\ \vdots \\ I(y = |\mathcal{Y}|)\mathbf{x} \end{pmatrix}$$

size : $m|\mathcal{Y}|$

m is the dimension of input space

- class specific feature weights

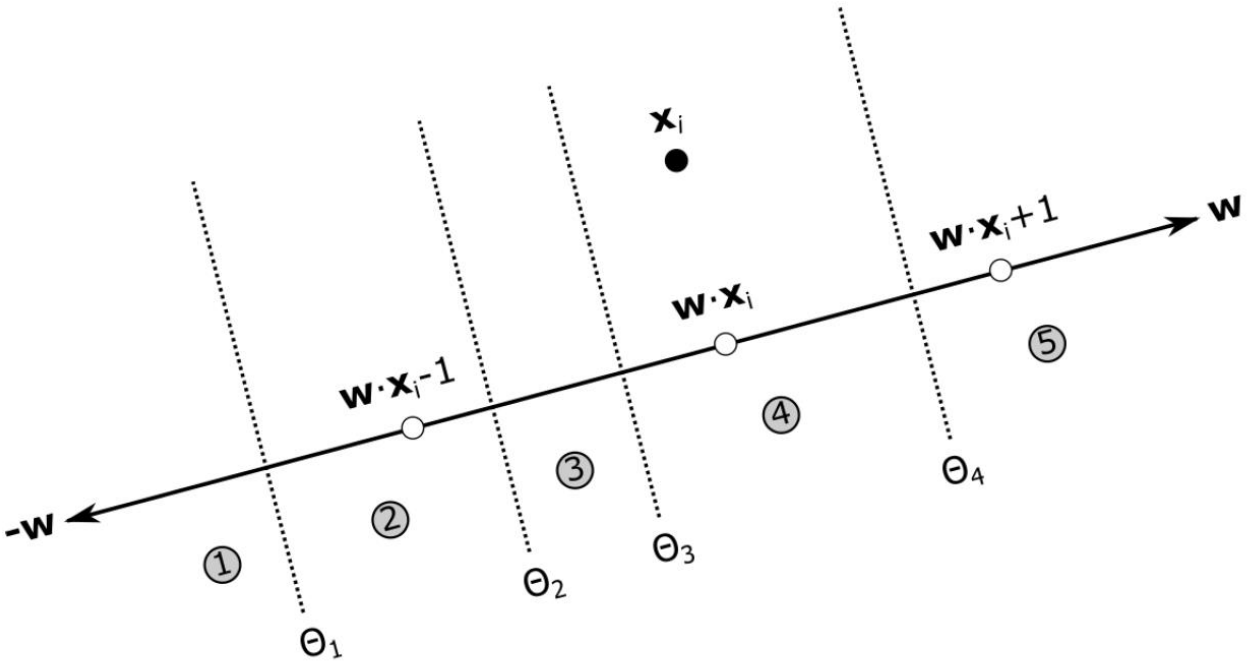
An example where multiclass representation are useful



AL^{ord} → Thresholded Regression Representation

AL^{ord} for the Thresholded Regression Representation

$$\mathrm{AL}^{\text{ord-th}}(\mathbf{x}_i, y_i) = \max_j \frac{j(\mathbf{w}\cdot\mathbf{x}_i + 1) + \sum_{k\geq j}\eta_k}{2} + \max_l \frac{l(\mathbf{w}\cdot\mathbf{x}_i - 1) + \sum_{k\geq l}\eta_k}{2} - y_i\mathbf{w}\cdot\mathbf{x}_i - \sum_{k\geq y_i}\eta_k$$

