Multi-class Support Vector Machine

Rizal Zaini Ahmad Fathony

November 10, 2016

University of Illinois at Chicago

Introduction

Support Vector Machine

- The Support Vector Machine is a classification algorithm developed based on a geometric intuition of finding large margin

Support Vector Machine

- The Support Vector Machine is a classification algorithm developed based on a geometric intuition of finding large margin
- SVM has demonstrated successful results in binary classification problems

Support Vector Machine

- The Support Vector Machine is a classification algorithm developed based on a geometric intuition of finding large margin
- SVM has demonstrated successful results in binary classification problems
- Several efforts have been proposed to bring the success of SVM in binary classification problems into multi-class classification problems

Support Vector Machine

- The Support Vector Machine is a classification algorithm developed based on a geometric intuition of finding large margin
- SVM has demonstrated successful results in binary classification problems
- Several efforts have been proposed to bring the success of SVM in binary classification problems into multi-class classification problems
- We will study different approaches in formulating multi-class SVM in both theoretical properties (Fisher consistency) and empirical performance of the models

Table of Contents

Formulations

Standard SVM Formulation

- Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_n, y_n)\}$$

  - $\mathbf{x}_i$: vector of features for the i-th example
  - y_i : label for the i-th example, $y_i \in \{-1, +1\}$
  - n : total number of training examples

Standard SVM Formulation

- Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_n, y_n)\}$$

  - $\mathbf{x}_i$: vector of features for the i-th example
  - y_i : label for the i-th example, $y_i \in \{-1, +1\}$
  - n : total number of training examples

- Goal:

Find the *maximum-margin* hyperplane

i.e. the hyperplane that separates positive examples from negative examples which has the largest margin

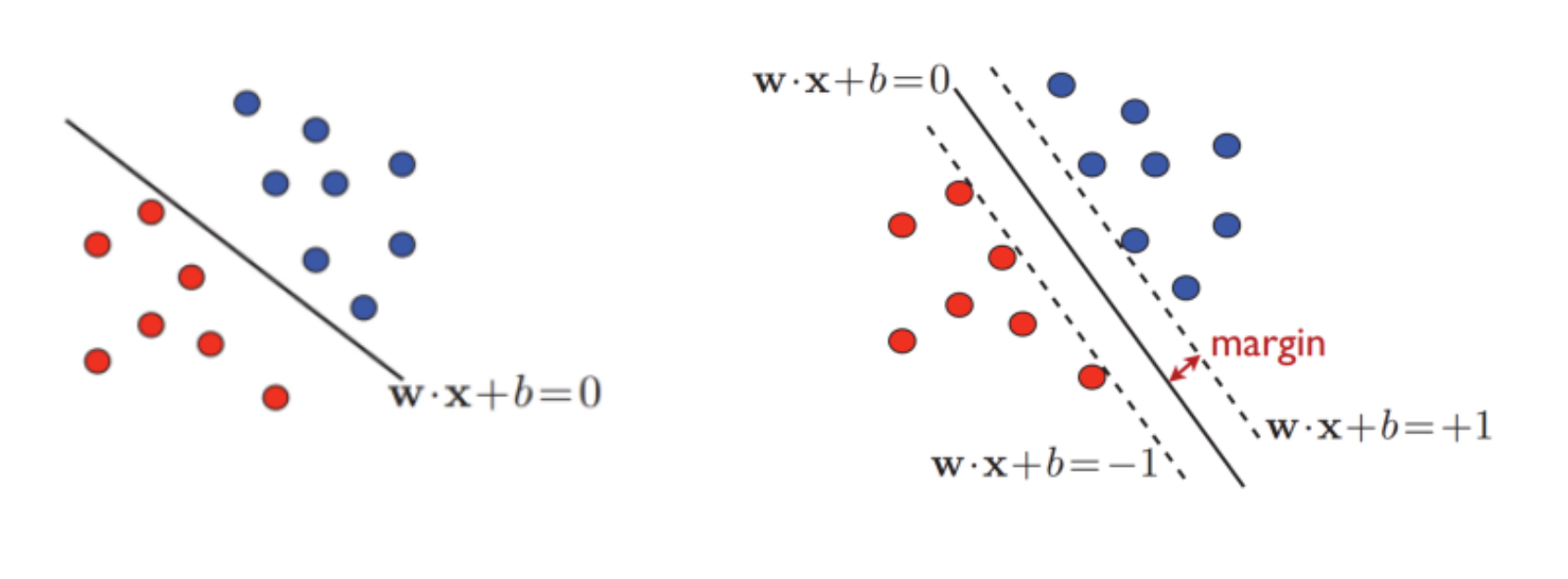
Hyperplane

- A hyperplane in a d dimensional data $\mathbb{R}^d$:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

  - $\mathbf{w} \in \mathbb{R}^d$: a non-zero vector normal to the hyperplane
  - $b \in \mathbb{R}$: a scalar

Maximum-margin hyperplane (right) and another hyperplane (left)



Margin:

$$\rho = \frac{1}{\|\mathbf{w}\|}$$

Marginal hyperplanes:

$$\mathbf{w}\cdot\mathbf{x}+b=+1$$
$$\mathbf{w}\cdot\mathbf{x}+b=-1$$

Mohri, M. *et al. Foundations of machine learning* (MIT press, 2012).

Optimization

- Maximizing margin $\rho = \frac{1}{\|\mathbf{w}\|}$
- Equivalent:
  Minimizing $\|\mathbf{w}\|$ or $\frac{1}{2}\|\mathbf{w}\|^2$

Optimization

- Maximizing margin $\rho = \frac{1}{\|\mathbf{w}\|}$
- Equivalent:
  Minimizing $\|\mathbf{w}\|$ or $\frac{1}{2}\|\mathbf{w}\|^2$
- Denote: $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b \rightarrow$ potential
- Marginal hyperplane definition
  $\Rightarrow |\mathbf{w} \cdot \mathbf{x}_i + b| \geq 1$ for each example $i \in [1, n]$

Optimization and Prediction

- Quadratic Programming Formulation:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall i \in [1,n]. \end{aligned}$$

Optimization and Prediction

- Quadratic Programming Formulation:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall i \in [1,n]. \end{aligned}$$

- Prediction for a new data $\mathbf{x}$:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}\cdot\mathbf{x} + b).$$

Soft-Margin SVM

- Real world data are not always **linearly separable**
- Allow violation, i.e. some points $\mathbf{x}_i$ can have

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \ngeq 1,$$

but add penalty to the optimization when there is a violation

Soft-Margin SVM

- Real world data are not always **linearly separable**
- Allow violation, i.e. some points $\mathbf{x}_i$ can have

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \not\geq 1,$$

  but add penalty to the optimization when there is a violation
- Introduce a slack variable ξ_i for each point $i \in [1, n]$

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

- $C \geq 0$: a parameter for balancing between maximizing margin and minimizing the violation

Hinge Loss

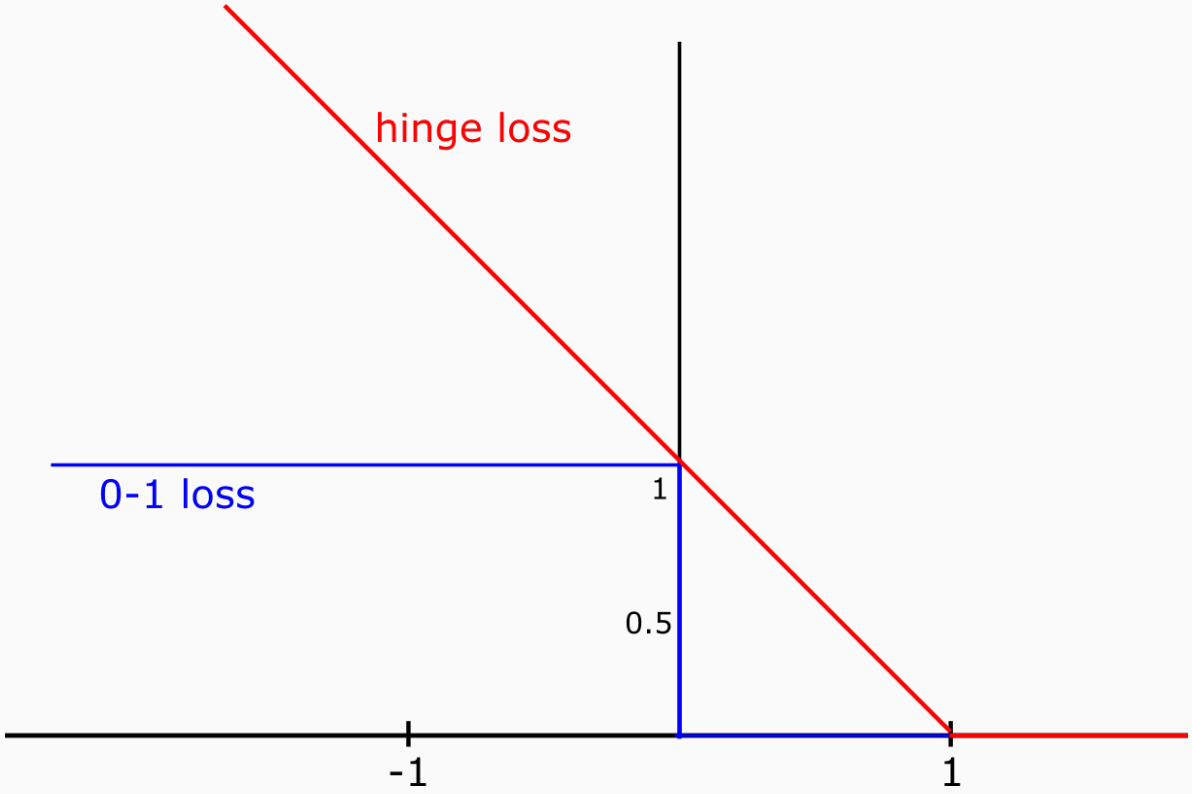
- Note that: $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b$
- The penalty ξ_i for example $\mathbf{x}_i$:
  - 0, if $y_i f(\mathbf{x}_i) \geq 1$
  - $1 - y_i f(\mathbf{x}_i)$, if $y_i f(\mathbf{x}_i) < 1$

Hinge Loss

- Note that: $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b$
- The penalty ξ_i for example $\mathbf{x}_i$:
  - 0, if $y_i f(\mathbf{x}_i) \geq 1$
  - $1 - y_i f(\mathbf{x}_i)$, if $y_i f(\mathbf{x}_i) < 1$
- The loss:

$$[1 - y_i f(\mathbf{x}_i)]_+$$

  where $[u]_+ = u$ if $u \geq 0$ and 0 otherwise
- **Hinge loss**


hinge loss
0-1 loss
1
0.5
-1
1

Multi-class Classification

- Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_n, y_n)\}$$

  - $\mathbf{x}_i$: vector of features for the i-th example
  - y_i : label for the i-th example

    y_i can have an integer value from 1 to k; $y_i \in [1, k]$
  - k : the number of classes
  - n : total number of training examples

Multi-class SVM Formulations

A. Multi-machine Formulations
   - One Versus One (OVO)
   - One Versus All (OVA)

Multi-class SVM Formulations

A. Multi-machine Formulations
   - One Versus One (OVO)
   - One Versus All (OVA)

B. All-in-one Machine Formulations
   - Weston and Watkins (WW) Formulation
   - Crammer and Singer (CS) Formulation
   - Lee, Lin, and Wahba (LLW) Formulation

