CS2109S Tutorial 5

Classification and Logistic Regression

(AY 24/25 Semester 2)

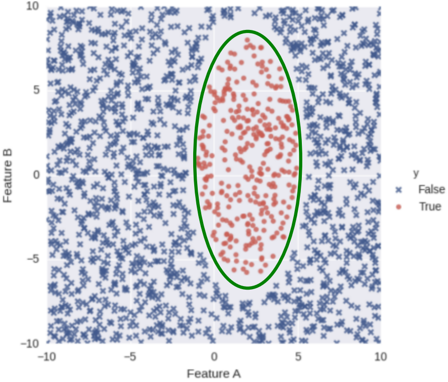
March 14, 2025

(Prepared by Benson)

Contents

Q1. Linear vs Non-linear Separability

Define a (minimal) set of features that will perfectly classify whether or not a bunny can be released into the wild.



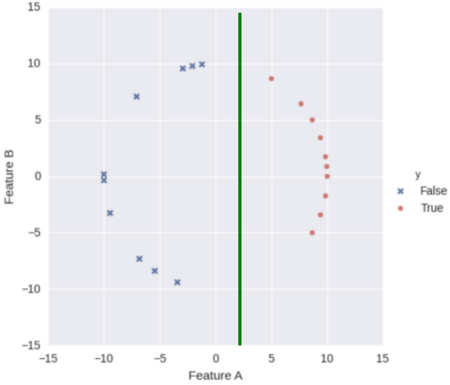
Equation of an ellipse:

$$\frac{(A-x)^2}{a^2} + \frac{(B-y)^2}{b^2} = 1$$

A minimal set of features is (A^2, B^2, A, B).

Q1. Linear vs Non-linear Separability

Define a (minimal) set of features that will perfectly classify whether or not a bunny can be released into the wild.



A minimal set of features is (A).

Recap: Logistic Regression

Target: Solve **classification** tasks (output a **probability** between 0 and 1).

- First Try (regression): $p = \boldsymbol{w} \cdot \boldsymbol{x}$
  - FAIL: $\boldsymbol{w} \cdot \boldsymbol{x}$ can be outside $[0, 1]$.
- Second Try (odds ratio): $\dfrac{p}{1-p} = \boldsymbol{w} \cdot \boldsymbol{x}$
  - FAIL: $\dfrac{p}{1-p}$ is always non-negative, but $\boldsymbol{w} \cdot \boldsymbol{x}$ can be outside $[0, \infty)$.
- Third Try (logits): $\log\left(\dfrac{p}{1-p}\right) = \boldsymbol{w} \cdot \boldsymbol{x}$ (works!)
  - Rearrange terms: $p = \dfrac{1}{1 + e^{-\boldsymbol{w} \cdot \boldsymbol{x}}}$

Recap: Logistic Regression

Hypothesis:

$$h_{\boldsymbol{w}}(x) = \frac{1}{1 + e^{-\boldsymbol{w}^\top \boldsymbol{x}}} = \sigma(\boldsymbol{w}^\top \boldsymbol{x})$$

$$\boxed{\sigma(z) = \frac{1}{1 + e^{-z}}}$$

Loss function:

- **Mean Squared Error** is not convex under logistic regression $\Rightarrow$ gradient descent might not reach global minimum.
- Use **binary cross entropy loss** instead:

"surprisal" in entropy!

$$BCE(\hat{y}) = \begin{cases} \boxed{-\log(\hat{y})} & \text{if } y = 1 \\ -\log(1 - \hat{y}) & \text{if } y = 0 \end{cases}$$

$$= -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

Q2. Loss Function of Logistic Regression

Extra Slide

$$h_{\boldsymbol{w}}(x) = \frac{1}{1 + e^{-\boldsymbol{w}^\top \boldsymbol{x}}}$$

(a) Write down the probability p as a function of $\boldsymbol{x}$ and calculate the derivative of $\log(p)$ with respect to each weight w_i.