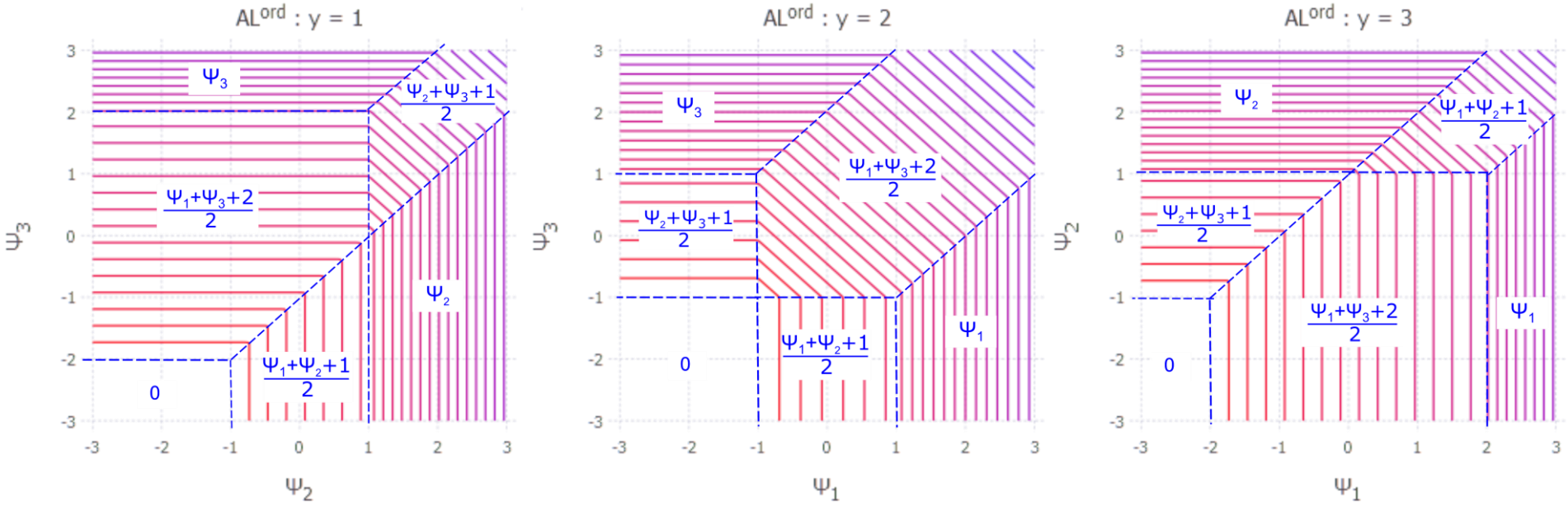


AL^{ord-th} : based on averaging the threshold label predictions for potentials $\boldsymbol{w}\cdot\boldsymbol{x}_i + 1$ and $\boldsymbol{w}\cdot\boldsymbol{x}_i - 1$

AL^{ord} → Multiclass Representation

AL^{ord} for the Multiclass Representation

$$\mathrm{AL}^{\text{ord-mc}}(\mathbf{x}_i, y_i) = \max_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{\mathbf{w}_j \cdot \mathbf{x}_i + \mathbf{w}_l \cdot \mathbf{x}_i + j - l}{2} - \mathbf{w}_{y_i} \cdot \mathbf{x}_i$$



$AL^{ord\text{-}mc}$: maximization over $\frac{|\mathcal{Y}|(|\mathcal{Y}|+1)}{2}$ hyperplanes

AL^{ord} → Fisher Consistency

Fisher Consistency in Ordinal Classification

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}}\, \mathbb{E}_{P(y|\mathbf{x})} \left[\text{surrogate}_{\mathbf{f}}(\mathbf{X}, Y)|\mathbf{X} = \mathbf{x}\right] \Rightarrow \underset{j}{\operatorname{argmax}}\, f_j^* \subseteq \underset{j}{\operatorname{argmin}} \sum_y P_y\, |j - y|$$

$\max f_j(\mathbf{x}) = 0$ constraint is employed to remove redundant solution

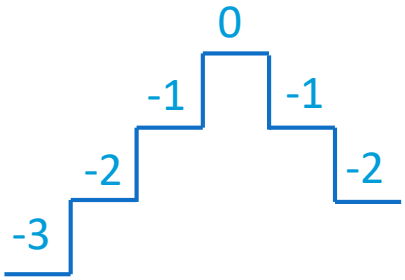
Properties of the minimizer $\boldsymbol{f}^*$