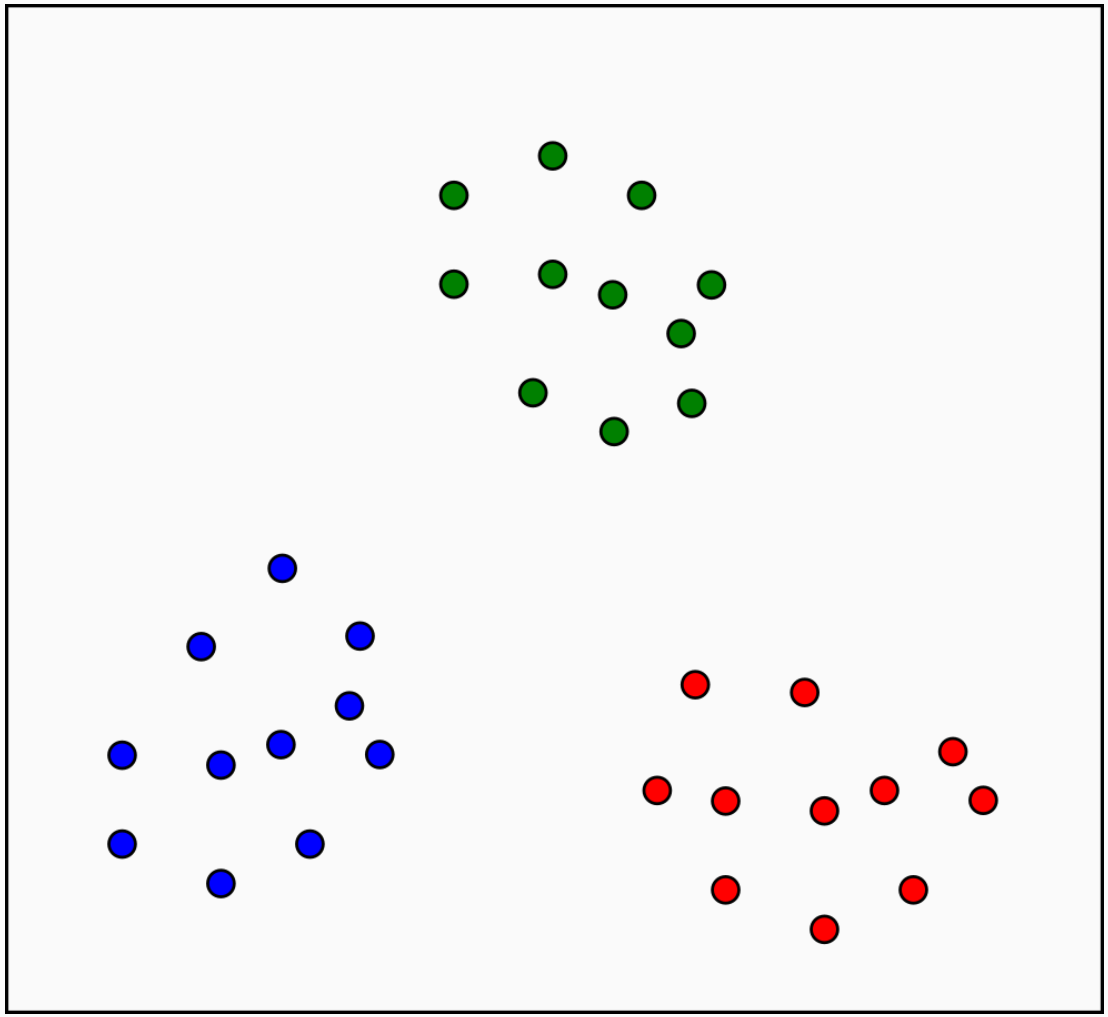
Multi-machine Formulations

- Divide a multi-class classification problem into several binary classification tasks.

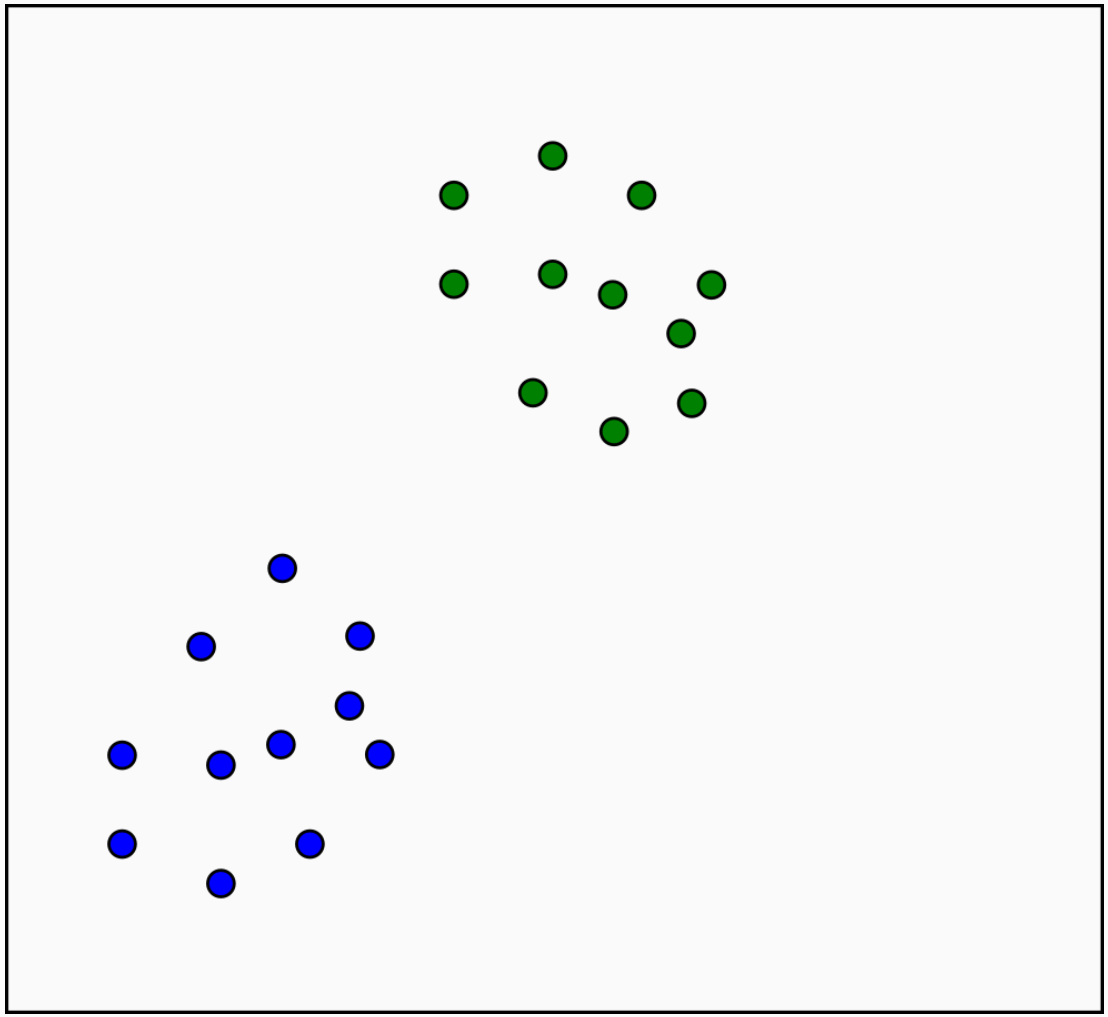
One Versus One

- Construct a binary classification problem for each pair of classes $(a, b) \in \{(a, b) | a < b \ , \ a, b \in [1, k]\}$
- Each classifier differentiate a-th class from b-th class. Resulting in a decision function $h^{a-b}(\mathbf{x})$

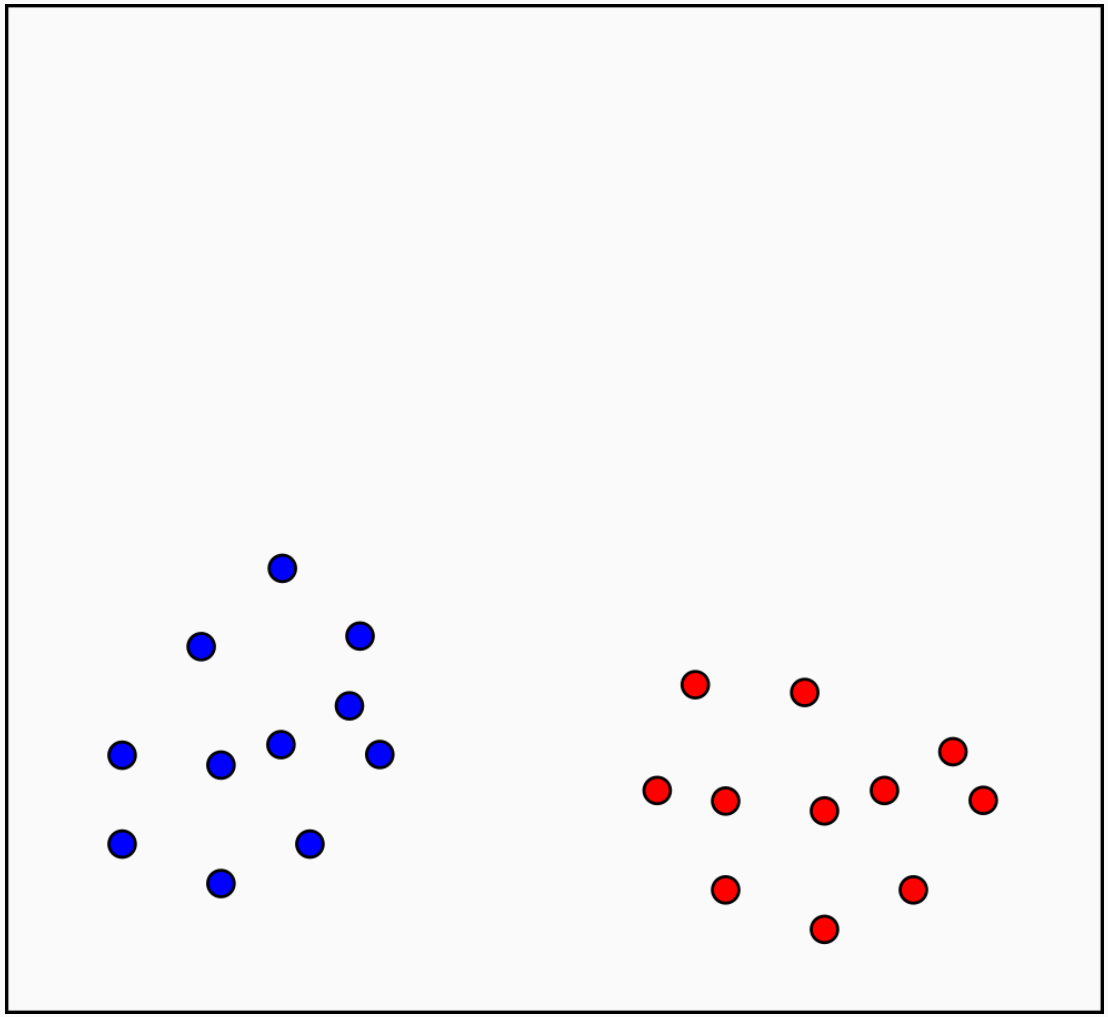
Deng, N. *et al. Support vector machines: optimization based theory, algorithms, and extensions* (CRC press, 2012).



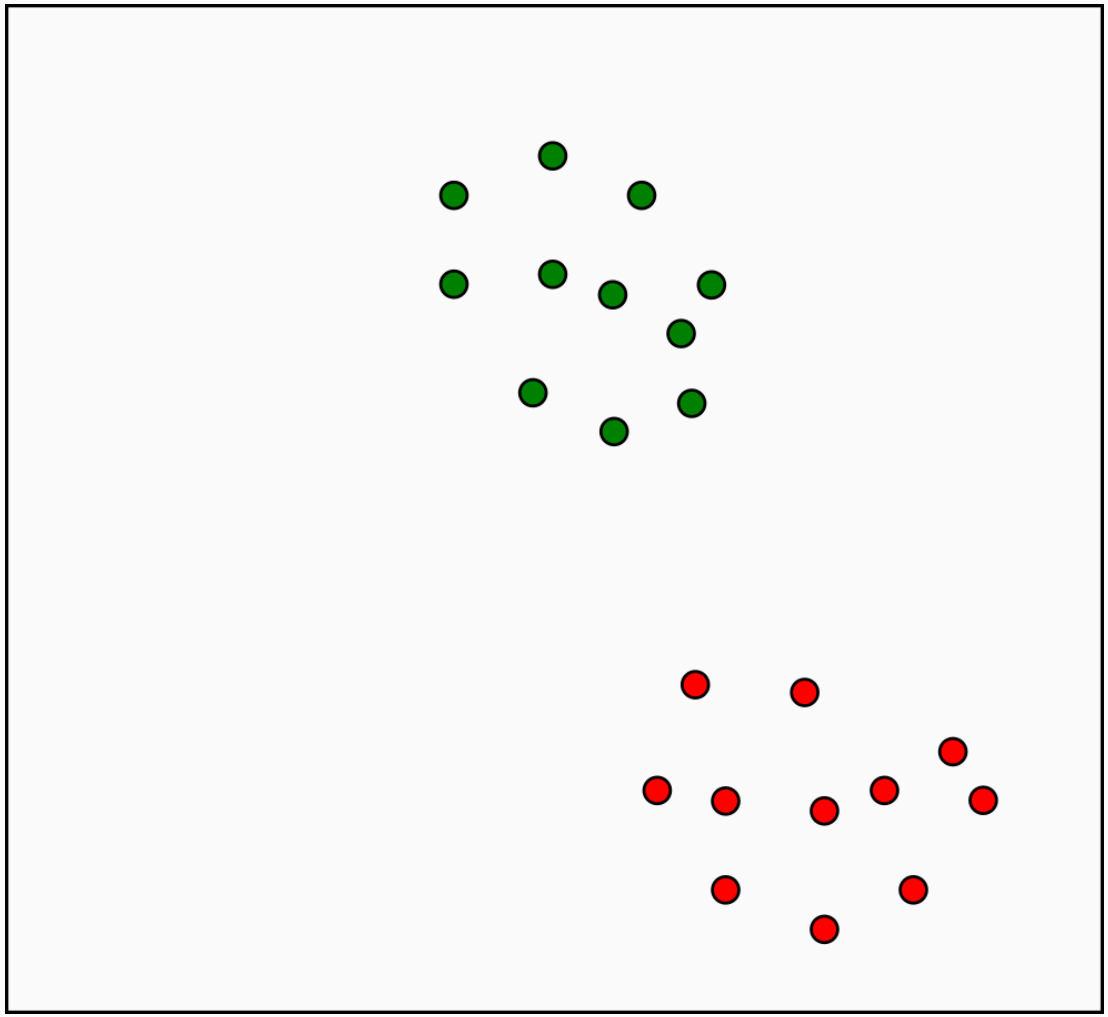
Three classes classification



First OVO model



Second OVO model



Third OVO model

One Versus One

- $k(k-1)/2$ decision functions in total
- Final decision: take the class which has the most votes