$$p = \frac{1}{1 + e^{-\boldsymbol{w}^\top \boldsymbol{x}}} = \frac{1}{1 + e^{-\boldsymbol{w} \cdot \boldsymbol{x}}} = \frac{1}{1 + e^{\sum_{i=1}^{n} -w_i x_i}}$$

$\log \frac{1}{x} = -\log x$

$$\log(p) = \log\left(\frac{1}{1 + e^{\sum_{i=1}^{n} -w_i x_i}}\right) = -\log\left(1 + e^{\sum_{i=1}^{n} -w_i x_i}\right)$$

Chain Rule

$\frac{\partial}{\partial x} \log x$

$$\frac{\partial \log(p)}{\partial w_i} = -\left(\frac{1}{1 + e^{\sum_{i=1}^{n} -w_i x_i}} \cdot \frac{\partial}{\partial w_i}\left(1 + e^{\sum_{i=1}^{n} -w_i x_i}\right)\right)$$

Chain Rule

$$= -\left(\frac{1}{1 + e^{\sum_{i=1}^{n} -w_i x_i}} \cdot \underbrace{\left(e^{\sum_{i=1}^{n} -w_i x_i}\right)}_{\frac{\partial}{\partial x}(1+e^x)}\underbrace{(-x_i)}_{\frac{\partial}{\partial w_i}\left(\sum_{i=1}^{n} -w_i x_i\right)}\right)$$

$$= \frac{e^{\sum_{i=1}^{n} -w_i x_i}}{1 + e^{\sum_{i=1}^{n} -w_i x_i}} \cdot (x_i) = \boxed{(1-p)x_i}$$

Q2. Loss Function of Logistic Regression

(b) Write down the probability $1-p$ as a function of x and calculate the derivative of $\log(1-p)$ with respect to each weight w_i.

$$1-p = \overbrace{\frac{e^{-\boldsymbol{w}^\top \boldsymbol{x}}}{1+e^{-\boldsymbol{w}^\top \boldsymbol{x}}}}^{\frac{1}{1+e^{-\boldsymbol{w}^\top \boldsymbol{x}}}} \cdot \frac{e^{\boldsymbol{w}^\top \boldsymbol{x}}}{e^{\boldsymbol{w}^\top \boldsymbol{x}}} = \frac{1}{1+e^{\boldsymbol{w}^\top \boldsymbol{x}}} = \frac{1}{1+e^{\boldsymbol{w}\cdot\boldsymbol{x}}} = \frac{1}{1+e^{\sum_{i=1}^n w_i x_i}}$$

$$\log(1-p) = \log\left(\frac{1}{1+e^{\sum_{i=1}^n w_i x_i}}\right) = -\log\left(1+e^{\sum_{i=1}^n w_i x_i}\right)$$

Chain Rule

$$\frac{\partial \log(1-p)}{\partial w_i} = -\left(\overbrace{\frac{1}{1+e^{\sum_{i=1}^n w_i x_i}}}^{\frac{\partial}{\partial x}\log x} \cdot \frac{\partial}{\partial w_i}\left(1+e^{\sum_{i=1}^n w_i x_i}\right)\right)$$

Chain Rule

$$= -\left(\frac{1}{1+e^{\sum_{i=1}^n w_i x_i}} \cdot \underbrace{\left(e^{\sum_{i=1}^n w_i x_i}\right)}_{\frac{\partial}{\partial x}(1+e^x)} \underbrace{(x_i)}_{\frac{\partial}{\partial w_i}\left(\sum_{i=1}^n w_i x_i\right)}\right)$$

$$= -\frac{e^{\sum_{i=1}^n w_i x_i}}{1+e^{\sum_{i=1}^n w_i x_i}} \cdot \frac{e^{\sum_{i=1}^n -w_i x_i}}{e^{\sum_{i=1}^n -w_i x_i}} \cdot (x_i) = -\frac{1}{1+e^{\sum_{i=1}^n -w_i x_i}} \cdot (x_i) = \boxed{-p x_i}$$

Q2. Loss Function of Logistic Regression

Derivative of Sigmoid Function

Let $\sigma(z) = \dfrac{1}{1+e^{-z}}$. We have $\sigma'(z) = \sigma(z)\,(1-\sigma(z))$.