The minimizer $\boldsymbol{f}^*$ satisfies the **loss reflective** property

examples:

[-1, 0, -1, -2] [0, -1, -2, -3]

[-2, -1, 0, -1, -2] [-3, -2, -1, 0, -1, -2]



Finding the minimizer $\boldsymbol{f}^*$

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}}\, \mathbb{E}_{P(y|\mathbf{x})} \left[\text{AL}_{\mathbf{f}}^{\text{ord}}(\mathbf{X}, Y)|\mathbf{X} = \mathbf{x}\right] = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_y P_y \left[\max_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{f_j + f_l + j - l}{2} - f_y\right]$$

↓ based on the **loss reflective** property

Equivalent with finding j^* (the class in a loss reflective $\boldsymbol{f}$ that has 0 value)

$$j^* = \underset{j}{\operatorname{argmin}} \sum_{y=1}^{|\mathcal{Y}|} P_y |j - y| \longrightarrow \text{Fisher Consistent}$$

AL^{ord} → Optimization → Primal

Optimization of AL^{ord} (Empirical Risk Minimization)

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left[\max_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{\theta^{\mathrm{T}} \left[\phi(\mathbf{x}_i, j) + \phi(\mathbf{x}_i, l)\right] + j - l}{2} - \theta^{\mathrm{T}} \left[\phi(\mathbf{x}_i, y_i)\right] \right]$$

Stochastic Average Gradient (SAG) (Schimidt et.al. 2013, 2015)

Average over the gradient of each example from the last iteration it was selected

Requires storing the gradient of each sample.

SAG for AL$^{ord\text{-}mc}$

Objective:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left[\max_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{\mathbf{w}_j \cdot \mathbf{x}_i + \mathbf{w}_l \cdot \mathbf{x}_i + j - l}{2} - \mathbf{w}_{y_i} \cdot \mathbf{x}_i \right]$$

Gradient for a single sample $\boldsymbol{x_i}$:

$$j^*, l^* = \operatorname*{argmax}_{j,l \in \{1,\ldots,|\mathcal{Y}|\}} \frac{\mathbf{w}_j \cdot \mathbf{x}_i + \mathbf{w}_l \cdot \mathbf{x}_i + j - l}{2} - \mathbf{w}_{y_i} \cdot \mathbf{x}_i$$

assuming $j^* \neq l^* \neq y_i$

$$\nabla_{\mathbf{w}_{j^*}} = \tfrac{1}{2}\mathbf{x}_i \qquad \nabla_{\mathbf{w}_{y_i}} = -\mathbf{x}_i$$

$$\nabla_{\mathbf{w}_{l^*}} = \tfrac{1}{2}\mathbf{x}_i \qquad \nabla_{\mathbf{w}_k} = \mathbf{0} \quad k \in \{1,\ldots,|\mathcal{Y}|\} \backslash \{j^*, l^*, y_i\}$$

Store j^* and l^* instead of full gradient

AL^{ord} → Optimization → Dual

Primal Quadratic Programming Formulation of AL^{ord} with L2 regularization

$$\min_{\theta} \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n}\left[\max_{j\in 1,\ldots,|\mathcal{Y}|}\frac{\theta\cdot\phi(\mathbf{x}_i,j)+j}{2} + \max_{j\in 1,\ldots,|\mathcal{Y}|}\frac{\theta\cdot\phi(\mathbf{x}_i,j)-j}{2} - \theta\cdot\phi(\mathbf{x}_i,y_i)\right]$$