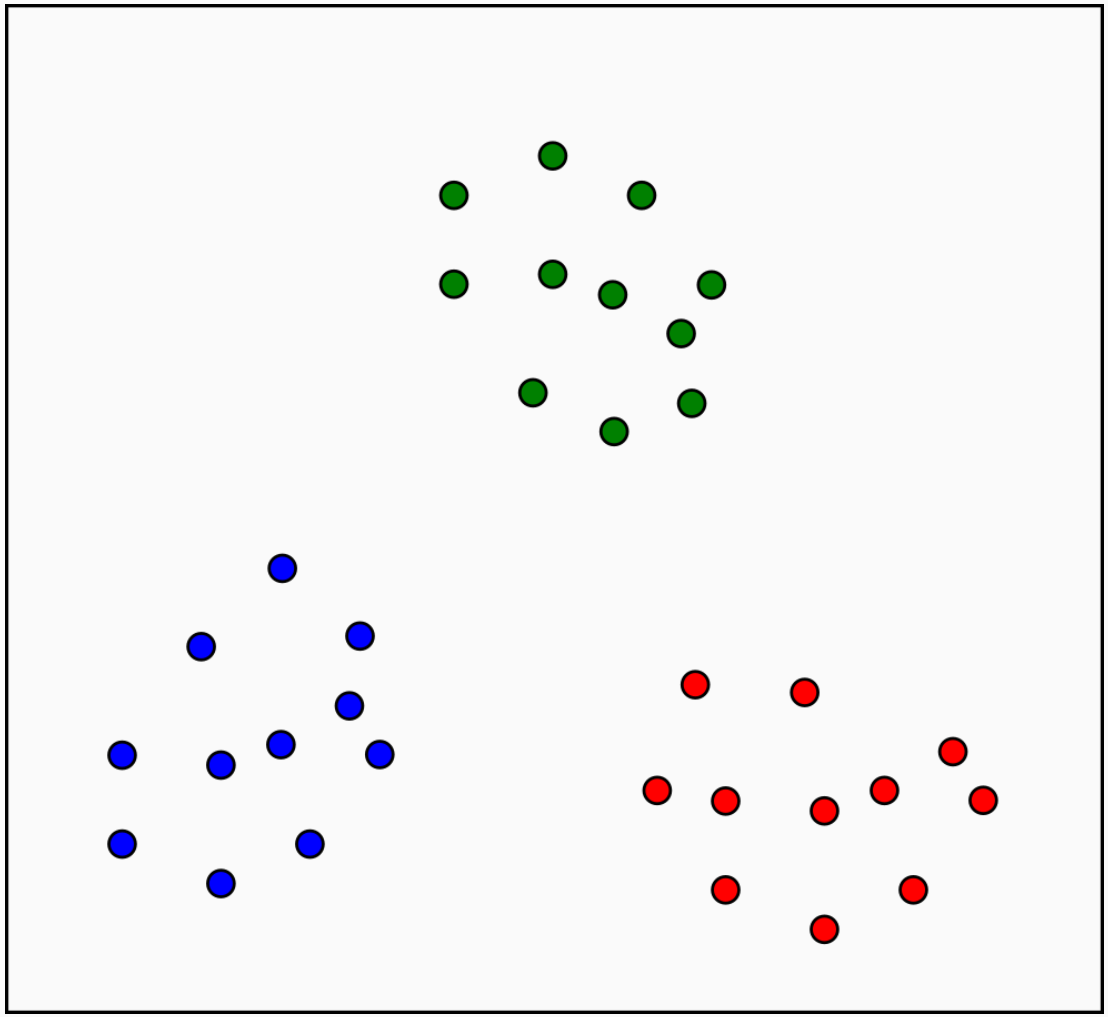
Deng, N. *et al. Support vector machines: optimization based theory, algorithms, and extensions* (CRC press, 2012).

One Versus All

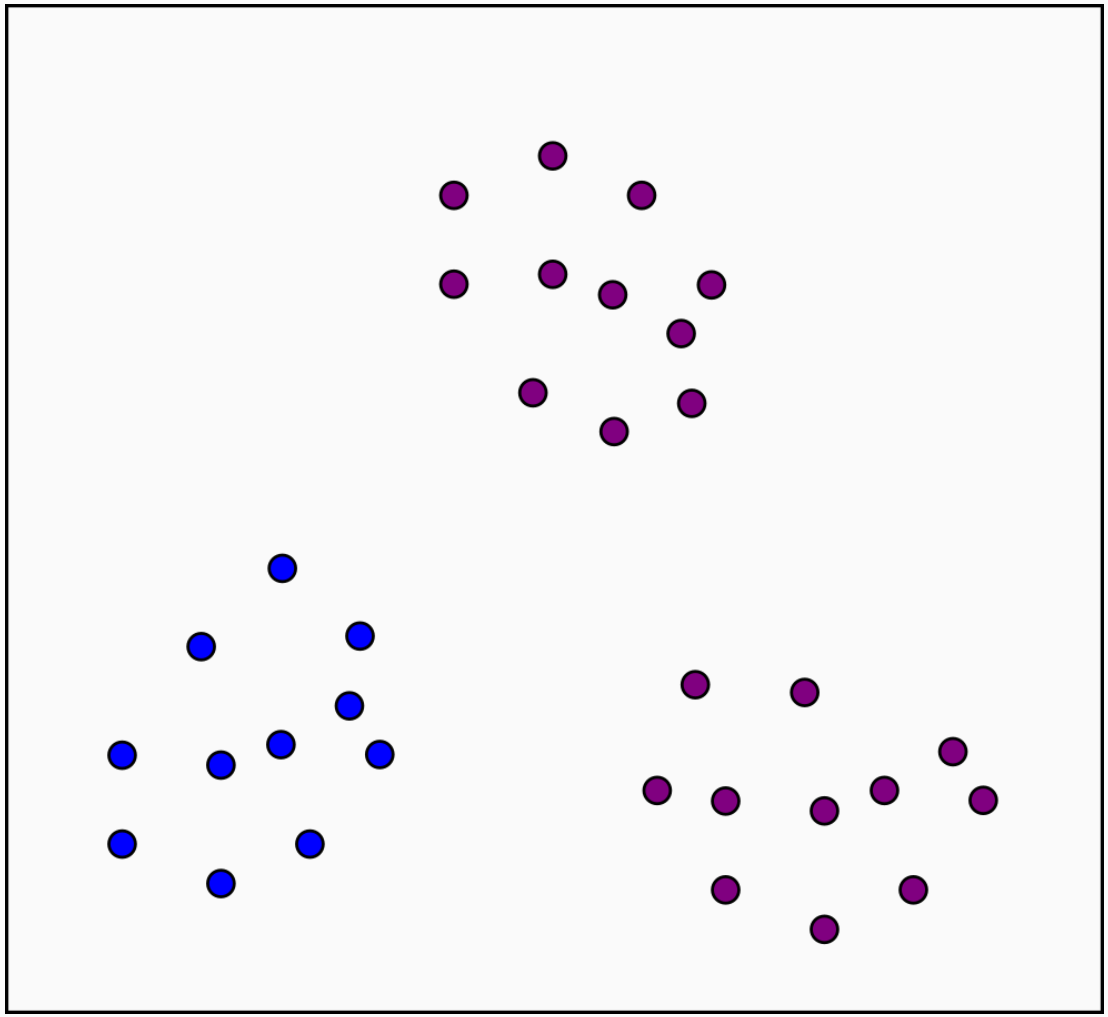
- Construct k binary classifiers
- The a-th binary classifier tries to separate a-th class from the rest

---

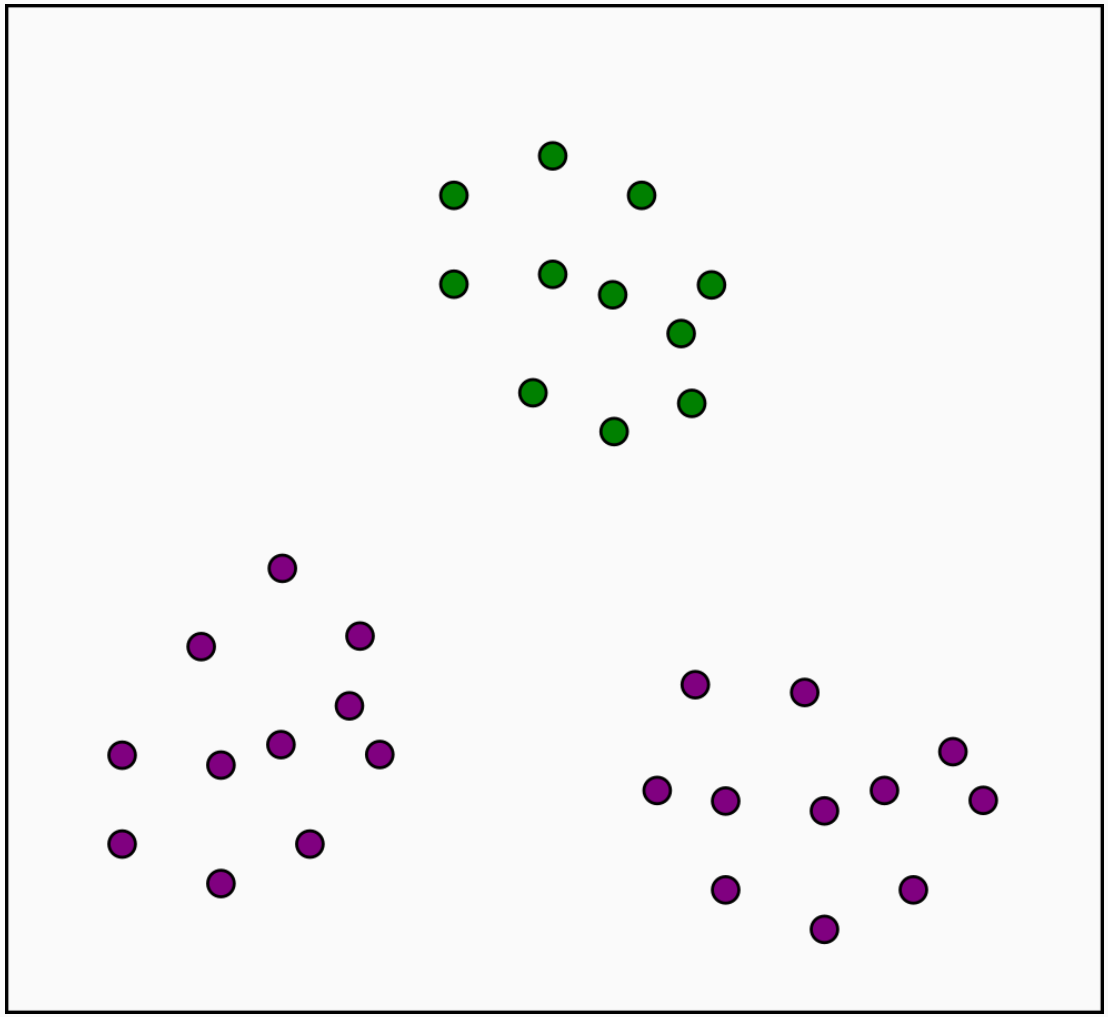
Deng, N. *et al. Support vector machines: optimization based theory, algorithms, and extensions* (CRC press, 2012).