Proof.

$$
\begin{aligned}
\sigma'(z) &= \frac{\frac{d}{dz}(1)\cdot(1+e^{-z}) - (1)\cdot\frac{d}{dz}(1+e^{-z})}{(1+e^{-z})^2} && \blacktriangleleft\ \text{quotient rule} \\
&= \frac{(0)\cdot(1+e^{-z}) - (1)\cdot(-e^{-z})}{(1+e^{-z})^2} \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1}{1+e^{-z}}\cdot\frac{e^{-z}}{1+e^{-z}} = \boxed{\sigma(z)(1-\sigma(z))}
\end{aligned}
$$

Q2. Loss Function of Logistic Regression

Derivative of Sigmoid Function

Let $\sigma(z) = \dfrac{1}{1+e^{-z}}$. We have $\sigma'(z) = \sigma(z)\,(1-\sigma(z))$.

Easier solution to (a), (b):

$$\frac{\partial \log(p)}{\partial w_i} \overset{\text{Chain Rule}}{=} \underbrace{\frac{1}{p}}_{\frac{\partial \log(p)}{\partial p}} \cdot \overbrace{\frac{\partial p}{\partial w_i}}^{p=\sigma(\boldsymbol{w}^\top \boldsymbol{x})} = \frac{1}{\cancel{p}} \cdot \overbrace{\cancel{p}(1-p)}^{\frac{\partial p}{\partial(\boldsymbol{w}^\top \boldsymbol{x})}} \cdot \underbrace{\frac{\partial}{\partial w_i}(\overbrace{\boldsymbol{w}^\top \boldsymbol{x}}^{\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}\cdot\boldsymbol{x} = \sum w_i x_i})}_{= x_i} = \boxed{(1-p)x_i}$$

Chain Rule

$$\frac{\partial \log(1-p)}{\partial w_i} = \underbrace{\frac{-1}{1-p}}_{\frac{\partial \log(1-p)}{\partial p}} \cdot \frac{\partial p}{\partial w_i} = \frac{-1}{\cancel{1-p}} \cdot p\cancel{(1-p)} \cdot \underbrace{\frac{\partial}{\partial w_i}(\boldsymbol{w}^\top \boldsymbol{x})}_{= x_i} = \boxed{-px_i}$$

Q2. Loss Function of Logistic Regression

(c) Using results from 3(a) and 3(b), derive $\frac{\partial L}{\partial w_i}$, where L is the loss function of logistic regression model.

$$L = -y\log(p) - (1-y)\log(1-p)$$

$$\begin{aligned}
\frac{\partial L}{\partial w_i} &= -\,y\frac{\partial \log(p)}{\partial w_i} - (1-y)\frac{\partial \log(1-p)}{\partial w_i} \\
&= -y(1-p)x_i - (1-y)(-px_i) \\
&= -yx_i + ypx_i + px_i - ypx_i \\
&= x_i(p-y) \\
&= x_i(h_{\boldsymbol{w}}(\boldsymbol{x}) - y)
\end{aligned}$$

Q4. Logistic Regression for Multi-Class Classification

(a) Compute the probability of an animal belonging to a certain class and classify them accordingly.

$$\boldsymbol{w}_{cat} = \begin{bmatrix} 4.2 & -0.01 & -0.12 \end{bmatrix}^\top$$
$$\boldsymbol{w}_{horse} = \begin{bmatrix} -20 & -0.08 & 35 \end{bmatrix}^\top$$
$$\boldsymbol{w}_{elephant} = \begin{bmatrix} -1250 & 0.82 & 0.9 \end{bmatrix}^\top$$

Weight (kg)	Length (m)
4.2	0.4
720	2.4
2350	5.5

First animal: $\boldsymbol{x} = \begin{bmatrix} 1 & 4.2 & 0.4 \end{bmatrix}^\top$