Constrained Primal QP

$$\min_{\theta} \frac{1}{2}\|\theta\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i + \frac{C}{2}\sum_{i=1}^{n}\delta_i$$

$$\text{subject to:} \quad \xi_i \geq \theta\cdot\phi(\mathbf{x}_i,j) - \theta\cdot\phi(\mathbf{x}_i,y_i) + j \qquad \forall i\in\{1,\ldots n\}; j\in\{1,\ldots,|\mathcal{Y}|\}$$

$$\delta_i \geq \theta\cdot\phi(\mathbf{x}_i,j) - \theta\cdot\phi(\mathbf{x}_i,y_i) - j \qquad \forall i\in\{1,\ldots n\}; j\in\{1,\ldots,|\mathcal{Y}|\}$$

Dual QP Formulation

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i,j} j(\alpha_{i,j}-\beta_{i,j}) - \frac{1}{2}\sum_{i,j,k,l}(\alpha_{i,j}+\beta_{i,j})(\alpha_{k,l}+\beta_{k,l})(\phi(\mathbf{x}_i,j)-\phi(\mathbf{x}_i,y_i))\cdot(\phi(\mathbf{x}_k,l)-\phi(\mathbf{x}_l,y_k))$$

$$\text{subject to: } \alpha_{i,j}\geq 0; \beta_{i,j}\geq 0; \sum_j \alpha_{i,j} = \tfrac{C}{2}; \sum_j \beta_{i,j} = \tfrac{C}{2}; i,k\in\{1,\ldots,n\}; j,l\in\{1,\ldots,|\mathcal{Y}|\}$$

Kernel trick can also be easily applied!

AL^{ord} → Experiments

Dataset properties

Dataset	# class	# train	# test	# features
diabetes	5	30	13	2
pyrimidines	5	51	23	27
triazines	5	130	56	60
wisconsin	5	135	59	32
machinecpu	10	146	63	6
autompg	10	274	118	7
boston	5	354	152	13
stocks	5	665	285	9
abalone	10	2923	1254	10
bank	10	5734	2458	8
computer	10	5734	2458	21
calhousing	10	14447	6193	8

AL^{ord} → Experiments → Linear Kernel

Results for Linear Kernel

Dataset	Threshold-based models				Multiclass-based models				
	$AL^{ord\text{-}th}$	RED^{th}	AT	IT	$AL^{ord\text{-}mc}$	RED^{mc}	CSOSR	CSOVO	CSOVA
diabetes	**0.696**	**0.715**	0.731	0.827	**0.629**	**0.700**	**0.715**	0.738	0.762
pyrimidines	0.654	0.678	0.615	0.626	**0.509**	0.565	**0.520**	0.576	**0.526**
triazines	**0.607**	0.683	0.649	0.654	0.670	0.673	0.677	0.738	0.732
wisconsin	**1.077**	**1.067**	**1.097**	1.175	1.136	1.141	1.208	1.275	1.338
machinecpu	**0.449**	**0.456**	**0.458**	**0.467**	0.518	0.515	0.646	0.602	0.702
autompg	**0.551**	**0.550**	**0.550**	0.617	0.599	0.602	0.741	0.598	0.731
boston	0.316	**0.304**	**0.306**	**0.298**	0.311	0.311	0.353	**0.294**	0.363
stocks	0.324	0.317	0.315	0.324	0.168	0.175	0.204	**0.147**	0.213
abalone	0.551	0.547	0.546	0.571	**0.521**	**0.520**	0.545	0.558	0.556
bank	0.461	0.460	0.461	0.461	**0.445**	**0.446**	0.732	0.448	0.989
computer	0.640	0.635	0.633	0.683	**0.625**	**0.624**	0.889	0.649	1.055
calhousing	1.190	1.183	1.182	1.225	1.164	**1.144**	1.237	1.202	1.601
average	0.626	0.633	0.629	0.661	0.613	0.618	0.706	0.652	0.797
# bold	5	5	4	2	5	5	2	2	1

