NYU

Multiclass Classification

DS-GA 1003: Machine Learning

Instructors: Samuel Deng and Nicholas Tomlin

Lab 7 presented by Karine

Tuesday, March 26, 2026

Agenda

- Introduction to Multiclass Classification
- One-vs-All (a.k.a. One-vs-Rest)
- Linear Multiclass Classification
- Optional: Extra Slides

Introduction

- This presentation is about multiclass classification (specifically linear).
- Multiclass classification: input space: $\mathcal{X}$, output space $\mathcal{Y}$.
  - $\mathcal{Y} = \{1, \ldots, k\}$ (classes) or $\mathcal{Y} = \mathbb{R}$ (scores)

NYU

Slides based on David Rosenberg's course materials:
https://github.com/davidrosenberg/mlcourse

Multiclass Classifications

- Today, we consider **Linear Multiclass Classification**.

Other ways (covered in this course):

- Logistic Regression can be extended to Categorical (using Softmax)
- Perceptron = simple 1 linear classifier
- Decision Tree
- Linear Multiclass SVM (see extra slides at the end of this presentation)

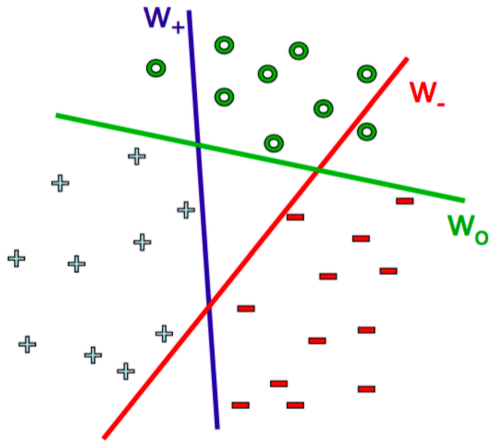
One-vs-All (a.k.a. One-vs-Rest)
=
Reduction to
k Binary Classifications

One-vs-All (a.k.a. One-vs-Rest)

We consider k classification problems. Each is a binary classification.

Example: $k = 3$

3 classes: +, -, o



One-vs-All

- Train k classifiers, one for each class $i \in \{1, \ldots, k\}$.
- Train ith classifier to distinguish class i from the rest.
- Suppose we have: $h_1, h_2, h_3, \ldots, h_k$ (k binary classifiers).
  - Can output *{-1, 1}* where *1* for class i and *-1* for other class.
  - Can output scores (the larger the better, confidence that class is *i*).
- Final prediction is: $i^* = \arg \max_{i \in \{1,\ldots,k\}} h_i(x)$.

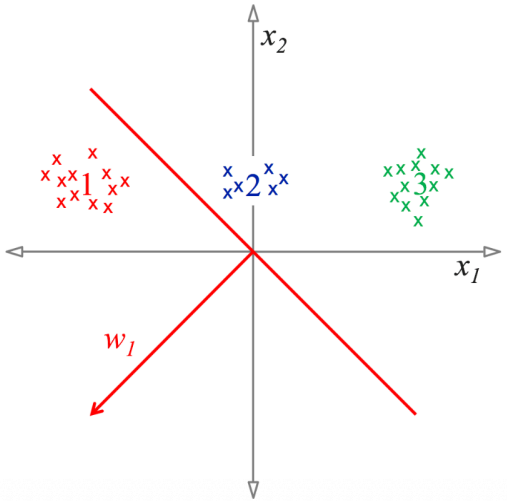
Good: Only reduce to k binary classifications, and done!

Bad: Scores can be inconsistent across classifiers

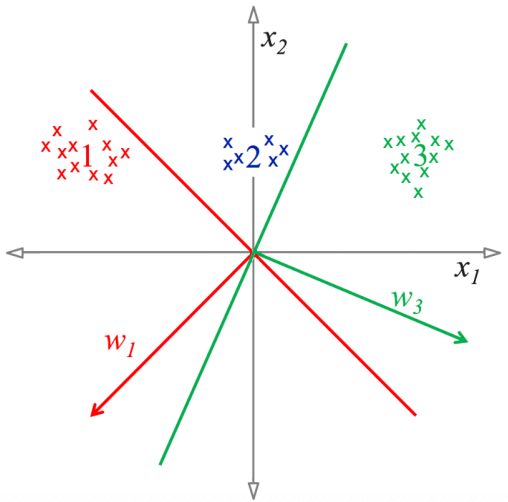
Reminder: Linear Binary Classifier

- Input space: X, output space: $y = \{-1, +1\}$.
- Linear classifier score function: $f(x) = < w, x > = w^\top \cdot x$
- Final classification prediction: sign($f(x)$) = *-1* or *+1*.

