Classificação multiclasses

Multiclass classification

C classes problem

Approach 1: Combine multiple binary classifiers

OVA scheme (*One versus All*):

- one classifier for each class: h_j is a binary classifier designed to recognize objects of class j amongst all obejcts
- total of C binary classifiers: h_j, $j = 1, 2, \ldots, C$
- assume each classifier returns a score in $[0, 1]$
- **Decision**: given $\mathbf{x}$, let $\hat{y} = \arg\max_j \{h_j(\mathbf{x})\}$

Multiclass classification

OVO scheme (*One versus One*):

- one classifier for each pair of classes: h_{jk} is a binary classifier trained using only examples from class j (positive) and k (negative)
- total of $\frac{C(C-1)}{2}$ binary classifiers: h_{jk}, $j < k$, $j, k = 1, 2, \ldots, C$ (note that for $k > j$, we have $h_{kj} = 1 - h_{jk}$)
- assume each classifier returns a score in $[0, 1]$
- **Decision**: given $\mathbf{x}$, let $\hat{y} = \underset{j \in \{1,2,\ldots,C\}}{\arg\max} \left\{ \sum_{k=1}^{C} h_{jk}(\mathbf{x}) \right\}$

Example of a binary classifier that outputs a score in $[0, 1]$

Logistic regression (sigmoid)

$$\hat{p}_1 = \hat{P}(y = 1|\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

Its output ($\hat{p}_1 = \hat{P}(y = 1|\mathbf{x})$) is interpreted as a probability

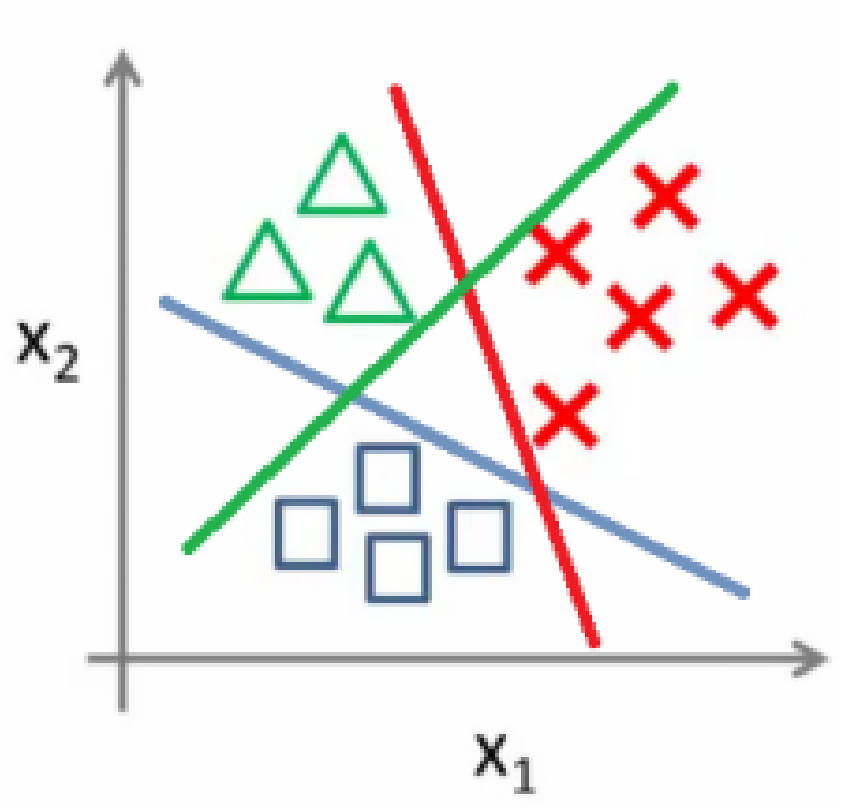
Note that OVA and OVO can be based on any type of binary classifiers. If the classifiers return a score value (that is, $P(y|\mathbf{x})$), then the rules given earlier can be used.

What if we use **hard classifiers** instead of **soft classifiers** ?

Hard classifier: output in $\{0, 1\}$ (class label y)
Soft classifier: output in $[0, 1]$ (conditional probability $P(y|\mathbf{x})$)

We can use, for instance, the **majority vote**

Voting may lead to **regions with undefined classification**



Fonte: https://utkuufuk.com/2018/06/03/one-vs-all-classification/

The triangular region at the center will receive no classification

There are many ways to **combine multiple binary classifiers** to implement multiclass classification.