AL^{ord} → Experiments → Gaussian Kernel

Results for Gaussian Kernel

All Threshold Intermediate Threshold

Dataset	AL$^{ord\text{-}th}$	SVORIM	SVOREX
diabetes	**0.696**	**0.665**	**0688**
pyrimidines	**0.478**	0.539	0.550
triazines	**0.609**	**0.612**	**0.604**
wisconsin	1.090	1.113	**1.049**
machinecpu	**0.452**	0.652	0.628
autompg	**0.529**	0.589	0.593
boston	**0.278**	0.324	0.316
stocks	**0.103**	**0.099**	**0.100**
average	0.531	0.574	0.566
# bold	7	3	4

Conclusion

Conclusion

A Establish connections between Adversarial Prediction and ERM

B Propose Adversarial Surrogate Losses:

1 Align better with the original loss
Optimizing Adversarial Loss in the ERM Framework
= Optimizing the original loss in the Adversarial Prediction Framework

2 Guarantee Fisher Consistency

3 Enable computational efficiency for rich feature space
via kernel trick and dual parameter sparsity

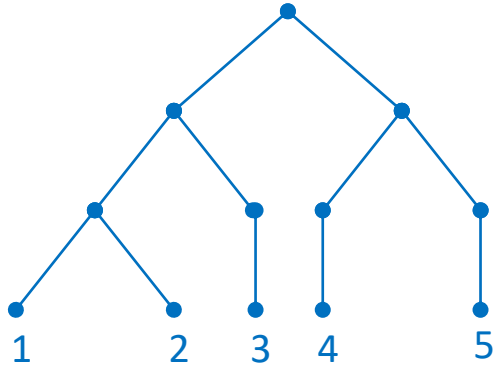
4 Perform well in practice

Ongoing and Future Works

[1.] Taxonomy-based Classification

Adversarial Surrogate Loss for Taxonomy-based Classification

Example:

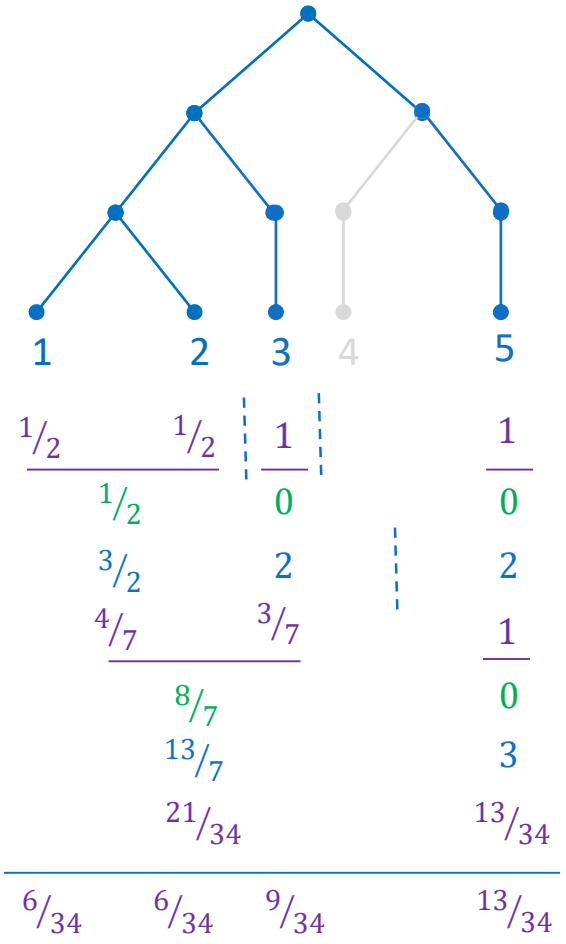


$$\begin{bmatrix} 0 & 1 & 2 & 3 & 3 \\ 1 & 0 & 2 & 3 & 3 \\ 2 & 2 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 2 \\ 3 & 3 & 3 & 2 & 0 \end{bmatrix}$$

Nash equilibrium:

Analyze non-zero probability strategy of the adversary

Adversary's probability:



$\mathcal{S} = \{1,2,3,5\}$

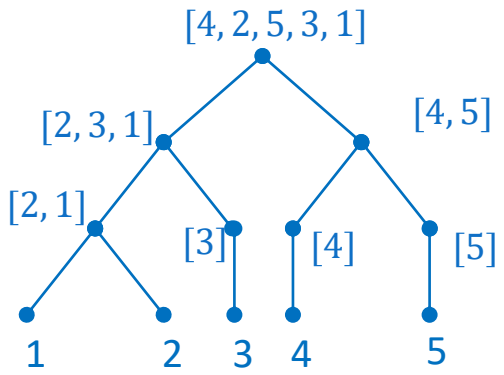
[1.] Taxonomy-based Classification → Algorithm

Algorithm for finding the adversarial loss

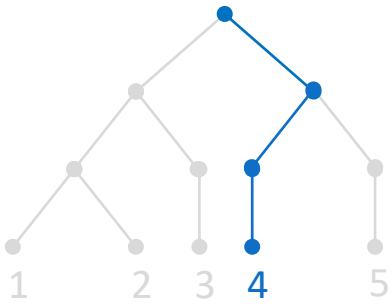
Potentials:

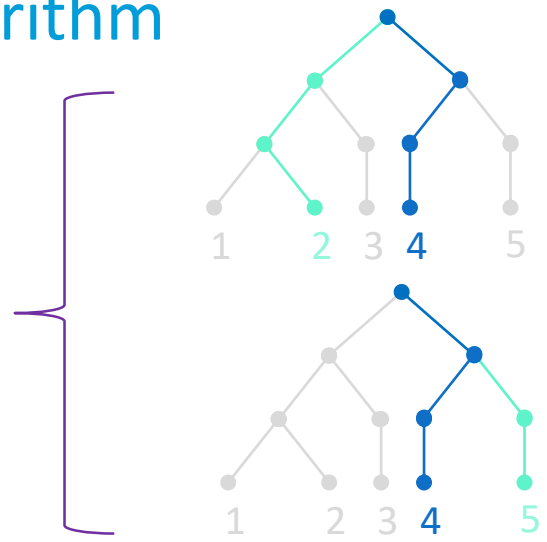
$$\boldsymbol{\psi} = [0.1, 1.2, 0.5, 1.5, 0.7]$$

Sorted index for each non-leaf nodes:



Algorithm:



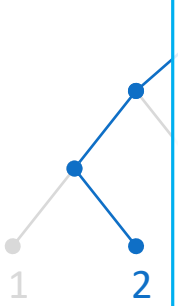


Complexity

$O(md^2k^2)$

m : max # of children
d : depth of the tree
k : # of class

Ref: LP complexity: $O(k^{3.5})$



Completed:

- Game analysis
- Algorithm
- Complexity analysis
- Algorithm implementation

Future:

- Formal proof
- More efficient implementation
- Real data experiments

[2.] Sequence Prediction with Ordinal Classification Loss

Adversarial Surrogate Loss for Sequence Prediction with Ordinal Classification Loss

Adversarial game over joint distributions:

$$\min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x}) \in \Xi} \mathbb{E}_{P(\mathbf{x})\hat{P}(\hat{\mathbf{y}}|\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x})} \left[\text{loss}(\hat{\mathbf{Y}}, \check{\mathbf{Y}}) \right]$$

$$\text{where: } \Xi : \mathbb{E}_{P(\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x})}[\Phi(\mathbf{X}, \check{\mathbf{Y}})] = \tilde{\mathbf{\Phi}},$$

$$\min_{\theta} \mathbb{E}_{P(\mathbf{x},\mathbf{y})} \left[\max_{\check{\mathbf{p}}_{\mathbf{X}}} \min_{\hat{\mathbf{p}}_{\mathbf{X}}} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}} \mathbf{L}'_{\mathbf{X},\theta} \check{\mathbf{p}}_{\mathbf{X}} \right]$$