- $\boldsymbol{w}_{cat} \cdot \boldsymbol{x} = 1 \cdot 4.2 + 4.2 \cdot (-0.01) + 0.4 \cdot (-0.12) = 4.11$
  $$p_{cat} = \frac{1}{1 + e^{-4.11}} = 0.984$$
- $\boldsymbol{w}_{horse} \cdot \boldsymbol{x} = -6.336$
  $$p_{horse} = \frac{1}{1 + e^{6.336}} = 0.00177$$
- $\boldsymbol{w}_{elephant} \cdot \boldsymbol{x} = -1246.196$
  $$p_{elephant} = \frac{1}{1 + e^{1246.196}} \approx 0$$

Q4. Logistic Regression for Multi-Class Classification

(a) Compute the probability of an animal belonging to a certain class and classify them accordingly.

$$\boldsymbol{w}_{cat} = \begin{bmatrix} 4.2 & -0.01 & -0.12 \end{bmatrix}^\top$$
$$\boldsymbol{w}_{horse} = \begin{bmatrix} -20 & -0.08 & 35 \end{bmatrix}^\top$$
$$\boldsymbol{w}_{elephant} = \begin{bmatrix} -1250 & 0.82 & 0.9 \end{bmatrix}^\top$$

Weight (kg)	Length (m)
4.2	0.4
720	2.4
2350	5.5

Second animal: $\boldsymbol{x} = \begin{bmatrix} 1 & 720 & 2.4 \end{bmatrix}^\top$

- $\boldsymbol{w}_{cat} \cdot \boldsymbol{x} = -3.288$
  $p_{cat} = \dfrac{1}{1 + e^{3.288}} = 0.0360$
- $\boldsymbol{w}_{horse} \cdot \boldsymbol{x} = 6.4$
  $p_{horse} = \dfrac{1}{1 + e^{-6.4}} = 0.998$
- $\boldsymbol{w}_{elephant} \cdot \boldsymbol{x} = -657.44$
  $p_{elephant} = \dfrac{1}{1 + e^{657.44}} \approx 0$

Q4. Logistic Regression for Multi-Class Classification

(a) Compute the probability of an animal belonging to a certain class and classify them accordingly.

$$\mathbf{w}_{cat} = \begin{bmatrix} 4.2 & -0.01 & -0.12 \end{bmatrix}^\top$$
$$\mathbf{w}_{horse} = \begin{bmatrix} -20 & -0.08 & 35 \end{bmatrix}^\top$$
$$\mathbf{w}_{elephant} = \begin{bmatrix} -1250 & 0.82 & 0.9 \end{bmatrix}^\top$$

Weight (kg)	Length (m)
4.2	0.4
720	2.4
2350	5.5

Third animal: $\mathbf{x} = \begin{bmatrix} 1 & 2350 & 5.5 \end{bmatrix}^\top$

- $\mathbf{w}_{cat} \cdot \mathbf{x} = -19.96$
  $$p_{cat} = \frac{1}{1 + e^{19.96}} \approx 0$$
- $\mathbf{w}_{horse} \cdot \mathbf{x} = -15.5$
  $$p_{horse} = \frac{1}{1 + e^{15.5}} \approx 0$$
- $\mathbf{w}_{elephant} \cdot \mathbf{x} = 681.95$
  $$p_{elephant} = \frac{1}{1 + e^{-681.95}} \approx 1$$

Q4. Logistic Regression for Multi-Class Classification

(b) What if we want to extend the classification task to classify other animals? Can we train a new model while keeping the weights of the previous models?

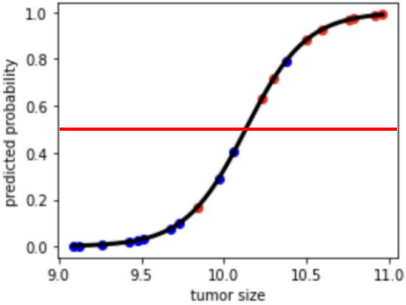
- For an animal that are very distinct with the three animals, we can create a new logistic regression model without changing the previous weights.
- For classifying a new animal that is similar with one of the classes (e.g, classifying a dog), we need to retrain the old models.