(see for instance: *A review on the combination of binary classifiers in multiclass problems*, Ana C. Lorena, André C. P. L. F. de Carvalho, João M. P. Gama)

Similarly, **classifier combination** / **ensemble of classifiers** are topics vastly studied in the field of machine learning

(see for instance: *Combining Pattern Classifiers: Methods and Algorithms*, Ludmila I. Kuncheva)

It is not our goal here to discuss them exhaustively

Approach 2: Inherently multiclass

Any method that estimates the C conditionals $P(y = j \mid \mathbf{x})$, $j = 1, 2, \ldots, C$ at once

Multinomial logistic regression

The generalization of logistic regression for multiple classes is known as **multinomial logistic regression**

To estimate the conditional probabilities we use the **softmax function**:

$$\hat{p}_j = \hat{P}(y = j|\mathbf{x}) = \frac{e^{\mathbf{w}_j^T \mathbf{x}}}{\sum_{i=1}^{C} e^{\mathbf{w}_i^T \mathbf{x}}}, \quad j = 1, 2, \ldots, C$$

Example for $C = 3$ classes:

$$\hat{p}_1 = \hat{P}(y = 1|\mathbf{x}) = \frac{e^{\mathbf{w}_1^T \mathbf{x}}}{e^{\mathbf{w}_1^T \mathbf{x}} + e^{\mathbf{w}_2^T \mathbf{x}} + e^{\mathbf{w}_3^T \mathbf{x}}}$$

$$\hat{p}_2 = \hat{P}(y = 2|\mathbf{x}) = \frac{e^{\mathbf{w}_2^T \mathbf{x}}}{e^{\mathbf{w}_1^T \mathbf{x}} + e^{\mathbf{w}_2^T \mathbf{x}} + e^{\mathbf{w}_3^T \mathbf{x}}}$$

$$\hat{p}_3 = \hat{P}(y = 3|\mathbf{x}) = \frac{e^{\mathbf{w}_3^T \mathbf{x}}}{e^{\mathbf{w}_1^T \mathbf{x}} + e^{\mathbf{w}_2^T \mathbf{x}} + e^{\mathbf{w}_3^T \mathbf{x}}}$$

Clearly $\hat{p}_1 + \hat{p}_2 + \hat{p}_3 = 1$

Also $0 \leq \hat{p}_j \leq 1$

Observe that in the binary classification case, we used

$$\hat{p}_1 = \hat{P}(y = 1|\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

It can be rewritten as:

$$\frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}} = \frac{e^{\mathbf{w}^T\mathbf{x}}}{e^{\mathbf{w}^T\mathbf{x}}} \frac{1}{(1 + e^{-\mathbf{w}^T\mathbf{x}})} = \frac{e^{\mathbf{w}^T\mathbf{x}}}{e^{\mathbf{w}^T\mathbf{x}} + 1}$$

Hence:

$$\begin{aligned} \hat{p}_0 &= \hat{P}(y = 0|\mathbf{x}) = 1 - \hat{P}(y = 1|\mathbf{x}) \\ &= 1 - \frac{e^{\mathbf{w}^T\mathbf{x}}}{e^{\mathbf{w}^T\mathbf{x}} + 1} = \frac{1}{1 + e^{\mathbf{w}^T\mathbf{x}}} \end{aligned}$$

and

$$\hat{p}_1 + \hat{p}_0 = \hat{P}(y = 1|\mathbf{x}) + \hat{P}(y = 0|\mathbf{x}) = 1$$

Recall (previous page):

$$\hat{P}(y=0|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}} \qquad \hat{P}(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}$$

The **softmax formulation for two classes**:

$$\hat{P}(y=0|\mathbf{x}) = \frac{1}{(1+e^{\mathbf{w}^T\mathbf{x}})}\frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}} = \frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}+e^{(\mathbf{w}+\mathbf{w}_0)^T\mathbf{x}}}$$

$$\hat{P}(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{(1+e^{\mathbf{w}^T\mathbf{x}})}\frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}} = \frac{e^{(\mathbf{w}+\mathbf{w}_0)^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}+e^{(\mathbf{w}+\mathbf{w}_0)^T\mathbf{x}}}$$

