Logistic regression

Classification

- Email: spam / not spam?
- Credit card charge: fraudulent yet/no?
- Cat/Dog?

Supervised learning for classification

- Very similar to regression
- Supervised data $(\boldsymbol{x}, y)$
- The only new thing here is that y can be only 0 and 1

Can we use regression to do classification?

- Yes. We use a hypothesis function $f(x, \theta)$ and a threshold e.g. $t = 0.5$
- As before, we will use $f(x, \theta) = \theta^T \boldsymbol{x}$
- If $f(x, \theta) \geq 0.5$ predict $\hat{y} = 1$
- If $f(x, \theta) < 0.5$ predict $\hat{y} = 0$

Motivations for moving beyond regression + threshold

- The value of $u = f()$ can be other numbers eg. -1000.0, -1, 0.4999
- All these map to the same classifier outcome based on a threshold?
- Idea:
    - Let us squeeze these values in the [0,1] range!

Let us think about these numbers

- **Probability** of event occuring

$$p = P(y = 1)$$

- **Odds** of an event occuring:

$$\frac{p}{1-p}$$

- **Log-odds** aka **logit** of the event occuring

$$u = log\left(\frac{p}{1-p}\right)$$

- We will search for the logit!

The inverse of this:

$$p = \frac{1}{1 + e^{-u}}$$

or

$$p = \frac{e^u}{e^u + 1}$$

- This is called the **logistic function**. It is an S-shaped (sigmoid) function.

What about multi-class classification?

- Assume that we have values $u_1, u_2, \ldots u_n$

$$p_k = \frac{e^{u_k}}{\sum_i e^{u_i}}$$

- This function is called a **softmax**

What is the u in this case?

- If u is the output of a **linear regression**, this is called **logistic regression**

$$u = \theta^T \boldsymbol{x}$$

$$f(x;\theta) = \frac{1}{1 + e^{-\theta^T \boldsymbol{x}}}$$

- However, the u can be calculated by other things as well. For instance, a neural network!

How do we find the θ?

- Stochastic gradient descent!
- But we need a good loss function, and the Euclidean distance just doesn't feel right

Logistic regression cost function

$$\mathcal{L}(\theta) = \begin{cases} -log(f(\boldsymbol{x};\theta)) & \text{if } y = 1 \\ -log(1 - f(\boldsymbol{x};\theta)) & \text{if } y = 0 \end{cases}$$

- Intuition: if $y = 1$ and $f(\boldsymbol{x};\theta)$, loss is zero
- But if $f(\boldsymbol{x};\theta)$ is close to zero, loss is very high

Making it work for gradient descent

$$\mathcal{L}(\theta) = \begin{cases} -log(f(\boldsymbol{x};\theta)) & \text{if } y = 1 \\ -log(1 - f(\boldsymbol{x};\theta)) & \text{if } y = 0 \end{cases}$$

- This function, with the bracketed cases, is difficult to differentiate.
- Exploiting the fact that y can only be 0 or 1, we can do a trick

$$\mathcal{L}(\theta) = -y \cdot log(f(\boldsymbol{x};\theta)) - (1 - y) \cdot log(1 - f(\boldsymbol{x};\theta))$$

- This function is differentiable, so we can use stochastic gradient descent.
- We will use batches, so the function will be of the form

$$\mathcal{L} = \frac{1}{m}\sum_i \text{etc}$$