One-vs-All Example: 3 Classes



class 1 vs. others



class 3 vs. others

What about
class 2 vs. others?

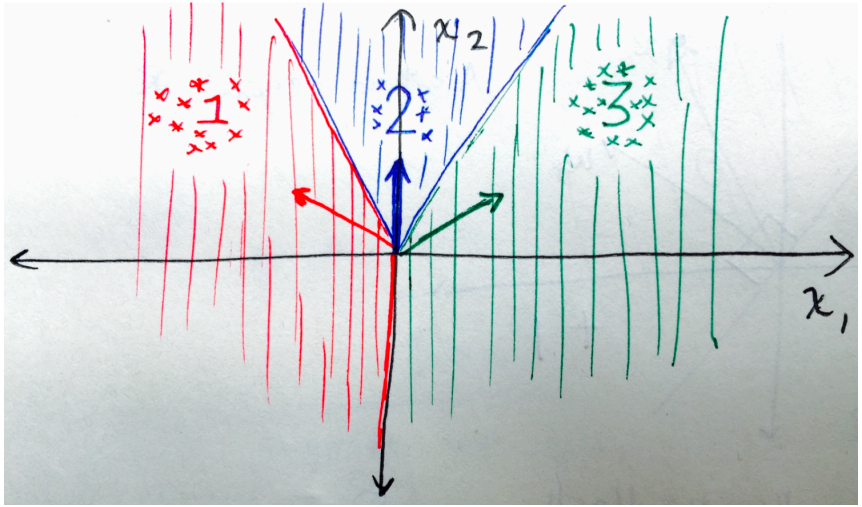
Bad: Fails for class 2.
Cannot find a line to
distinguish class 2 from
other classes.

NYU

A Solution with Linear Functions?

Here, we cannot use One-vs-All approach.

We need another learning algorithm that can find class 2.



Plot courtesy of David Rosenberg.

Reframing
Base Hypothesis Space

Reframing Base Hypothesis Space

- Input space: $\mathcal{X}$, output space $y \in \mathbb{R}$.
- $h_i(x)$ **scores** how likely x is from class i.
- Reframe **Base Hypothesis Space**:

$$\mathcal{H} = \{h : \mathcal{X} \to \mathbb{R}\} \quad \Rightarrow \quad \mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$$

What is h?
Compute compatibility score between input x and output y.

NYU

Reframing Base Hypothesis Space

- Base Hypothesis Space: $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$
- Training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
- What type of h do we want?
- Want $\boldsymbol{h(x, y)}$ to be **large** when $\boldsymbol{x}$ **has label** $\boldsymbol{y}$, and small otherwise.

Learning in Multiclass Hypothesis Space

- $h(x, y)$ classifies (x_i, y_i) correctly iff:

$$h(x_i, y_i) > h(x_i, y) \quad \forall y \neq y_i$$

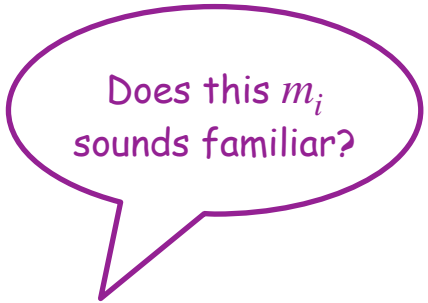
- $h(x, y)$ should give the highest score for the correct y.

- An equivalent condition is to use $\max$:

$$h(x_i, y_i) > \max_{y \neq y_i} h(x_i, y)$$

- We define m_i. Classification is correct if $m_i > 0$. Generally want m_i to be large.

$$m_i = h(x_i, y_i) - \max_{y \neq y_i} h(x_i, y)$$



NYU

Linear Multiclass Hypothesis Space

Multiclass Hypothesis Space

Learning function is $h(x, y)$

- Base Hypothesis Space: $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$
  - $h(x, y)$ gives **compatibility score** between input x and output y.
- Multiclass Hypothesis Space: $\mathcal{F} = \left\{ f : x \to \arg\max_{y \in \mathcal{Y}} h(x, y) \,\middle|\, h \in \mathcal{H} \right\}$
  - Prediction function $f(x) = \arg\max_{y} h(x, y).$

Prediction function is $f(x)$

- For each $f \in \mathcal{F}$, there is a compatibility score function $h \in \mathcal{H}$.

