Multi-Class Classification

Mario Parente

March 9, 2023

A Real Classification Problem

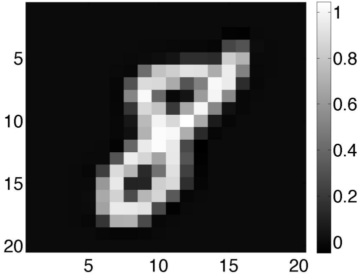
Classify handwritten digits.

$y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

We don't know how to solve this yet

Hand-written digit classification

Input: 20×20 grayscale image



Unroll the image into a vector

$$\begin{bmatrix} x_1 & x_{21} & \dots & x_{381} \\ x_2 & x_{22} & \dots & x_{382} \\ & & \vdots & \\ x_{20} & x_{40} & \dots & x_{400} \end{bmatrix}$$

Feature vector $\mathbf{x} \in \mathbb{R}^{400}$

$$\mathbf{x} = (x_1, \dots, x_{400})^T$$

Multi-class Classification

Input: $\mathbf{x} \in \mathbb{R}^m$ (continuous or discrete)
Labels: $y \in \{1, \ldots, K\}$

Multi-class Classification

Input: $\mathbf{x} \in \mathbb{R}^m$ (continuous or discrete)
Labels: $y \in \{1, \ldots, K\}$

Exercise: solve using logistic regression

- Use one or more binary $(y \in \{0, 1\})$ classifiers
- Hint: think about prediction first, then training.

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{x}^T$	y	y_1	y_2	y_3
$\cdots$	1	1	0	0
$\cdots$	2	0	1	0
$\cdots$	3	0	0	1

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{x}^T$	y	y_1	y_2	y_3
$\cdots$	1	1	0	0
$\cdots$	2	0	1	0
$\cdots$	3	0	0	1

Training?

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{x}^T$	y	y_1	y_2	y_3
$\cdots$	1	1	0	0
$\cdots$	2	0	1	0
$\cdots$	3	0	0	1

Training? for each class c, fit a binary classifier using training labels $y_c^{(i)}$ to get parameter vector $\boldsymbol{\theta}_c$

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{x}^T$	y	y_1	y_2	y_3
$\cdots$	1	1	0	0
$\cdots$	2	0	1	0
$\cdots$	3	0	0	1

Training? for each class c, fit a binary classifier using training labels $y_c^{(i)}$ to get parameter vector $\boldsymbol{\theta}_c$

Prediction?

One vs. All Classification

Learn a separate classifier for each class $c = 1, \ldots, K$

Labels for learning class c?

Let $y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{x}^T$	y	y_1	y_2	y_3
$\cdots$	1	1	0	0
$\cdots$	2	0	1	0
$\cdots$	3	0	0	1

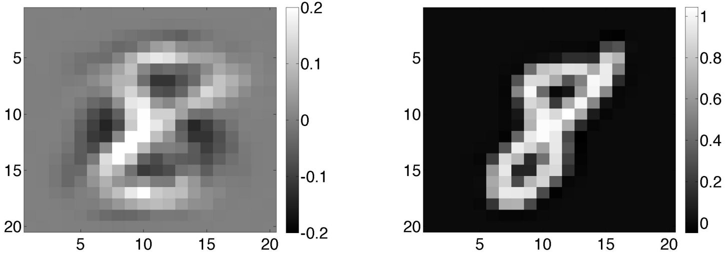
Training? for each class c, fit a binary classifier using training labels $y_c^{(i)}$ to get parameter vector $\boldsymbol{\theta}_c$

Prediction? make a prediction for each class and choose the one with *highest probability*

$$\text{predict } y = \mathrm{argmax}_c \, h_{\boldsymbol{\theta}_c}(\mathbf{x})$$

Visualization

Format weight vector as an image:



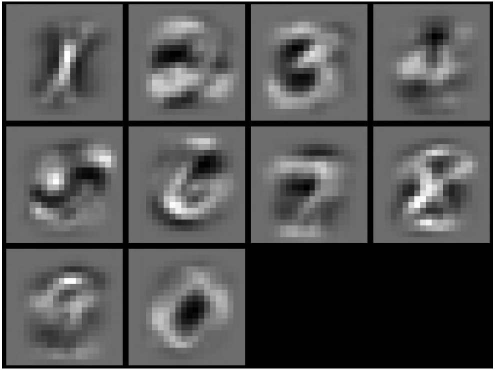
$\boldsymbol{\theta}_8$ $\qquad$ $\mathbf{x}$

Recall that

$$\text{Prediction} = \begin{cases} 1 & \boldsymbol{\theta}^T\mathbf{x} \geq 0 \\ 0 & \boldsymbol{\theta}^T\mathbf{x} < 0 \end{cases}$$

Dot product $=$ multiply together corresponding pixels and add

Visualization: One vs. All



One vs. One

Fit a classifier for each pair of classes

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

Let $y_{cd}^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{if } y^{(i)} = d \end{cases}$