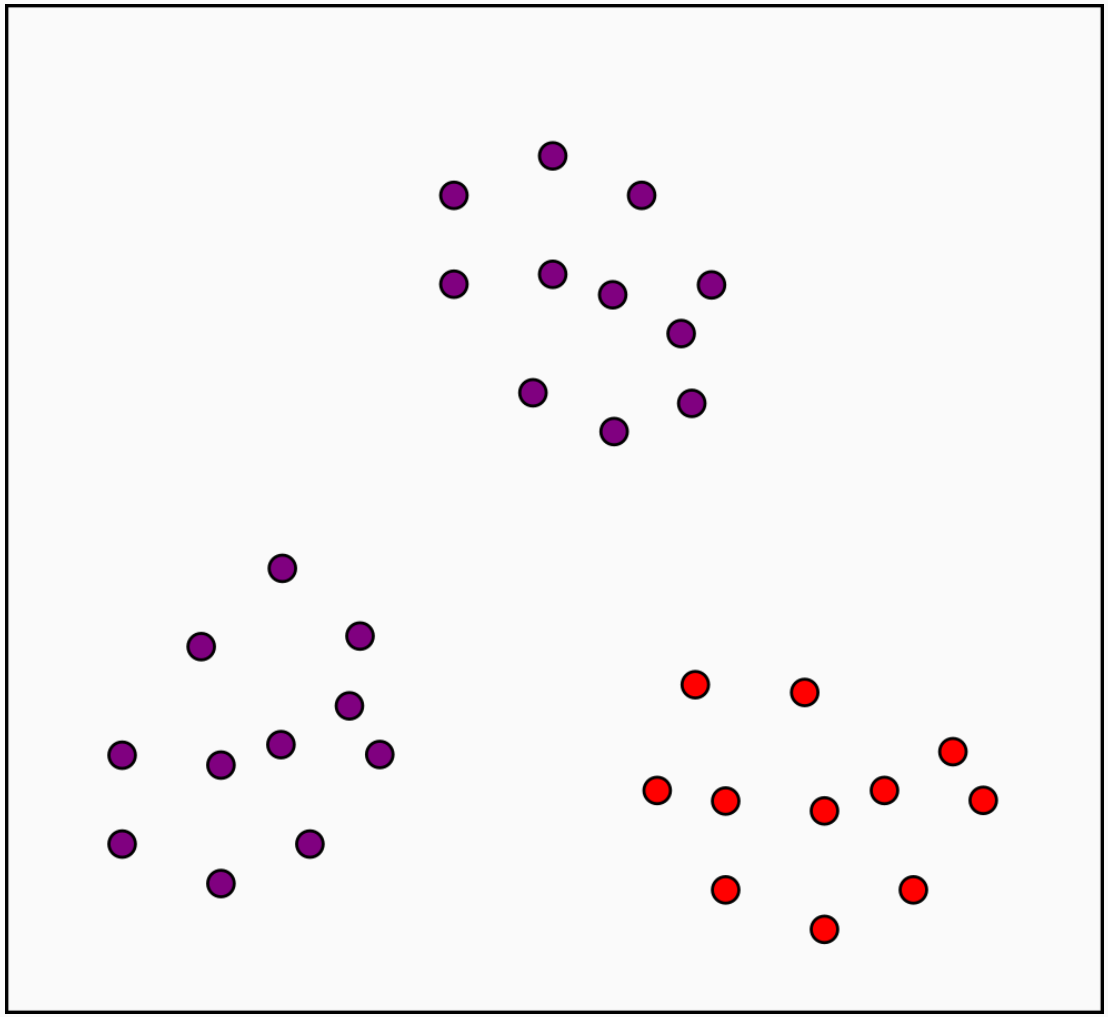
Three classes classification



First OVA model



Second OVA model



Third OVA model

One Versus All

- Let $f_a(\mathbf{x}) = \mathbf{w}_a \cdot \mathbf{x} + b_a$ be the potential function constructed by the a-th binary classifier, where:
  The classifier will pick class a if $f_a(\mathbf{x}) > 0$
- Final decision:

$$\hat{y} = \underset{a \in [1,k]}{\operatorname{argmax}} f_a(\mathbf{x})$$

Deng, N. *et al. Support vector machines: optimization based theory, algorithms, and extensions* (CRC press, 2012).

All-in-one Machine Formulations

- Construct a single model that considers all classes
- Directly modifies the optimization in binary SVM by:
  1. Modifying the objective function
  2. Modifying the constraints

All-in-one Machine Formulations

- Construct a single model that considers all classes
- Directly modifies the optimization in binary SVM by:
  1. Modifying the objective function
  2. Modifying the constraints
- Formulations:
  1. Weston and Watkins (WW) Formulation
  2. Crammer and Singer (CS) Formulation
  3. Lee, Lin, and Wahba (LLW) Formulation

Weston and Watkins (WW) Formulation

- A parameter $\mathbf{w}_j$ for each class
- A slack variable $\xi_{i,j}$ for each example and each class

---

Weston, J., Watkins, C., *et al. Support vector machines for multi-class pattern recognition.* in *ESANN* **99** (1999), 219–224.

Weston and Watkins (WW) Formulation

- A parameter $\mathbf{w}_j$ for each class
- A slack variable $\xi_{i,j}$ for each example and each class
- Define: the potential function for class j

$$f_j(\mathbf{x}_i) = \mathbf{w}_j \cdot \mathbf{x}_i + b_j$$

Weston, J., Watkins, C., *et al. Support vector machines for multi-class pattern recognition.* in *ESANN* **99** (1999), 219–224.

Standard Binary SVM

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

Weston and Watkins (WW) Formulation

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\sum_{j\in\{1,\cdots,k\}\setminus y_i}\xi_{i,j} \\ \text{subject to:} \quad & (\mathbf{w}_{y_i}\cdot\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j\cdot\mathbf{x}_i + b_j) \geq 2 - \xi_{i,j} \\ & \xi_{i,j} \geq 0, \quad i \in [1, n], \quad j \in \{1,\cdots,k\}\setminus y_i \end{aligned}$$

Weston and Watkins (WW) Formulation

- Prediction:

$$h(\mathbf{x}) = \underset{j}{\operatorname{argmax}} \left[\mathbf{w}_j \cdot \mathbf{x} + b_j\right] = \underset{j}{\operatorname{argmax}} \, f_j(\mathbf{x})$$

Weston, J., Watkins, C., *et al. Support vector machines for multi-class pattern recognition.* in *ESANN* **99** (1999), 219–224.

Crammer and Singer (CS) Formulation

- A parameter $\mathbf{w}_j$ for each class
- Only one slack variable ξ_i for each example, (instead of k)

Crammer, K. & Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research* **2,** 265–292 (2002).

Weston and Watkins (WW) Formulation

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\sum_{j\in\{1,\cdots,k\}\backslash y_i}\xi_{i,j}$$

$$\text{subject to:} \quad (\mathbf{w}_{y_i}\cdot\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j\cdot\mathbf{x}_i + b_j) \geq 2 - \xi_{i,j}$$

$$\xi_{i,j} \geq 0, \quad i \in [1,n], \quad j \in \{1,\cdots,k\}\backslash y_i$$

Crammer and Singer (CS) Formulation

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to:} \quad (\mathbf{w}_{y_i}\cdot\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j\cdot\mathbf{x}_i + b_j) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad i \in [1,n], \quad j \in \{1,\cdots,k\}\backslash y_i$$

Lee, Lin, and Wahba (LLW) Formulation

- A parameter $\mathbf{w}_j$ for each class
- A slack variable $\xi_{i,j}$ for each example and each class

---

Lee, Y. *et al.* Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* **99,** 67–81 (2004).

Lee, Lin, and Wahba (LLW) Formulation

- A parameter $\mathbf{w}_j$ for each class
- A slack variable $\xi_{i,j}$ for each example and each class
- Use the absolute potential value $f_j(\mathbf{x}_i)$
  Instead of using the relative potential difference $f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i)$

Lee, Y. *et al.* Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* **99,** 67–81 (2004).

Weston and Watkins (WW) Formulation

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\sum_{j\in\{1,\cdots,k\}\backslash y_i}\xi_{i,j}$$

$$\text{subject to:} \quad \xi_{i,j} \geq 2 + f_j(\mathbf{x}_i) - f_{y_i}(\mathbf{x}_i)$$

$$\xi_{i,j} \geq 0, \quad i \in [1,n], \quad j \in \{1,\cdots,k\}\backslash y_i$$

Lee, Lin, and Wahba (LLW) Formulation

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\sum_{j\in\{1,\cdots,k\}\backslash y_i}\xi_{i,j}$$

$$\text{subject to:} \quad \xi_{i,j} \geq f_j(\mathbf{x}_i) + \frac{1}{k-1}; \qquad \sum_{j=1}^{k} f_j(\mathbf{x}_i) = 0$$

$$\xi_{i,j} \geq 0; \qquad i \in [1,n], \quad j \in \{1,\cdots,k\}\backslash y_i$$

Fisher Consistency

Fisher Consistency in Binary Classification

- Fisher consistency / Bayes Consistency:
  Requires a classifier to asymptotically yields Bayes decision boundary

[1] Lin, Y. Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6,** 259–275 (2002).

Fisher Consistency in Binary Classification

- Fisher consistency / Bayes Consistency:
  Requires a classifier to asymptotically yields Bayes decision boundary
- Binary case:
  A loss $V(f(\mathbf{x}, y))$ is Fisher consistent if:

  The minimizer of $\mathbb{E}[V(f(\mathbf{X}, Y))|\mathbf{X} = \mathbf{x}]$ has the same sign as the Bayes decision $P(Y = 1|\mathbf{X} = \mathbf{x}) - \frac{1}{2}$

[1] Lin, Y. Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6,** 259–275 (2002).

Fisher Consistency in Binary Classification

- Fisher consistency / Bayes Consistency:
  Requires a classifier to asymptotically yields Bayes decision boundary
- Binary case:
  A loss $V(f(\mathbf{x}, y))$ is Fisher consistent if:

  $$\text{The minimizer of } \mathbb{E}[V(f(\mathbf{X}, Y))|\mathbf{X} = \mathbf{x}] \text{ has the same sign as}$$
  $$\text{the Bayes decision } P(Y = 1|\mathbf{X} = \mathbf{x}) - \tfrac{1}{2}$$

- Binary SVM is Fisher consistent[1]
  The minimizer of $\mathbb{E}[[1 - Yf(\mathbf{X})]_+|\mathbf{X} = \mathbf{x}]$ is
  $\text{sign}(P(Y = 1|\mathbf{X} = \mathbf{x}) - \frac{1}{2})$

[1] Lin, Y. Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery* **6,** 259–275 (2002).

Fisher Consistency in Multi-class Classification

- k class. $y \in [1, k]$
- Let: $P_j(\mathbf{x}) = P(Y = j | \mathbf{X} = \mathbf{x})$

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Fisher Consistency in Multi-class Classification

- k class. $y \in [1, k]$
- Let: $P_j(\mathbf{x}) = P(Y = j | \mathbf{X} = \mathbf{x})$
- Potential vectors : $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})]^T$
- Denote: $\mathbf{f}^*(\mathbf{x}) = [f_1^*(\mathbf{x}), \cdots, f_k^*(\mathbf{x})]^T$ is the minimizer of $\mathbb{E}[V(f(\mathbf{X}, Y))|\mathbf{X} = \mathbf{x}]$

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Fisher Consistency in Multi-class Classification

- k class. $y \in [1, k]$
- Let: $P_j(\mathbf{x}) = P(Y = j | \mathbf{X} = \mathbf{x})$
- Potential vectors : $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})]^T$
- Denote: $\mathbf{f}^*(\mathbf{x}) = [f_1^*(\mathbf{x}), \cdots, f_k^*(\mathbf{x})]^T$ is the minimizer of $\mathbb{E}[V(f(\mathbf{X}, Y))|\mathbf{X} = \mathbf{x}]$
- Fisher consistency requires:

$$\underset{j}{\operatorname{argmax}} f_j^*(\mathbf{x}) = \underset{j}{\operatorname{argmax}} P_j(\mathbf{x})$$

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Fisher Consistency in Multi-class Classification

- k class. $y \in [1, k]$
- Let: $P_j(\mathbf{x}) = P(Y = j | \mathbf{X} = \mathbf{x})$
- Potential vectors : $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})]^T$
- Denote: $\mathbf{f}^*(\mathbf{x}) = [f_1^*(\mathbf{x}), \cdots, f_k^*(\mathbf{x})]^T$ is the minimizer of $\mathbb{E}[V(f(\mathbf{X}, Y))|\mathbf{X} = \mathbf{x}]$
- Fisher consistency requires:

$$\underset{j}{\operatorname{argmax}}\, f_j^*(\mathbf{x}) = \underset{j}{\operatorname{argmax}}\, P_j(\mathbf{x})$$

- Remove redundant solutions:
  Employ the constraint: $\sum_{i=1}^{k} f_j(\mathbf{x}) = 0$

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

All-in-One Machines

Simplify the losses for analysis: change the constants to 1

1. LLW loss:
$$V_{\text{LLW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 + f_j(\mathbf{x})]_+$$
2. WW loss:
$$V_{\text{WW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 - (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$
3. CS loss:
$$V_{\text{CS}}(f(\mathbf{X}, Y)) = [1 - \min_j (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$
4. Naive loss:
$$V_{\text{Naive}}(f(\mathbf{X}, Y)) = [1 - f_y(\mathbf{x})]_+$$

WW and CS: Relative potential differences

LLW and Naive: Absolute potential values

Fisher Consistency of the All-in-One Machines SVM

A. Fisher Consistency of the All-in-One Machines SVM
   1. Inconsistency of the Naive Formulation
   2. Consistency of the LLW Formulation
   3. Inconsistency of the WW Formulation
   4. Inconsistency of the CS Formulation

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Fisher Consistency of the All-in-One Machines SVM

A. Fisher Consistency of the All-in-One Machines SVM
   1. Inconsistency of the Naive Formulation
   2. Consistency of the LLW Formulation
   3. Inconsistency of the WW Formulation
   4. Inconsistency of the CS Formulation

B. Modification of the Inconsistent Formulations
   1. Modification of the Naive Formulation
   2. Modification of the WW Formulation
   3. Modification of the CS Formulation

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Inconsistency of the Naive Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{Naive}}(f(\mathbf{X}, Y))] = \mathbb{E}[[1 - f_Y(\mathbf{x})]_+]$ is equal to minimizing $\sum_{l=1}^{k} P_l(\mathbf{x})([1 - f_l(\mathbf{x})]_+)$

Inconsistency of the Naive Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{Naive}}(f(\mathbf{X}, Y))] = \mathbb{E}[[1 - f_Y(\mathbf{x})]_+]$ is equal to minimizing $\sum_{l=1}^{k} P_l(\mathbf{x})([1 - f_l(\mathbf{x})]_+)$
- We want to find properties of the minimizer $\mathbf{f}^*$

Lemma 1.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[[1 - f_Y(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} P_l(\mathbf{x})([1 - f_l(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = -(k-1)$ if $j = \text{argmin}_j P_j(\mathbf{x})$ and 1 otherwise.