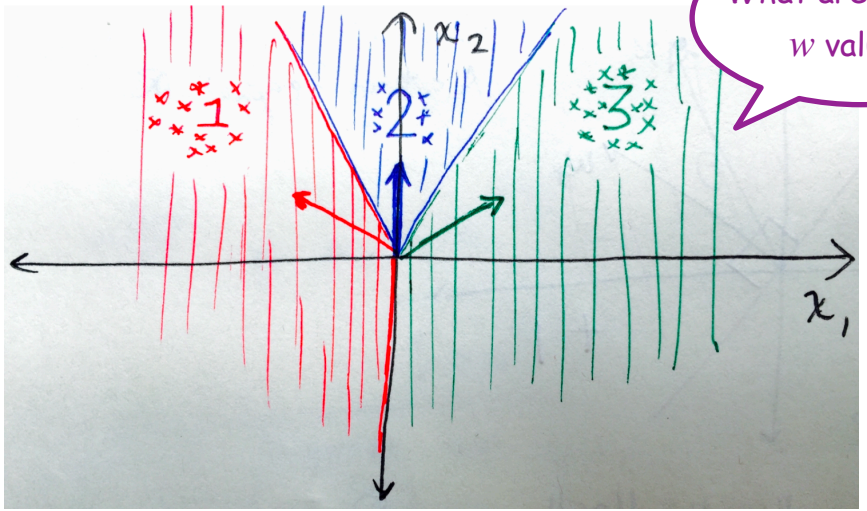
Linear Score Function

- A **linear** score function is: $h(x, y) = \langle w, \Psi(x, y) \rangle$ where

  $\Psi(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is a **feature map** and $w \in \mathbb{R}^d$ is a **weight** vector.

- $\Psi(x, y)$ extracts features relevant to how compatible y is with x.
- The linear score function is **linear** in w.

Example: $\mathcal{X} = \mathbb{R}^2, \quad \mathcal{Y} = \{1,2,3\}$



What are possible w values?

w_1 points towards top left

$$w_1 = \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

$$w_2 = (0,1)$$

w_2 points up

$$w_3 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

w_3 points towards top right

How to apply this to the linear score function framework?

Example: $\mathcal{X} = \mathbb{R}^2, \quad \mathcal{Y} = \{1,2,3\}$

- We need a **feature map** $\Psi(x, y)$ and a **weight vector** w.
- We have **prediction function** $(x_1, x_2) \rightarrow \arg \max_{i \in \{1,2,3\}} \langle w_i, (x_1, x_2) \rangle$.
- How can we get this into the form: $x \rightarrow \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$?

Example: $\mathcal{X} = \mathbb{R}^2, \quad \mathcal{Y} = \{1,2,3\}$

- We can stack w_i together to get: $w = \Big(\underbrace{-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_1}, \underbrace{0,1}_{w_2}, \underbrace{\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_3} \Big)$

- We can define: $\Psi : \quad \mathbb{R}^2 \times \{1,2,3\} \to \mathbb{R}^6$

$$\Psi(x,\ 1) := (x_1,\ x_2,\ 0,\ 0,\ 0,\ 0)$$
$$\Psi(x,\ 2) := (0,\ 0,\ x_1,\ x_2,\ 0,\ 0)$$
$$\Psi(x,\ 3) := (0,\ 0,\ 0,\ 0,\ x_1,\ x_2)$$

For each class

- Then, we can compute the dot product $\langle w, \Psi(x,y) \rangle = \langle w_y, x \rangle$

Summary & Takeaway

- **Multiclass classification** extends binary classification to k labels.
- **One-vs-All** reduces to **k binary classifiers**, but can **struggle if data is not linearly separable**.
- A solution is to **use a feature map** $\Psi(x, y)$ and a **weight vector** w.
- **Linear Multiclass models** compute **compatibility scores** as $\langle w, \Psi(x, y) \rangle$ and **predict the highest-scoring label**.

Optional: details in the following slides.

Thank you! 😄

Extra Slides

Example NLP: Part-of-speech classification

- $\mathcal{X}$ = All possible words.
- $\mathcal{Y}$ = [NOUN, VERB, ADJECTIVE, ADVERB, ARTICLE, PREPOSITION]. (6 labels)
- Features of $x \in \mathcal{X}$ can be: [the word itself], ENDS_IN_ly, ENS_IN_ness…
- How can we choose: $\Psi(x, y) = \big(\Psi_1(x, y), \Psi_2(x, y), \ldots, \Psi_d(x, y)\big)$?

$$
\begin{aligned}
\Psi_1(x, y) &= \mathbf{1}[x = \text{apple AND } y = \text{NOUN}] \\
\Psi_2(x, y) &= \mathbf{1}[x = \text{run AND } y = \text{NOUN}] \\
\Psi_3(x, y) &= \mathbf{1}[x = \text{run AND } y = \text{VERB}] \\
\Psi_4(x, y) &= \mathbf{1}[x \text{ ENDS\_IN\_ly AND } y = \text{ADVERB}] \\
&\vdots \quad \vdots \quad \vdots
\end{aligned}
$$

Each feature is a function of x and y.