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

$$\text{Let } y_{cd}^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{if } y^{(i)} = d \end{cases}$$

$\mathbf{x}^T$	y	y_{12}	y_{13}	y_{23}
$\cdots$	1	1	1	-
$\cdots$	2	0	-	1
$\cdots$	3	-	0	0

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

Let $y_{cd}^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{if } y^{(i)} = d \end{cases}$

$\mathbf{x}^T$	y	y_{12}	y_{13}	y_{23}
$\cdots$	1	1	1	-
$\cdots$	2	0	-	1
$\cdots$	3	-	0	0

Training? for each pair $c \neq d$, fit a binary classifier with labels $y_{cd}^{(i)}$ using **only examples from class** c **or** d

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

Let $y_{cd}^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{if } y^{(i)} = d \end{cases}$

$\mathbf{x}^T$	y	y_{12}	y_{13}	y_{23}
$\cdots$	1	1	1	-
$\cdots$	2	0	-	1
$\cdots$	3	-	0	0

Training? for each pair $c \neq d$, fit a binary classifier with labels $y_{cd}^{(i)}$ using **only examples from class** c **or** d

- Result: parameter vector $\boldsymbol{\theta}_{cd}$

Prediction?

One vs. One

Fit a classifier for each pair of classes

Labels for discriminating c from d?

Let $y_{cd}^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{if } y^{(i)} = d \end{cases}$

$\mathbf{x}^T$	y	y_{12}	y_{13}	y_{23}
$\cdots$	1	1	1	-
$\cdots$	2	0	-	1
$\cdots$	3	-	0	0

Training? for each pair $c \neq d$, fit a binary classifier with labels $y_{cd}^{(i)}$ using **only examples from class** c **or** d

- Result: parameter vector $\boldsymbol{\theta}_{cd}$

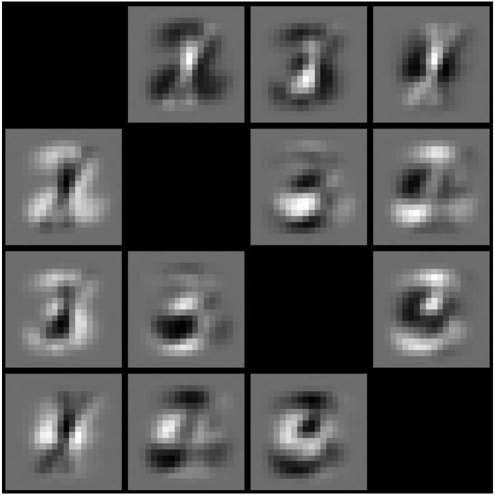
Prediction? voting scheme.

One vs. one Classification

Prediction? voting scheme.

$\mathbf{x}^T$	y_{12}	y_{13}	y_{23}	y
${\mathbf{x}^{(1)}}^T$	1	1	0	1
${\mathbf{x}^{(2)}}^T$	1	0	1	-
${\mathbf{x}^{(3)}}^T$	0	1	0	3

Visualization: One vs. One



Multiclass model

Previously we defined our model as

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{logistic}(\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n) = \text{logistic}(\boldsymbol{\theta}^T \mathbf{x})$$

where $\mathbf{x} = \left[1, x_1, \ldots, x_n\right]$ and $\theta = \left[\theta_0, \theta_1, \ldots, \theta_n\right]$.

We will now define our model as

$$h_{\mathbf{w}}(\mathbf{x}) = \text{logistic}(b + w_1 x_1 + \ldots + w_n x_n) = \text{logistic}(\mathbf{w}^T \mathbf{x} + b)$$

where $\mathbf{x} = \left[x_1, \ldots, x_n\right] \in \mathbb{R}^n$ is the **original feature vector** with no 1 added $\mathbf{w} \in \mathbb{R}^n$ is a **weight vector** (equivalent to $\theta_1, \ldots, \theta_n$ in the old notation) b is a scalar **intercept parameter** (equivalent to θ_0 in our old notation)

One vs. all Classification

For each class $c = 1, \ldots, K$

fit a logistic regression model to distinguish class c from the others using the labels

$$y_c^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} = c \\ 0 & \text{otherwise.} \end{cases}$$

This training procedure will result in a weight vector $\mathbf{w}_c$ and an intercept parameter b_c that can be used to predict the probability that a new example $\mathbf{x}$ belongs to class c:

$$\text{logistic}(\mathbf{w}_c^T \mathbf{x} + b_c) = \text{probability that } \mathbf{x} \text{ belongs to class } c.$$

The overall training procedure will yield one weight vector for each class. To make the final prediction for a new example, select the class with highest predicted probability:

$$\text{predicted class} = \text{the value of } c \text{ that maximizes logistic}(\mathbf{w}_c^T \mathbf{x} + b_c).$$