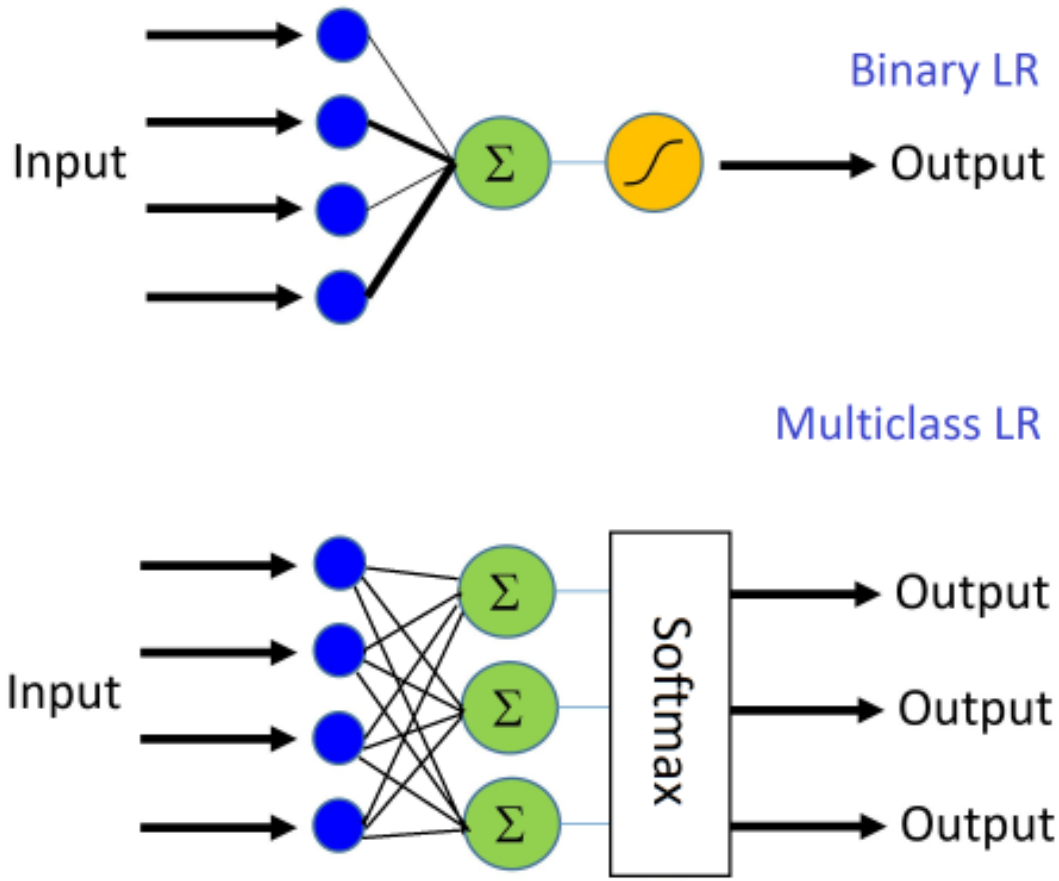
Recall (previous page):

$$\hat{P}(y=0|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}} \qquad \hat{P}(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}$$

The **softmax formulation for two classes**:

$$\hat{P}(y=0|\mathbf{x}) = \frac{1}{(1+e^{\mathbf{w}^T\mathbf{x}})}\frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}} = \frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}} + e^{(\underbrace{\mathbf{w}+\mathbf{w}_0}_{\mathbf{w}_1})^T\mathbf{x}}}$$

$$\hat{P}(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{(1+e^{\mathbf{w}^T\mathbf{x}})}\frac{e^{\mathbf{w}_0^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}}} = \frac{e^{\overbrace{(\mathbf{w}+\mathbf{w}_0)}^{\mathbf{w}_1}{}^T\mathbf{x}}}{e^{\mathbf{w}_0^T\mathbf{x}} + e^{(\underbrace{\mathbf{w}+\mathbf{w}_0}_{\mathbf{w}_1})^T\mathbf{x}}}$$



Fonte: https://www.cntk.ai/pythondocs/CNTK_103B_MNIST_LogisticRegression.html

Cost function for multi-output case

One-hot encoding of the output:

For each input $\mathbf{x}^{(i)}$, the output is a vector $\mathbf{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_C^{(i)})$ with $y_j^{(i)} = 1 \iff \mathbf{x}^{(i)}$ is from class j, $j = 1, 2, \ldots, C$

Cross-entropy loss (wrt inputs $\mathbf{x}^{(i)} \in D$):

$$\sum_{i=1}^{N} \sum_{j=1}^{C} y_j^{(i)} \log \hat{p}_j^{(i)}$$

Note that: $\hat{p}_j^{(i)} = \hat{P}(y^{(i)} = j \,|\, \mathbf{x}^{(i)})$, $\sum_{j=1}^{C} \hat{p}_j^{(i)} = 1$, and the parameters to be optimized, $\mathbf{w}_j$, are those in the softmax function $\dfrac{e^{\mathbf{w}_j^T \mathbf{x}}}{\sum_{i=1}^{C} e^{\mathbf{w}_i^T \mathbf{x}}}$