Lemma 1.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[[1 - f_Y(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} P_l(\mathbf{x})([1 - f_l(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = -(k-1)$ if $j = \text{argmin}_j P_j(\mathbf{x})$ and 1 otherwise.

- The minimization can be reduced to: (proof omitted)

$$\begin{aligned} \max_{\mathbf{f}} \quad & \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x}) \\ \text{subject to:} \quad & \sum_{l=1}^{k} f_l(\mathbf{x}) = 0 \\ & f_j(\mathbf{x}) \leq 1, \forall l \in [1, k] \end{aligned}$$

Lemma 1.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[[1 - f_Y(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} P_l(\mathbf{x})([1 - f_l(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = -(k-1)$ if $j = \text{argmin}_j P_j(\mathbf{x})$ and 1 otherwise.

- The minimization can be reduced to: (proof omitted)

$$\begin{aligned} \max_{\mathbf{f}} \quad & \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x}) \\ \text{subject to:} \quad & \sum_{l=1}^{k} f_l(\mathbf{x}) = 0 \\ & f_j(\mathbf{x}) \leq 1, \forall l \in [1, k] \end{aligned}$$

- The solution for the maximization above:
  satisfies $f_j^*(\mathbf{x}) = -(k-1)$ if $j = \text{argmin}_j P_j(\mathbf{x})$ and 1 otherwise
- The Naive hinge loss formulation is not Fisher consistent

Consistency of the LLW Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{LLW}}(f(\mathbf{X}, Y))] = \mathbb{E}[\sum_{j \neq Y}[1 + f_j(\mathbf{X})]_+]$ is equal to minimizing $\sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 + f_j(\mathbf{x})]_+)$

Consistency of the LLW Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{LLW}}(f(\mathbf{X}, Y))] = \mathbb{E}[\sum_{j \neq Y}[1 + f_j(\mathbf{X})]_+]$ is equal to minimizing $\sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 + f_j(\mathbf{x})]_+)$
- We want to find properties of the minimizer $\mathbf{f}^*$

Lemma 2.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[\sum_{j \neq Y}[1 + f_j(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 + f_j(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = k - 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and -1 otherwise.

Lemma 2.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[\sum_{j \neq Y}[1 + f_j(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 + f_j(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = k - 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and -1 otherwise.

Proof

- The minimization can be reduced to: (proof omitted)

$$\begin{aligned} \max_{\mathbf{f}} \quad & \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x}) \\ \text{subject to:} \quad & \sum_{l=1}^{k} f_l(\mathbf{x}) = 0 \\ & f_l(\mathbf{x}) \geq -1, \forall l \in [1, k] \end{aligned}$$

Lemma 2.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[\sum_{j \neq Y}[1 + f_j(\mathbf{X})]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 + f_j(\mathbf{x})]_+)$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following: $f_j^*(\mathbf{x}) = k - 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and -1 otherwise.

Proof

- The minimization can be reduced to: (proof omitted)

$$\begin{aligned} \max_{\mathbf{f}} \quad & \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x}) \\ \text{subject to:} \quad & \sum_{l=1}^{k} f_l(\mathbf{x}) = 0 \\ & f_l(\mathbf{x}) \geq -1, \forall l \in [1, k] \end{aligned}$$

  - The solution for the maximization above: satisfies $f_j^*(\mathbf{x}) = k - 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and -1 otherwise
- The LLW formulation is Fisher consistent

Inconsistency of the WW Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{WW}}(f(\mathbf{X}, Y))] = \mathbb{E}[\sum_{j \neq y} [1 - (f_Y(\mathbf{x}) - f_j(\mathbf{x}))]_+]$ is equal to minimizing $\sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 - (f_l(\mathbf{x}) - f_j(\mathbf{x}))]_+)$

Inconsistency of the WW Formulation

- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{WW}}(f(\mathbf{X}, Y))] = \mathbb{E}[\sum_{j \neq y} [1 - (f_Y(\mathbf{x}) - f_j(\mathbf{x}))]_+]$ is equal to minimizing $\sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 - (f_l(\mathbf{x}) - f_j(\mathbf{x}))]_+)$
- We focus on the case where $k = 3$, and find the minimizer $\mathbf{f}^*$

Lemma 3.

Consider the case where $k = 3$ with $\frac{1}{2} > P_1 > P_2 > P_3$. The minimizer $\mathbf{f}^* = (f_1^*, f_2^*, f_3^*)$ of $\mathbb{E}[\sum_{j \neq y} [1 - (f_Y(\mathbf{X}) - f_j(\mathbf{X}))]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 - (f_l(\mathbf{x}) - f_j(\mathbf{x}))]_+)$ is the following:

(1) If $P_2 = \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$ and $f_1^* - f_3^* = 1$.

(2) If $P_2 > \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$, $f_1^* = f_2^*$ and $f_2^* - f_3^* = 1$.

(3) If $P_2 < \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$, $f_2^* = f_3^*$ and $f_1^* - f_2^* = 1$.

Lemma 3.

Consider the case where $k = 3$ with $\frac{1}{2} > P_1 > P_2 > P_3$. The minimizer $\mathbf{f}^* = (f_1^*, f_2^*, f_3^*)$ of $\mathbb{E}[\sum_{j \neq y} [1 - (f_Y(\mathbf{X}) - f_j(\mathbf{X}))]_+ | \mathbf{X} = \mathbf{x}] = \sum_{l=1}^{k} \sum_{j \neq l} P_l(\mathbf{x})([1 - (f_l(\mathbf{x}) - f_j(\mathbf{x}))]_+)$ is the following:

(1) If $P_2 = \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$ and $f_1^* - f_3^* = 1$.

(2) If $P_2 > \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$, $f_1^* = f_2^*$ and $f_2^* - f_3^* = 1$.

(3) If $P_2 < \frac{1}{3}$, any $\mathbf{f}^*$ satisfying $f_1^* \geq f_2^* \geq f_3^*$, $f_2^* = f_3^*$ and $f_1^* - f_2^* = 1$.

From Lemma 3:

- In the case of $k = 3$ with $\frac{1}{2} > P_1 > P_2 > P_3$
- The WW formulation is Fisher consistent only when $P_2 < \frac{1}{3}$

Inconsistency of the CS Formulation

- Denote $\mathbf{g}(\mathbf{f}(\mathbf{x}), y) = \{f_y(\mathbf{x}) - f_j(\mathbf{x});\, j \neq y\}$
  The CS loss can be rewritten as: $[1 - \min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)]_+$
- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\mathrm{CS}}(f(\mathbf{X}, Y))] = \mathbb{E}[[1 - \min_j (f_Y(\mathbf{X}) - f_j(\mathbf{X}))]_+]$ is equal to minimizing $\sum_{l=1}^{k} P_l(\mathbf{x})([1 - \min \mathbf{g}(\mathbf{f}(\mathbf{x}), l)]_+)$

Inconsistency of the CS Formulation

- Denote $\mathbf{g}(\mathbf{f}(\mathbf{x}), y) = \{f_y(\mathbf{x}) - f_j(\mathbf{x}); j \neq y\}$
  The CS loss can be rewritten as: $[1 - \min \mathbf{g}(\mathbf{f}(\mathbf{x}), y)]_+$
- For any fixed $\mathbf{X} = \mathbf{x}$:
  Minimizing $\mathbb{E}[V_{\text{CS}}(f(\mathbf{X}, Y))] = \mathbb{E}[[1 - \min_j (f_Y(\mathbf{X}) - f_j(\mathbf{X}))]_+]$ is equal to minimizing $\sum_{l=1}^{k} P_l(\mathbf{x})([1 - \min \mathbf{g}(\mathbf{f}(\mathbf{x}), l)]_+)$
- We want to find properties of the minimizer $\mathbf{f}^*$

Lemma 4.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[1 - \min_j (f_Y(\mathbf{X}) - f_j(\mathbf{X}))_+ | \mathbf{X} = \mathbf{x}]$ subject to $\sum_{j=1}^{k} f_j(\mathbf{x}) = 0$ satisfies the following properties:

(1) If $\max_j P_j > \frac{1}{2}$, then $\operatorname{argmax}_j f_j^* = \operatorname{argmax}_j P_j$ and $\min \mathbf{g}^*(\mathbf{f}(\mathbf{x}), \operatorname{argmax}_j f_j^*) = 1$.

(2) If $\max_j P_j < \frac{1}{2}$, then $\mathbf{f}^* = \mathbf{0}$

Lemma 4.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[1 - \min_j (f_Y(\mathbf{X}) - f_j(\mathbf{X}))_+ | \mathbf{X} = \mathbf{x}]$ subject to $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ satisfies the following properties:
(1) If $\max_j P_j > \frac{1}{2}$, then $\operatorname{argmax}_j f_j^* = \operatorname{argmax}_j P_j$ and $\min \mathbf{g}^*(\mathbf{f}(\mathbf{x}), \operatorname{argmax}_j f_j^*) = 1$.
(2) If $\max_j P_j < \frac{1}{2}$, then $\mathbf{f}^* = \mathbf{0}$

From Lemma 4:

- For the problem with $k > 2$, the existence of a domination class ($P_j > \frac{1}{2}$) cannot be guaranteed
- If $\max_j P_j < \frac{1}{2}$ for a given $\mathbf{x}$, then $\mathbf{f}^*(\mathbf{x}) = \mathbf{0}$
  In this case $\operatorname{argmax}_j f_j(\mathbf{x})$ cannot uniquely determined
- The CS formulation is Fisher consistent only when there is a domination class

Modification of the Inconsistent Formulations

B. Modification of the Inconsistent Formulations

1. Modification of the Naive Formulation
2. Modification of the WW Formulation
3. Modification of the CS Formulation

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Modification of the Naive Formulation

Reduced problem in the Naive Formula (Inconsistent Loss)

$$\max_{\mathbf{f}} \quad \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x})$$

$$\text{subject to:} \quad \sum_{l=1}^{k} f_l(\mathbf{x}) = 0, \quad f_l(\mathbf{x}) \leq 1, \quad \forall l \in [1, k]$$

Reduced problem in the LLW Formula (Consistent Loss)

$$\max_{\mathbf{f}} \quad \sum_{l=1}^{k} P_l(\mathbf{x}) f_l(\mathbf{x})$$

$$\text{subject to:} \quad \sum_{l=1}^{k} f_l(\mathbf{x}) = 0, \quad f_l(\mathbf{x}) \geq -1, \quad \forall l \in [1, k]$$

$\rightarrow$ The only difference is the constraint for $f_l(\mathbf{x})$

Modification of the Naive Formulation

- If we add an additional constraint $f_l(\mathbf{x}) \geq -\frac{1}{k-1}, \forall l \in [1, k]$ to the Naive formulation, the minimizer becomes: $f_j^*(\mathbf{x}) = 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and $-\frac{1}{k-1}$ otherwise which indicates consistency.