- Then, $\Psi(x = \text{run}, y = \text{NOUN}) = (0, 1, 0, 0, \ldots)$

Feature vector

Example NLP: How does it work?

- $\Psi(x, y) = \big(\Psi_1(x, y), \Psi_2(x, y), \ldots, \Psi_d(x, y)\big) \in \mathbb{R}^d$

$$\begin{aligned} \Psi_1(x, y) &= \mathbf{1}[x = \text{apple AND } y = \text{NOUN}] \\ \Psi_2(x, y) &= \mathbf{1}[x = \text{run AND } y = \text{NOUN}] \\ &\vdots \quad \vdots \quad \vdots \end{aligned}$$

- After training, we have learned $w \in \mathbb{R}^d$. For example: $w = (5,\ 3,\ 1,\ 4, \ldots)$.
- To predict label for $x =$ apple, we can compute the scores for each $y \in \mathcal{Y}$:

$$\begin{aligned} &\langle w,\ \Psi(x = \text{apple}, y = \text{VERB})\rangle \\ &\langle w,\ \Psi(x = \text{apple}, y = \text{NOUN})\rangle \\ &\langle w,\ \Psi(x = \text{apple}, y = \text{ADVERB})\rangle \\ &\vdots \end{aligned}$$

- Then, choose class y^* that gives the highest score.

Another Approach: Use Label Features

- What if we have a very large number of classes?
- **Make features for the classes** y.
- Common in advertising: $\mathcal{X}$: User and user context. $\mathcal{Y}$: A large set of banner ads.
- Suppose user x is shown many banner ads.
- We want to predict which one the user will click on.
- Possible features:

$$\Psi_1(x, y) = \mathbf{1}[x \text{ interested in sports AND } y = \text{ad relevant to sports}]$$
$$\Psi_2(x, y) = \mathbf{1}[x \text{ is in target demographic group of } y]$$
$$\Psi_3(x, y) = \mathbf{1}[x \text{ clicked on ad from company sponsoring } y]$$

Extra Slides:
Linear Multiclass SVM

The Margin for Multiclass

- Let $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be our **compatibility score function**.
- Define the **"margin"** between a correct class and any other class.

Definition

The **margin** of the score function on ith example (x_i, y_i) for class y is:

$$m_{i,y}(h) = h(x_i, y_i) - h(x_i, y)$$

- Want $m_{i,y}(h)$ to be large and positive for all $y \neq y_i$.
- For our linear hypothesis space, **margin** is:

$$m_{i,y}(w) = \langle w,\ \Psi(x_i, y_i) \rangle - \langle w,\ \Psi(x_i, y) \rangle$$

Multiclass SVM with Hinge Loss

- Recall binary SVM (without bias term):

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2 + \frac{c}{n}\sum_{i=1}^{n}\left(1 - y_i w^\top x_i\right)_+$$

Recall $(x)_+ = \max(0,\ x)$. $y_i w^\top x_i$ is the margin.

Worst-case margin = the largest loss for incorrect answers

- Multiclass SVM (Version 1):

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2 + \frac{c}{n}\sum_{i=1}^{n}\max_{y \neq y_i}\left(1 - m_{i,y}(w)\right)_+$$

where $m_{i,y}(w) = \left\langle w,\ \Psi(x_i, y_i)\right\rangle - \left\langle w,\ \Psi(x_i, y)\right\rangle$.

Class-Sensitive Loss

- When we have multiclass, not all errors are the same. Some are worse than others.
- Rather than 0/1 Loss, we may be interested by a more general loss:

$$\Delta : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}^{\geq 0}$$

Where $\mathcal{Y}$ = {all possible true classes}, $\mathcal{A}$ = {all possible predicted classes}

- We can use this Δ as our **target margin** for multiclass SVM.
- Multiclass SVM (Version 2):

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2 + \frac{c}{n}\sum_{i=1}^{n} \max_{y \neq y_i} \left(\Delta(y_i, y) - m_{i,y}(w)\right)_+$$

- We can think of $\Delta(y_i, y)$ as the **target margin** for example i and class y. Because if each margin $m_{i,y}(w) \geq$ target margin, then we don't incur a loss on example i.