Exponential size of matrix $L_{x,\theta}$

$$\text{loss}(\hat{\mathbf{y}}, \check{\mathbf{y}}) = \sum_{t=1}^{T} |\hat{y}_t - \check{y}_t| \qquad \Phi(\mathbf{x}, \check{\mathbf{y}}) = \sum_{t=1}^{T-1} \phi(\mathbf{x}, y_t, y_{t+1})$$

Double oracle (Li, et.al., 2016)

guarantee

Nash equilibrium:

Exists an equilibrium where only **two** strategies that has **non-zero** probability(*)

$$v = \max_{\mathbf{y},\mathbf{y}'} \frac{1}{2} \left[\sum_{t=1}^{T} |y_t - y'_t| + \theta^T \sum_{t=1}^{T-1} \left(\phi(\mathbf{x}, y_t, y_{t+1}) + \phi(\mathbf{x}, y'_t, y'_{t+1}) \right) \right] - \theta^T \sum_{t=1}^{T-1} \phi(\mathbf{x}, \tilde{y}_t, \tilde{y}_{t+1})$$

Can be solved using dynamic programming, complexity: $O(T|\mathcal{Y}|^4)$

Completed:

- Game analysis for ordinal classification loss
- Algorithm
- Complexity analysis
- Algorithm implementation

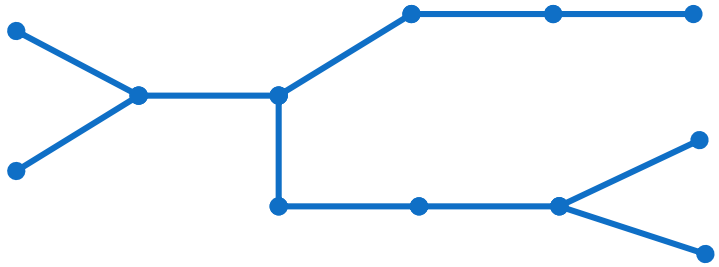
Future:

- Formal proof
- Real data experiments
- Extend analysis to other additive multiclass losses (e.g. zero-one loss)

(*) Formal proof is in on going work

[3.] Adversarial Graphical Model

Focus on tree structures with additive loss



Adversarial Game Over Marginal Distributions

$$\max_{\substack{\check{P}_{i,j}(\check{y}_i,\check{y}_j)\\ \check{P}_i(\check{y}_i)}} \min_{\hat{P}_i(\hat{y}_i)} \sum_i \mathbb{E}_{\check{P}_i(\check{y}_i)\hat{P}_i(\hat{y}_i)}\text{loss}(\hat{y}_i,\check{y}_i) + \sum_{i,j}\mathbb{E}_{\check{P}_{i,j}(\check{y}_i,\check{y}_j)}\theta^T\phi(\mathbf{x},\check{y}_i,\check{y}_j) - \sum_{i,j}\theta^T\phi(\mathbf{x},\tilde{y}_i,\tilde{y}_j)$$

$$s.t. \sum_j \check{P}_{i,j}(\check{y}_i,\check{y}_j) = \check{P}_i(\check{y}_i),\ \forall i; \quad \sum_i \check{P}_{i,j}(\check{y}_i,\check{y}_j) = \check{P}_j(\check{y}_j),\ \forall j$$

$$\max_{\substack{\check{P}_{i,j}(\check{y}_i,\check{y}_j)\\ \check{P}_i(\check{y}_i)}} \sum_i v_i + \sum_{i,j}\mathbb{E}_{\check{P}_{i,j}(\check{y}_i,\check{y}_j)}\theta^T\phi(\mathbf{x},\check{y}_i,\check{y}_j) - \sum_{i,j}\theta^T\phi(\mathbf{x},\tilde{y}_i,\tilde{y}_j)$$

$$s.t. \sum_j \check{P}_{i,j}(\check{y}_i,\check{y}_j) = \check{P}_i(\check{y}_i),\ \forall i; \quad \sum_i \check{P}_{i,j}(\check{y}_i,\check{y}_j) = \check{P}_j(\check{y}_j),\ \forall j$$

$$v_i \le \sum_l \check{P}_i(l)\text{loss}(k,l),\ \forall i,k.$$

Completed:

- Linear Programing Formulation
- Polynomial runtime $O(|\mathcal{Y}|^7|E|^{3.5})$
- Naïve implementation

Future:

- Better ways to solve the LP
- Adversarial surrogate loss for graphical models
- Real data experiments

[*.] Timeline

[1.] Taxonomy Classification

- Formal proof
- More efficient implementation
- Real data experiments

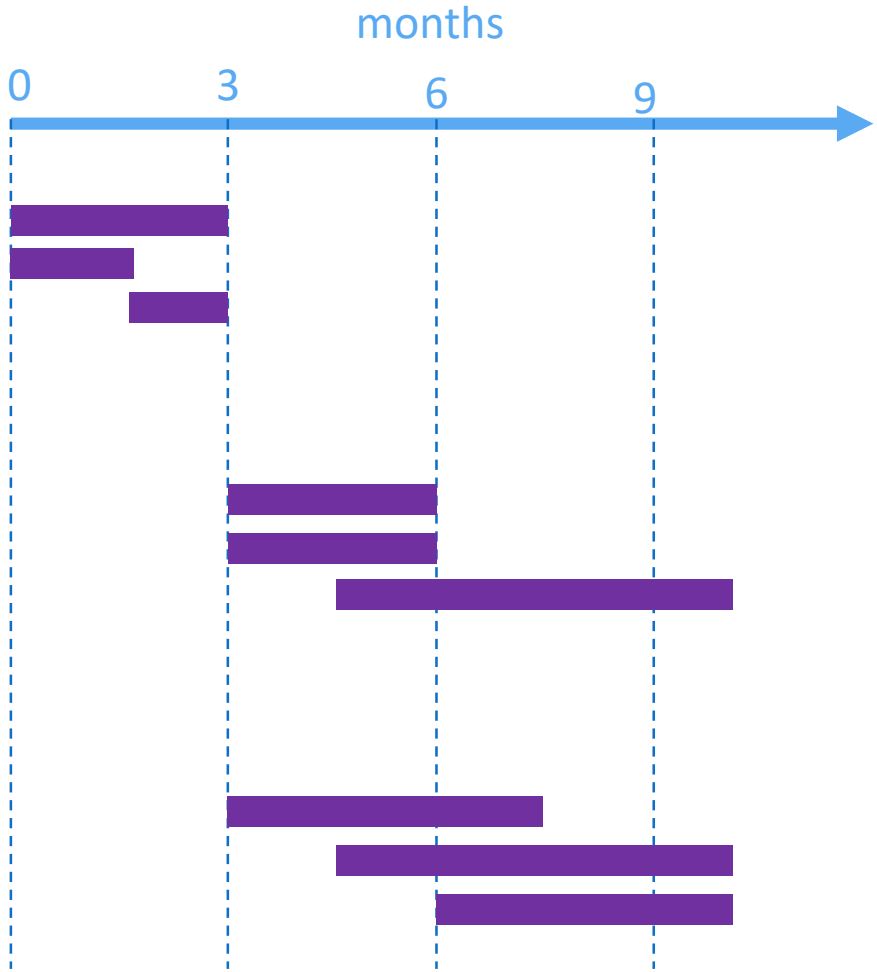
[2.] Adversarial Loss for Sequence Prediction

- Formal proof for ordinal classification case
- Real data experiments
- Extend analysis to other additive multiclass losses

[3.] Adversarial Graphical Models

(with focus on tree structures)

- Better ways to solve the LP
- Adversarial surrogate loss for graphical models
- Real data experiments



Outline

1. The Adversarial Surrogate Loss for Multiclass Zero-One Classification
2. Adversarial Surrogate Losses for Multiclass Ordinal Classification
3. Ongoing and Future Works

12

Thank You