Modification of the Naive Formulation

- If we add an additional constraint $f_l(\mathbf{x}) \geq -\frac{1}{k-1}, \forall l \in [1, k]$ to the Naive formulation, the minimizer becomes: $f_j^*(\mathbf{x}) = 1$ if $j = \text{argmax}_j P_j(\mathbf{x})$ and $-\frac{1}{k-1}$ otherwise which indicates consistency.
- By rescaling the constant, we get the following consistent loss:

$$V_{\text{Consistent-Naive}}(f(\mathbf{X}, Y)) = \quad [k - 1 - f_y(\mathbf{x})]_+$$

$$\text{subject to:} \quad \sum_{j=1}^{k} f_j(\mathbf{x}) = 0; \quad f_l(\mathbf{x}) \geq -1, \forall l \in [1, k]$$

Modification of the WW Formulation

- Note that the WW loss:

$$V_{\mathrm{WW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 - (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$

- Add a new constraint $-1 \leq f_j(\mathbf{x}) \leq k - 1$, change the constant part, the loss reduces to:

$$\begin{aligned} V(f(\mathbf{X}, Y)) = \quad & k[k - 1 - f_y(\mathbf{x})]_+ \\ \text{subject to:} \quad & \sum_{j=1}^{k} f_j(\mathbf{x}) = 0; \quad f_l(\mathbf{x}) \geq -1, \, \forall l \in [1, k] \end{aligned}$$

- The loss is equivalent to the Consistent-Naive formulation. Therefore it is Fisher consistent.

Modification of the WW Formulation : Optimization

- The constraint $-1 \leq f_j(\mathbf{x}) \leq k-1, \forall j$ can be difficult to achieve for all possible $\mathbf{x}$ in the feature spaces
- It is suggested that we need to restrict the constraint to the training data points only.

$$\min_{\mathbf{f}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{f}_j\|^2 + C\sum_{i=1}^{n} f_{y_i}(\mathbf{x}_i)$$

$$\text{subject to:} \quad \sum_{j=1}^{k} f_j(\mathbf{x}_i) = 0;\ f_j(\mathbf{x}) \geq -1;\ \forall l \in [1,k],\ i \in [1,n].$$

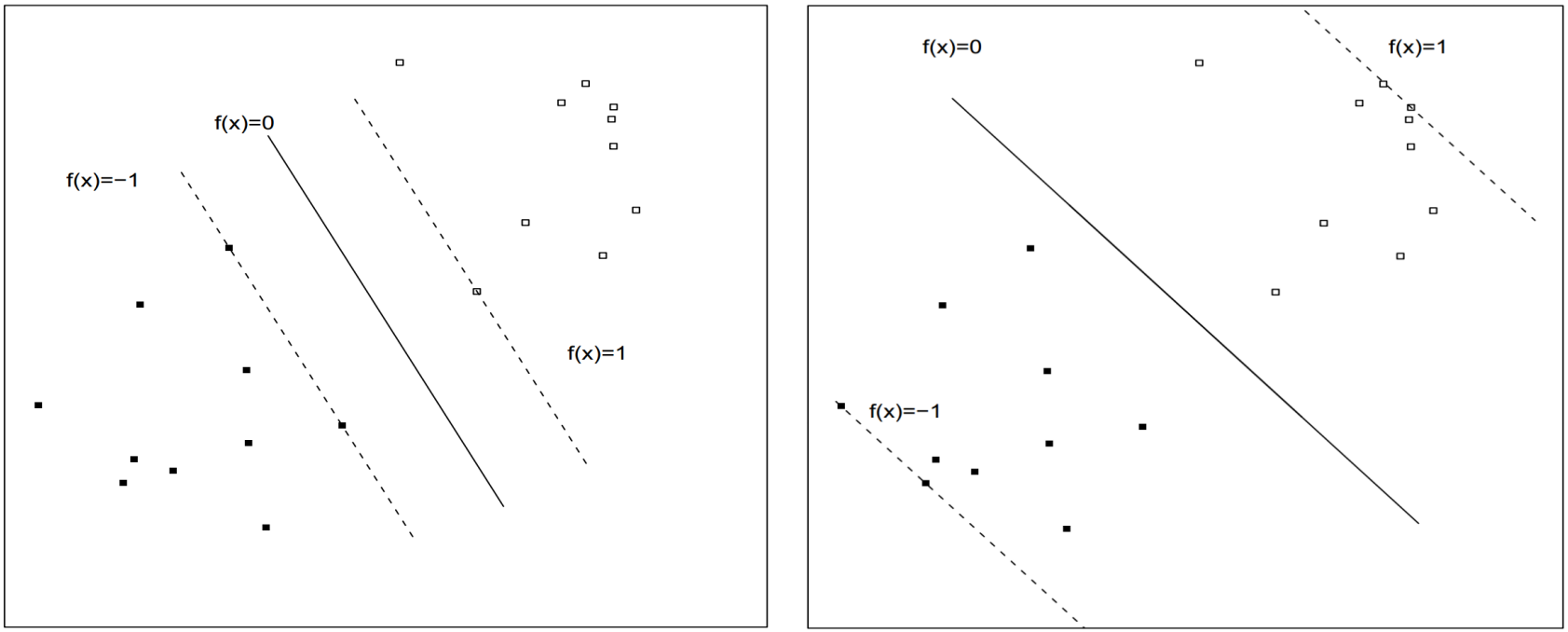
Modification of the WW Formulation : Optimization

- The constraint $-1 \leq f_j(\mathbf{x}) \leq k-1, \forall j$ can be difficult to achieve for all possible $\mathbf{x}$ in the feature spaces
- It is suggested that we need to restrict the constraint to the training data points only.

$$\min_{\mathbf{f}} \quad \frac{1}{2}\sum_{j=1}^{k}\|\mathbf{f}_j\|^2 + C\sum_{i=1}^{n} f_{y_i}(\mathbf{x}_i)$$

$$\text{subject to:} \quad \sum_{j=1}^{k} f_j(\mathbf{x}_i) = 0;\ f_j(\mathbf{x}) \geq -1;\ \forall l \in [1,k],\ i \in [1,n].$$

- To better understand the formulation above, we analyze the binary case version ($y \in \{\pm 1\}$)

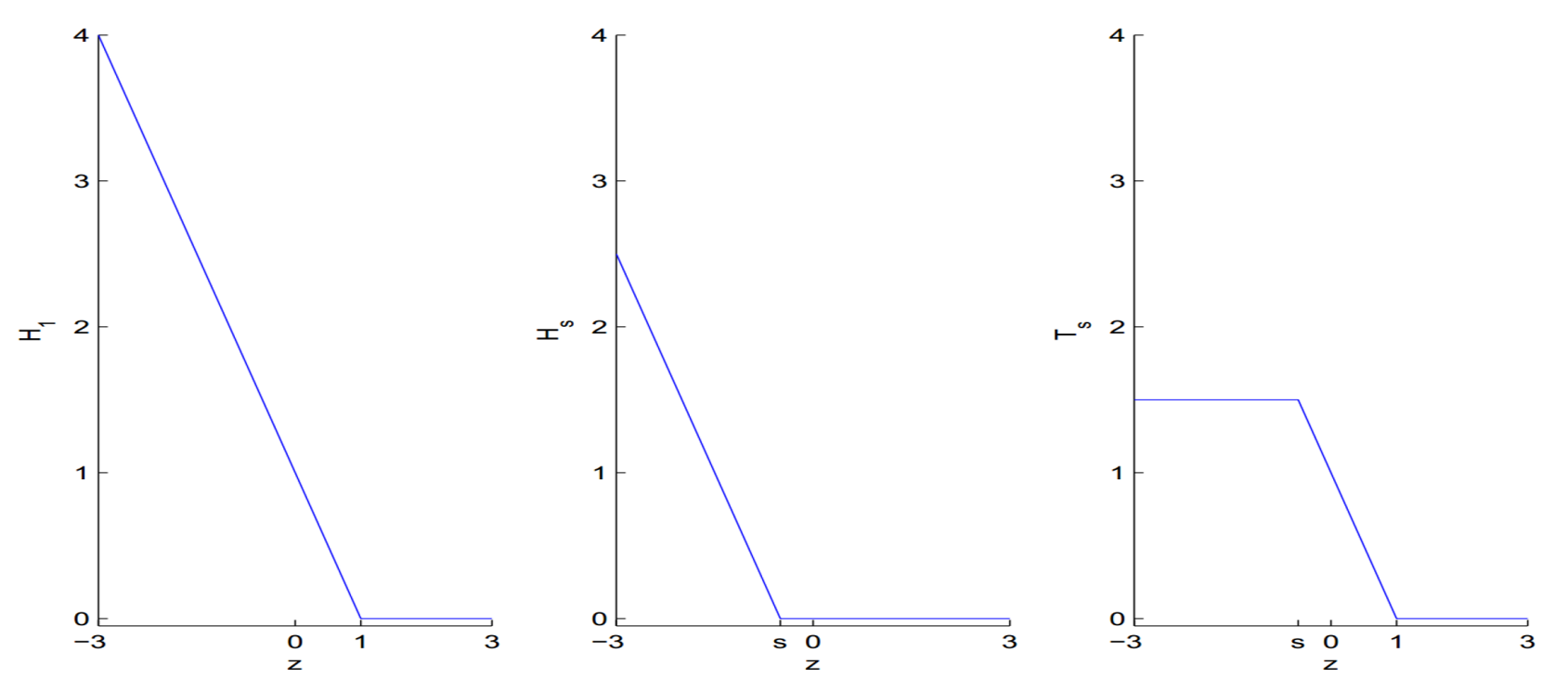


An example of standard binary SVM solution (left) and modified WW formulation solution (right) in a two dimensional dataset.

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Modification of the CS Formulation

- The CS formulation cannot easily modified by adding a bounded constraint as in the WW formulation
- We explore the idea of truncating the hinge loss



Function plot of $H_1(u)$ (left), $H_s(u)$ (middle), and $T_s(u)$ (right)

Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Modification of the CS Formulation

- For any $s \leq 0$, it can be proven that the truncated version of the CS formulation is Fisher consistent, even in the case there is no dominating class

Experiments

Experiments

A. Artificial Benchmark Problem
   1. Artificial Benchmark Setup
   2. Benchmark Result

Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Experiments

A. Artificial Benchmark Problem
   1. Artificial Benchmark Setup
   2. Benchmark Result

B. Empirical Comparison
   1. Experiment Setup
   2. Experiment Result

Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Artificial Benchmark Setup

- Help understand when and why some formulations deliver substantially sub-optimal solutions

Artificial Benchmark Setup

- Help understand when and why some formulations deliver substantially sub-optimal solutions
- Domain: $X = S^1 = \{x \in \mathbb{R}^2 \mid \|x\| = 1\} \rightarrow$ unit circle
- Circle is parameterized using:
  $\beta(t) = (\cos(t \cdot \frac{\pi}{10}), \sin(t \cdot \frac{\pi}{10}))$ where $t \in [0, 20]$

Artificial Benchmark Setup

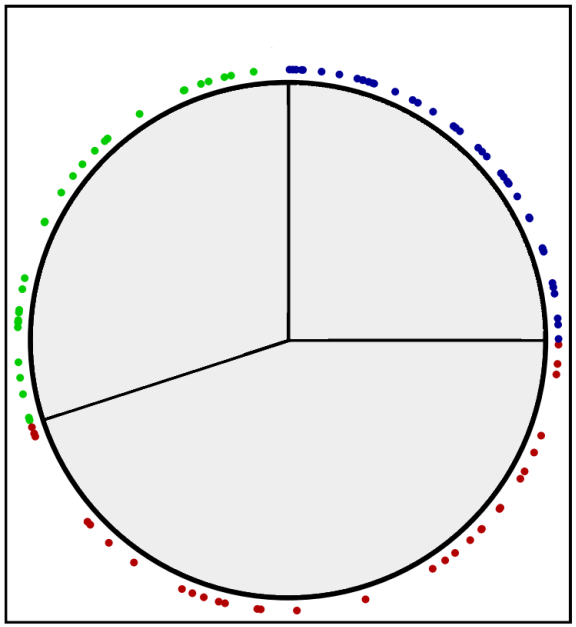
- Help understand when and why some formulations deliver substantially sub-optimal solutions
- Domain: $X = S^1 = \{x \in \mathbb{R}^2 \mid \|x\| = 1\} \rightarrow$ unit circle
- Circle is parameterized using:
  $\beta(t) = (\cos(t \cdot \frac{\pi}{10}), \sin(t \cdot \frac{\pi}{10}))$ where $t \in [0, 20]$
- 3 classes classification, $Y = \{1, 2, 3\}$

Artificial Benchmark Setup

- Noise-less problem
  - The label y is drawn uniformly from Y
  - Then x is drawn uniformly at random from sector X_y

