Multiclass linear prediction

Multiclass classification

Of the classification methods we have studied so far, which seem inherently binary?

- Nearest neighbor?
- Generative models?
- Linear classifiers?

From binary to multiclass logistic regression

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^d$, classifier given by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x + b}}{1 + e^{w \cdot x + b}}$$

From binary to multiclass logistic regression

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^d$, classifier given by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x + b}}{1 + e^{w \cdot x + b}}$$

When the labels are $\mathcal{Y} = \{1, 2, \ldots, k\}$:
Specify a classifier by $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$:

$$\Pr(y = j|x) = \frac{e^{w_j \cdot x + b_j}}{e^{w_1 \cdot x + b_1} + \ldots + e^{w_k \cdot x + b_k}}$$

From binary to multiclass logistic regression

Binary logistic regression: for $\mathcal{X} = \mathbb{R}^d$, classifier given by $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

$$\Pr(y = 1|x) = \frac{e^{w \cdot x + b}}{1 + e^{w \cdot x + b}}$$

When the labels are $\mathcal{Y} = \{1, 2, \ldots, k\}$:
Specify a classifier by $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$:

$$\Pr(y = j|x) = \frac{e^{w_j \cdot x + b_j}}{e^{w_1 \cdot x + b_1} + \ldots + e^{w_k \cdot x + b_k}}$$

Prediction: given a point x, predict label

$$\arg\max_j \Pr(y = j|x) = \arg\max_j (w_j \cdot x + b_j)$$

Multiclass logistic regression

- **Label space**: $\mathcal{Y} = \{1, 2, \ldots, k\}$
- **Parametrized classifier**: $w_1, \ldots, w_k \in \mathbb{R}^d$, $b_1, \ldots, b_k \in \mathbb{R}$:

$$\Pr(y = j|x) = \frac{e^{w_j \cdot x + b_j}}{e^{w_1 \cdot x + b_1} + \ldots + e^{w_k \cdot x + b_k}}$$

- **Prediction**: given a point x, predict label $\arg\max_j (w_j \cdot x + b_j)$.
- **Learning**: Given: $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$.
  Find: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k$ that maximize the likelihood

$$\prod_{i=1}^{n} \Pr(y^{(i)}|x^{(i)})$$

  Taking negative log gives a convex minimization problem.

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, 2, \ldots, k\}$

Model: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$

Prediction: On instance x, predict label $\arg\max_j (w_j \cdot x + b_j)$

Multiclass Perceptron

Setting: $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{1, 2, \ldots, k\}$

Model: $w_1, \ldots, w_k \in \mathbb{R}^d$ and $b_1, \ldots, b_k \in \mathbb{R}$

Prediction: On instance x, predict label $\arg\max_j (w_j \cdot x + b_j)$

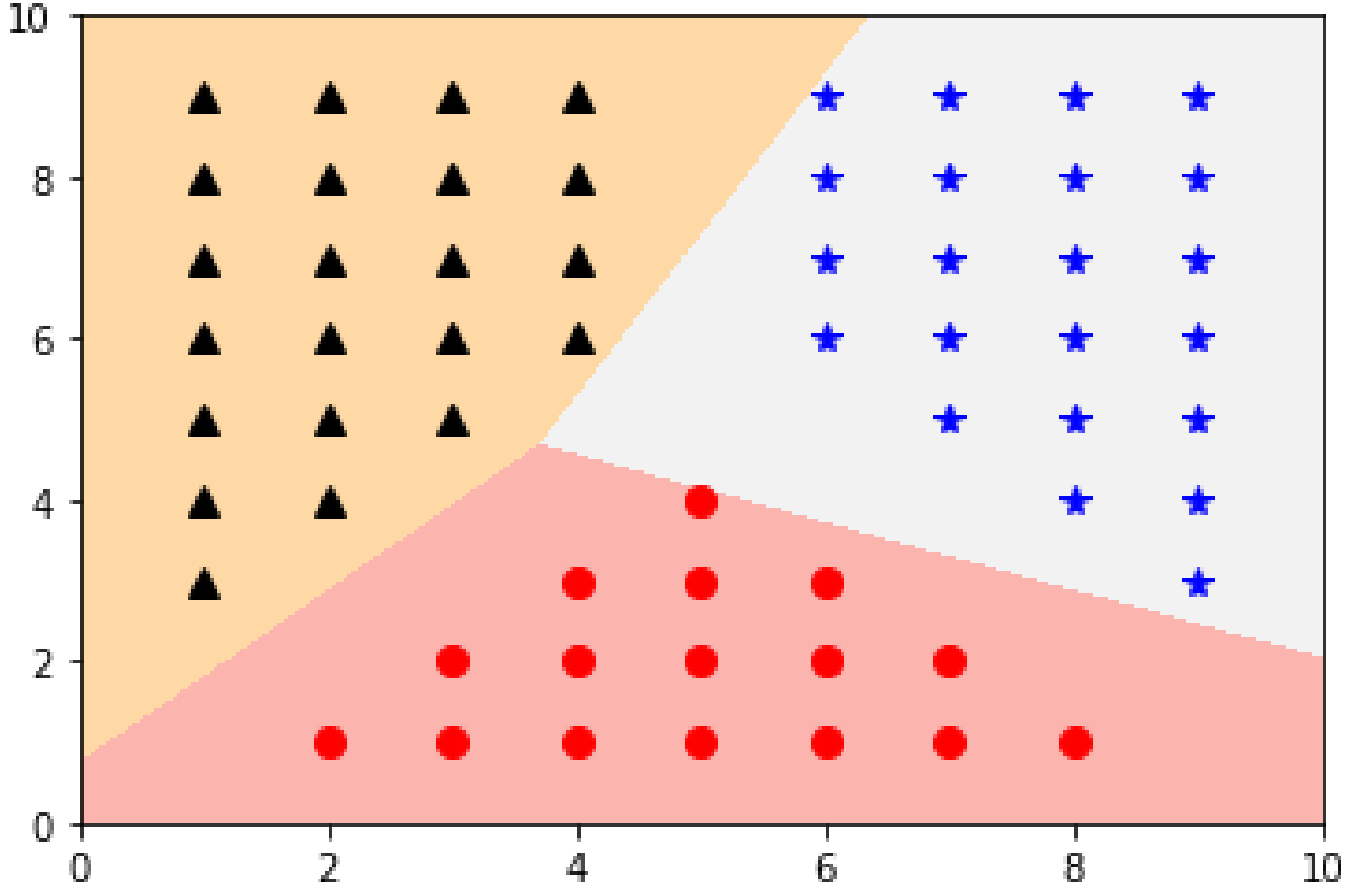
Learning. Given training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$:

- Initialize $w_1 = \cdots = w_k = 0$ and $b_1 = \cdots = b_k = 0$
- Repeat while some training point (x, y) is misclassified:

$$\begin{aligned} \text{for correct label } y: \quad & w_y = w_y + x \\ & b_y = b_y + 1 \\ \text{for predicted label } \widehat{y}: \quad & w_{\widehat{y}} = w_{\widehat{y}} - x \\ & b_{\widehat{y}} = b_{\widehat{y}} - 1 \end{aligned}$$

Multiclass Perceptron: example

Multiclass Perceptron: example



For logistic regression, if $y_i \in \{-1, 1\}$ then the loss function, L is as below:

$$L = \sum_{i=1}^{n} \log\left(1 + e^{-y_i(w^T x_i)}\right)$$ (here bias is captured in w)

For the same problem, if $y_i \in \{0, 1\}$ i.e. the negative class labelled as 0 instead of -1, the loss function becomes:

~~$L = -\sum_{i=1}^{n} y_i (\log y_i$~~

$$L = -\left[\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

where $\hat{y}_i$ is the prediction

For the multiclass problem, we have $y_i \in \{1, 2, \ldots, K\}$, one of K classes and K-sets of weights and biases i.e $(w_1, b_1), (w_2, b_2) \ldots (w_K, b_K)$
The loss function in this case is

~~$L = -\sum_{i=1}^{n} \sum_{k=1}^{K} 1\{y_i$~~

$$L = -\sum_{i=1}^{n} \sum_{k=1}^{K} 1\{y_i = k\} \log \hat{y}_{i,k}$$

where $1\{\epsilon\}$ is an Indicator function, which equates to 1 if condition C is True or else 0

Ex: $1\{\text{sun rises in the East}\} = 1$

$1\{x\%2 == 0 \;\forall\, x \in \text{Prime numbers}\} = 0$

$$\hat{y}_{i,k} = e^{(x_i \cdot \vec{w}_k^T + b_k)} \Big/ \sum_{j=1}^{K} e^{(x_i \vec{w}_j^T + b_j)}$$ (softmax)

Example of multiclass logistic regression

We have 4 classes G (Gryffindor), R (Ravenclaw), H (Hufflepuff) & S (Slytherine) We are given a bunch of students with four traits rated on a scale of 0 to 10 for each of them. The traits are smart, resourceful, creative & risktaking.

	Smart	Resourceful	Creative	Risktaking	Labels
x_1	8	5	2	6	→ R (Ravenclaw)
x_2	5	4	3	7	→ G (Gryffindor)
x_3	5	7	4	4	→ S (Slytherine)
	⋮	⋮	⋮	⋮	

Lets pick x_1.

Say ~~let~~ $x_1\vec{w}_g + b_g$, $x_1\vec{w}_r + b_r$, $x_1\vec{w}_h + b_h$, $x_1\vec{w}_s + b_s$ be (11.5, 10, 7.2, 11)

Softmax of these logits gives (0.54, 0.12, 0.007, 0.33).

The loss for this sample is l_1 =

$$-\left(0(\log 0.54) + 1(\log 0.12) + 0(\log(0.007)) + 0(\log 0.33)\right)$$

$l_1 = -\log 0.12$

We compute similarly $l_2, \ldots l_n$

$L = \sum l_i$ or $L = \frac{\sum l_i}{n}$

Note that the denominator n is not required as it is a constant. We can ignore it and still get the correct result.

Applying the learning procedure for multiclass perceptron for the same data:

Initially $w_R = (0,0,0,0)$, $b_R = 0$. Similarly $w_g = w_r = w_s = (0,0,0,0)$ and $b_g = b_r = b_s = 0$

Lets pick x_1:

$w_g x_1 + b_g$, $w_r x_1 + b_r$, $w_R x_1 + b_R$, $w_s x_1 + b_s = (0,0,0,0)$

As all the values are same, we arbitrarily pick the ~~first~~ class G as our prediction.

$$w_r = \begin{pmatrix} +8 \\ +5 \\ +2 \\ +6 \end{pmatrix} \quad b_r = +1 \quad \text{and} \quad w_g = \begin{pmatrix} -8 \\ -5 \\ -2 \\ -6 \end{pmatrix}, \quad b_g = -1$$

Next we continue for $x_2 \ldots x_n$ updating the weights whenever we make a wrong prediction. Finally stopping at convergence.