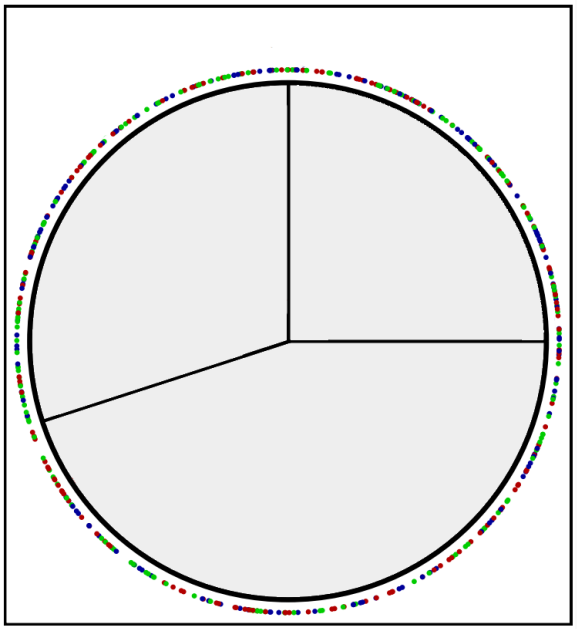
    Sectors: $X_1 = \beta([0, 5))$, $X_2 = \beta([5, 11))$, and $X_3 = \beta([11, 20))$
- Bayes-optimal prediction:

  Predict label y on sector X_y



Artificial Benchmark Setup

- Noisy problem
  - The same step as in the noise-less problem
  - Reassign 90% of the labels uniformly at random
  - Therefore, the distribution of X is remain unchanged
    The conditional distributions of the label given a x point are changed:
    Conditioned on $x \in X_z$, the event of $y = z$ has probability 40%, while the other two cases have probability of 30%
- Bayes-optimal prediction:
  Predict label y on sector X_y



Artificial Benchmark Result

Multi-class SVM Loss Review:

1. LLW loss:
$$V_{\text{LLW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 + f_j(\mathbf{x})]_+$$
2. WW loss:
$$V_{\text{WW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 - (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$
3. CS loss:
$$V_{\text{CS}}(f(\mathbf{X}, Y)) = [1 - \min_j (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$

WW and CS: Relative potential differences, i.e. $(f_y(\mathbf{x}) - f_j(\mathbf{x}))$
LLW: Absolute potential values, i.e. $f_j(\mathbf{x})$

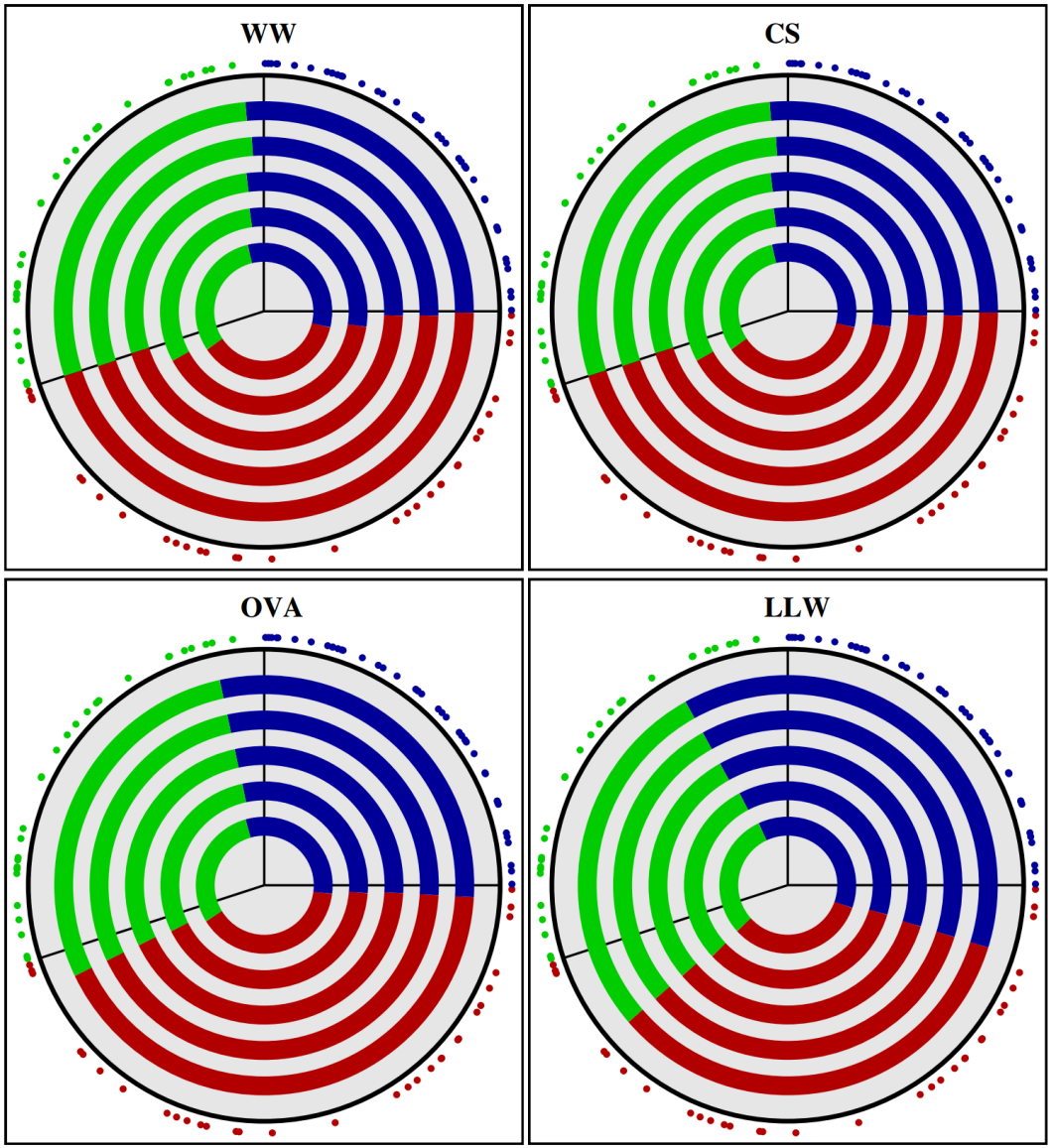
Artificial Benchmark Result

Multi-class SVM Loss Review:

1. LLW loss:
$$V_{\text{LLW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 + f_j(\mathbf{x})]_+$$
2. WW loss:
$$V_{\text{WW}}(f(\mathbf{X}, Y)) = \sum_{j \neq y} [1 - (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$
3. CS loss:
$$V_{\text{CS}}(f(\mathbf{X}, Y)) = [1 - \min_j (f_y(\mathbf{x}) - f_j(\mathbf{x}))]_+$$

WW and CS: Relative potential differences, i.e. $(f_y(\mathbf{x}) - f_j(\mathbf{x}))$
LLW: Absolute potential values, i.e. $f_j(\mathbf{x})$

OVA: k binary classifiers, the loss in each classifier depends on the potential $f_j(\mathbf{x})$. Therefore, the loss for OVA can be viewed as the summation over absolute potential value losses.



Noise-less problem

Sector separators:
Bayes-optimal predictor.

Colors:
Blue = Class 1
Green = Class 2
Red = Class 3

Points outside the circle:
100 training samples.

Colored circles:
Classifier prediction for $C = 10^n, n \in \{0, 1, 2, 3, 4\}$, from inner to outer circles

---

Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Noise-less problem results

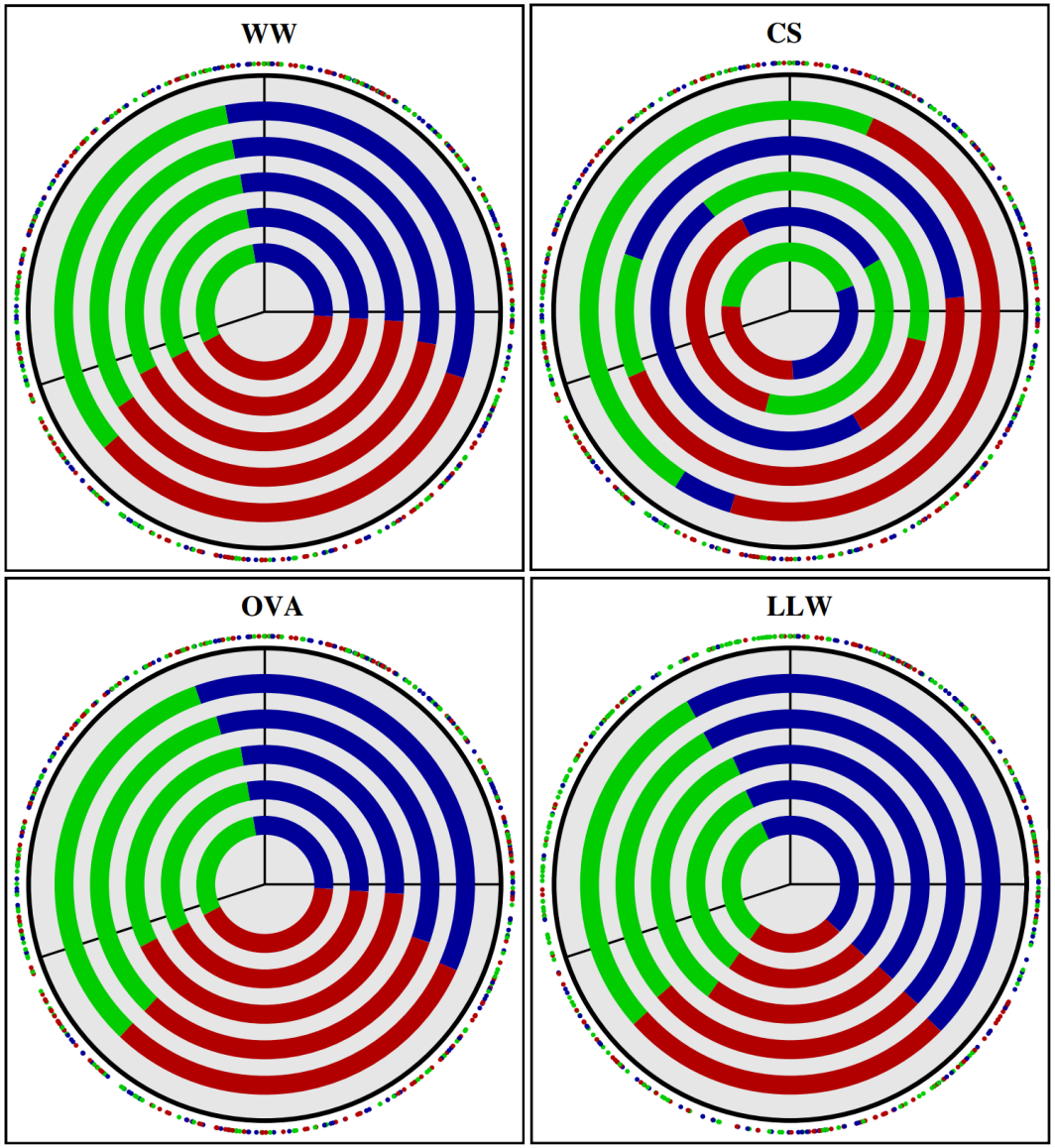
- Sub-optimal solution of absolute potential values losses (LLW and OVA)
  - Both the LLW and OVA formulations give sub-optimal solutions

Noise-less problem results

- Sub-optimal solution of absolute potential values losses (LLW and OVA)
  - Both the LLW and OVA formulations give sub-optimal solutions
  - Fisher consistency property of the LLW formulation does not help

Noise-less problem results

- Sub-optimal solution of absolute potential values losses (LLW and OVA)
  - Both the LLW and OVA formulations give sub-optimal solutions
  - Fisher consistency property of the LLW formulation does not help
  - Dogan claimed that the sub-optimal solutions are caused by the absolute potential values used in the loss construction, which are not compatible with the form of the decision function.



Noisy problem

Sector separators:
Bayes-optimal predictor.

Colors:
Blue = Class 1
Green = Class 2
Red = Class 3

Points outside the circle:
500 training samples.

Colored circles:
Classifier prediction for $C = 10^n, n \in \{-4, -3, -2, -1, 0\}$, from inner to outer circles

---

Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Review of Lemma 4. in the CS Formulation

Lemma 4.

The minimizer $\mathbf{f}^*$ of $\mathbb{E}[1 - \min_j (f_Y(\mathbf{X}) - f_j(\mathbf{X}))_+ | \mathbf{X} = \mathbf{x}]$ subject to $\sum_{j=1}^k f_j(\mathbf{x}) = 0$ satisfies the following properties:

(1) If $\max_j P_j > \frac{1}{2}$, then $\operatorname{argmax}_j f_j^* = \operatorname{argmax}_j P_j$ and $\min \mathbf{g}^*(\mathbf{f}(\mathbf{x}), \operatorname{argmax}_j f_j^*) = 1$.

(2) If $\max_j P_j < \frac{1}{2}$, then $\mathbf{f}^* = \mathbf{0}$

Experiment Setup

- 17 datasets from UCI ML repository and `libsvm`'s collection

Experiment Setup

- 17 datasets from UCI ML repository and `libsvm`'s collection
- Data pre-processing:
  Rescale to unit variance (based on the training statistics)
- Model Selection (for selecting C):
  Five-folds cross-validation, repeated ten times

Experiment Setup

- 17 datasets from UCI ML repository and `libsvm`'s collection
- Data pre-processing:
  Rescale to unit variance (based on the training statistics)
- Model Selection (for selecting C):
  Five-folds cross-validation, repeated ten times
- Evaluation:
  - 100 different random splits of training and testing data
  - The setup yields 100 different testing accuracies
  - Paired U-tests at significance level 0.01

Datasets

Dataset	Number of classes	Training data	Testing data
Covertype	7	406708	174304
Letter	26	14000	6000
News-20	20	14000	6000
Sector	105	6412	3207
Usps	10	7291	2007
Abalone	27	3133	1044
Car	4	1209	519
Glass	6	149	65
Iris	3	105	45
Opt. Digits	10	3823	1797
Page Blocks	5	3831	1642
Sat	7	4435	2000
Segment	7	1617	693
Soy Bean	19	214	93
Vehicle	4	592	254
Red wine	10	1119	480
White wine	10	3429	1469

Empirical Result

Dataset	OVA	WW	CS	LLW
Covertype	50.59 (±5.49)	**70.55** (±0.09)	45.73 (±5.88)	21.87 (±23.19)
Letter	63.69 (±0.48)	69.39 (±0.63)	**76.59** (±0.61)	12.78 (±0.40)
News-20	85.36 (±0.32)	85.13 (±0.15)	85.17 (±0.32)	**86.71** (±0.39)
Sector	94.53 (±0.22)	94.10 (±0.33)	94.80 (±0.29)	94.82 (±0.28)
Usps	94.50 (±0.39)	94.46 (±0.57)	**95.26** (±0.46)	78.18 (±5.27)
Abalone	18.95 (±0.86)	**21.70** (±1.30)	14.12 (±1.64)	16.56 (±1.17)
Car	71.69 (±1.73)	**73.76** (±1.68)	**73.15** (±2.02)	65.34 (±12.17)
Glass	**56.98** (±6.44)	**61.93** (±6.63)	**61.93** (±6.04)	46.78 (±6.77)
Iris	91.11 (±4.85)	**95.88** (±1.71)	**91.76** (±7.18)	74.65 (±7.52)
Opt. Digits	95.98 (±0.60)	96.03 (±0.37)	**96.42** (±0.37)	73.56 (±2.11)
Page Blocks	70.44 (±21.20)	91.14 (±5.41)	**94.20** (±2.34)	93.22 (±1.02)
Sat	75.04 (±0.96)	**77.40** (±3.00)	66.87 (±9.90)	51.47 (±9.01)
Segment	92.54 (±0.75)	**92.43** (±2.13)	**92.43** (±2.13)	74.50 (±1.32)
Soy Bean	**90.65** (±3.03)	87.75 (±3.16)	83.49 (±5.80)	77.95 (±9.97)
Vehicle	52.02 (±11.98)	72.75 (±4.13)	72.75 (±4.13)	63.21 (±10.63)
Red wine	53.38 (±2.63)	**58.37** (±1.69)	55.61 (±2.47)	57.26 (±2.02)
White wine	50.73 (±1.27)	**51.78** (±1.24)	50.85 (±1.12)	46.44 (±1.74)

Accuracies and standard deviations for each dataset.

Highlighted numbers: the best model and other models that is not significantly worse than the best one using paired U-test with $\alpha = 0.01$

Empirical Result

Dataset	OVA	WW	CS	LLW
Covertype	50.59 (±5.49)	**70.55** (±0.09)	45.73 (±5.88)	21.87 (±23.19)
Letter	63.69 (±0.48)	69.39 (±0.63)	**76.59** (±0.61)	12.78 (±0.40)
News-20	85.36 (±0.32)	85.13 (±0.15)	85.17 (±0.32)	**86.71** (±0.39)
Sector	94.53 (±0.22)	94.10 (±0.33)	94.80 (±0.29)	94.82 (±0.28)
Usps	94.50 (±0.39)	94.46 (±0.57)	**95.26** (±0.46)	78.18 (±5.27)
Abalone	18.95 (±0.86)	**21.70** (±1.30)	14.12 (±1.64)	16.56 (±1.17)
Car	71.69 (±1.73)	**73.76** (±1.68)	**73.15** (±2.02)	65.34 (±12.17)
Glass	**56.98** (±6.44)	**61.93** (±6.63)	**61.93** (±6.04)	46.78 (±6.77)
Iris	91.11 (±4.85)	**95.88** (±1.71)	**91.76** (±7.18)	74.65 (±7.52)
Opt. Digits	95.98 (±0.60)	96.03 (±0.37)	**96.42** (±0.37)	73.56 (±2.11)
Page Blocks	70.44 (±21.20)	91.14 (±5.41)	**94.20** (±2.34)	93.22 (±1.02)
Sat	75.04 (±0.96)	**77.40** (±3.00)	66.87 (±9.90)	51.47 (±9.01)
Segment	92.54 (±0.75)	**92.43** (±2.13)	**92.43** (±2.13)	74.50 (±1.32)
Soy Bean	**90.65** (±3.03)	87.75 (±3.16)	83.49 (±5.80)	77.95 (±9.97)
Vehicle	52.02 (±11.98)	72.75 (±4.13)	72.75 (±4.13)	63.21 (±10.63)
Red wine	53.38 (±2.63)	**58.37** (±1.69)	55.61 (±2.47)	57.26 (±2.02)
White wine	50.73 (±1.27)	**51.78** (±1.24)	50.85 (±1.12)	46.44 (±1.74)

WW : highlighted 9 times
CS : highlighted 8 times

Empirical Result

Dataset	OVA	WW	CS	LLW
Covertype	50.59 (±5.49)	**70.55** (±0.09)	45.73 (±5.88)	21.87 (±23.19)
Letter	63.69 (±0.48)	69.39 (±0.63)	**76.59** (±0.61)	12.78 (±0.40)
News-20	85.36 (±0.32)	85.13 (±0.15)	85.17 (±0.32)	**86.71** (±0.39)
Sector	94.53 (±0.22)	94.10 (±0.33)	94.80 (±0.29)	94.82 (±0.28)
Usps	94.50 (±0.39)	94.46 (±0.57)	**95.26** (±0.46)	78.18 (±5.27)
Abalone	18.95 (±0.86)	**21.70** (±1.30)	14.12 (±1.64)	16.56 (±1.17)
Car	71.69 (±1.73)	**73.76** (±1.68)	**73.15** (±2.02)	65.34 (±12.17)
Glass	**56.98** (±6.44)	**61.93** (±6.63)	**61.93** (±6.04)	46.78 (±6.77)
Iris	91.11 (±4.85)	**95.88** (±1.71)	**91.76** (±7.18)	74.65 (±7.52)
Opt. Digits	95.98 (±0.60)	96.03 (±0.37)	**96.42** (±0.37)	73.56 (±2.11)
Page Blocks	70.44 (±21.20)	91.14 (±5.41)	**94.20** (±2.34)	93.22 (±1.02)
Sat	75.04 (±0.96)	**77.40** (±3.00)	66.87 (±9.90)	51.47 (±9.01)
Segment	92.54 (±0.75)	**92.43** (±2.13)	**92.43** (±2.13)	74.50 (±1.32)
Soy Bean	**90.65** (±3.03)	87.75 (±3.16)	83.49 (±5.80)	77.95 (±9.97)
Vehicle	52.02 (±11.98)	72.75 (±4.13)	72.75 (±4.13)	63.21 (±10.63)
Red wine	53.38 (±2.63)	**58.37** (±1.69)	55.61 (±2.47)	57.26 (±2.02)
White wine	50.73 (±1.27)	**51.78** (±1.24)	50.85 (±1.12)	46.44 (±1.74)

"News-20" and "Sector" : high dimensional feature spaces
(62,061 and 55,197 features respectively)
Other datasets : rather low dimensional feature spaces

Conclusions

Conclusions

- We explored the efforts on bringing the success of SVM in binary classification problems into multi-class classification problems

Conclusions

- We explored the efforts on bringing the success of SVM in binary classification problems into multi-class classification problems
- We described formulation of each model in both learning and prediction tasks

Conclusions

- We explored the efforts on bringing the success of SVM in binary classification problems into multi-class classification problems
- We described formulation of each model in both learning and prediction tasks
- We discussed the Fisher consistency properties of the all-in-one machine formulations

Conclusions

- We explored the efforts on bringing the success of SVM in binary classification problems into multi-class classification problems
- We described formulation of each model in both learning and prediction tasks
- We discussed the Fisher consistency properties of the all-in-one machine formulations
- We showed the consistency of the LLW formulation and the inconsistency of the WW and CS formulations

Conclusions

- We studied the modification proposed by Liu[2] to make the WW and CS formulations Fisher consistent

[2] Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Conclusions

- We studied the modification proposed by Liu[2] to make the WW and CS formulations Fisher consistent
- The modifications of the WW formulation:
  - Results in a new classification model which enforce all points to lie inside the classification boundary
  - The model loses the sparsity property

[2] Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Conclusions

- We studied the modification proposed by Liu[2] to make the WW and CS formulations Fisher consistent
- The modifications of the WW formulation:
  - Results in a new classification model which enforce all points to lie inside the classification boundary
  - The model loses the sparsity property
  - Sparsity is a key property in analyzing the SVM's theoretical properties, e.g. analyzing generalization bounds of the model
  - The effect of losing the sparsity to the prediction performance need to be analyzed for the proposed model.

[2] Liu, Y. *Fisher consistency of multicategory support vector machines* in *International Conference on Artificial Intelligence and Statistics* (2007), 291–298.

Conclusions

- The modifications of the CS formulation:
  - Introduce a truncated version of hinge loss
  - The truncated loss version fix the inconsistency of the CS formulation

Conclusions

- The modifications of the CS formulation:
  - Introduce a truncated version of hinge loss
  - The truncated loss version fix the inconsistency of the CS formulation
  - The optimization is no-longer convex
  - The convergence to global optimum cannot be guaranteed

Conclusions

- The modifications of the CS formulation:
  - Introduce a truncated version of hinge loss
  - The truncated loss version fix the inconsistency of the CS formulation
  - The optimization is no-longer convex
  - The convergence to global optimum cannot be guaranteed
  - Local optimum solution may effect the prediction performance

Conclusions

- We discussed the experiment result presented in Dogan's paper[3]

[3] Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Conclusions

- We discussed the experiment result presented in Dogan's paper[3]
- Interesting result of the LLW formulation:
  Although it has the Fisher consistency property, it performs poorly in the data which has low-dimensional feature spaces
- This poor results are confirmed in both by artificial benchmark study and empirical evaluation on real datasets

[3] Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Conclusions

- We discussed the experiment result presented in Dogan's paper[3]
- Interesting result of the LLW formulation:
  Although it has the Fisher consistency property, it performs poorly in the data which has low-dimensional feature spaces
- This poor results are confirmed in both by artificial benchmark study and empirical evaluation on real datasets
- The source of the problem is possibly caused by the construction of the LLW loss which uses the the absolute potential values instead of the relative potential differences.

[3] Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Conclusions

- We discussed the experiment result presented in Dogan's paper[3]
- Interesting result of the LLW formulation:
  Although it has the Fisher consistency property, it performs poorly in the data which has low-dimensional feature spaces
- This poor results are confirmed in both by artificial benchmark study and empirical evaluation on real datasets
- The source of the problem is possibly caused by the construction of the LLW loss which uses the the absolute potential values instead of the relative potential differences.
- Employing kernel trick to the LLW formulation is suggested.

[3] Dogan, U. *et al.* A Unified View on Multi-class Support Vector Classification. *The Journal of Machine Learning Research* (2015).

Conclusions

- The WW and CS models which based on the relative potential differences, perform well in most datasets, with a slight advantages for the WW model.

Conclusions

- The WW and CS models which based on the relative potential differences, perform well in most datasets, with a slight advantages for the WW model.
- Dogan recommends relative potential difference based model for almost all applications.

Conclusions

- The WW and CS models which based on the relative potential differences, perform well in most datasets, with a slight advantages for the WW model.
- Dogan recommends relative potential difference based model for almost all applications.
- The WW formulation is more preferred over the CS formulation for its slightly more stable performance.

Conclusions

- A new research question:
  Is it possible to have a Fisher consistent formulation of multi-class SVM which performs well on low-dimensional feature spaces dataset?

Conclusions

- A new research question:
  Is it possible to have a Fisher consistent formulation of multi-class SVM which performs well on low-dimensional feature spaces dataset?
- The answer might be:
  To construct a Fisher consistent loss which use the relative potential differences rather than on the absolute potential values

Conclusions

- A new research question:
  Is it possible to have a Fisher consistent formulation of multi-class SVM which performs well on low-dimensional feature spaces dataset?
- The answer might be:
  To construct a Fisher consistent loss which use the relative potential differences rather than on the absolute potential values
- A following research needs to be conducted

Thank You!