Q3. Precision, recall, F1 score and ROC curve

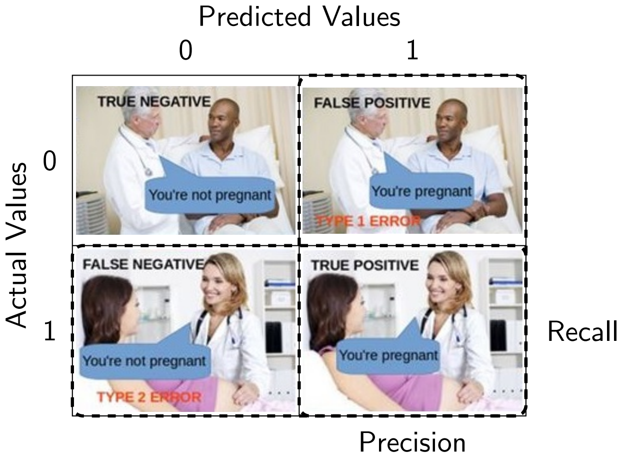
(a) For the threshold $p = 0.5$, come up with the confusion matrix.

Actual Values \ Predicted Values	0	1
0	10	1
1	1	8

Model M outputs label 1 if $M(x)$ is greater than or equal to the threshold, otherwise the model outputs 0.



Q3. Precision, recall, F1 score and ROC curve



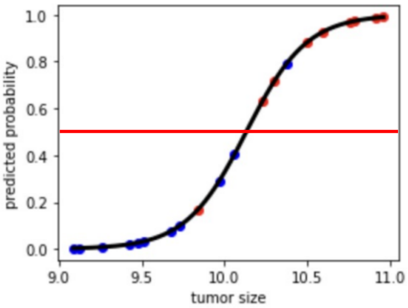
Q3. Precision, recall, F1 score and ROC curve

(b) For the threshold $p = 0.5$, find the precision, recall and F1 score.

Model M outputs label 1 if $M(x)$ is greater than or equal to the threshold, otherwise the model outputs 0.

		Predicted Values 0	Predicted Values 1
Actual Values	0	10	1
	1	1	8

- Precision $= \dfrac{TP}{TP + FP} = \dfrac{8}{8+1} = \dfrac{8}{9}$
- Recall $= \dfrac{TP}{TP + FN} = \dfrac{8}{8+1} = \dfrac{8}{9}$
- F1 Score $= \dfrac{2}{\frac{1}{P} + \frac{1}{R}} = \dfrac{2}{\frac{8}{9} + \frac{8}{9}} = \dfrac{8}{9}$



Q3. Precision, recall, F1 score and ROC curve

(d) In this question's case for detecting tumours, should we maximize precision or recall? Explain the reason for your choice.

- If cancer detection is being performed as a regular check up, then precision should be maximized; as we do not want to start cancer treatment on a person unless we are sure that he has cancer.
- If cancer detection is being performed as part of cancer treatment progress monitoring, then recall should be maximized; as we do not want to stop the ongoing treatment unless we are sure that there is no residual tumour cell left in the patient.

Q5. Evaluating Logistic Regression

Which of the following evaluation metrics is the least appropriate when comparing a logistic regression model's output with the target label? Explain your answer.

(a) Accuracy
(b) Precision, Recall
(c) Binary Cross Entropy Loss
(d) Mean Squared Error (MSE)

Q5. Evaluating Logistic Regression

Evaluation Metric:

- Judges the performance, doesn't care about the process.

[MRQ] Which of the following link(s) are pruned? Shade <u>all</u> that is/are true.
◯ a ◯ b ◯ c ◯ d ◯ e
◯ f ◯ g ◯ h ◯ i ◯ j
◯ k ◯ l ◉ m ✓ ◯ n ◯ o
◯ p ◯ q ◉ r ✗ ◯ s ◉ t ✓
◯ u ◯ v ◯ w ◯ x ◯ y
◯ z 0 / 4

Loss Function:

- Helps with model training. Minimized by the optimizer.

[MRQ] Which of the following link(s) are pruned? Shade <u>all</u> that is/are true.
◯ a ◯ b ◯ c ◯ d ◯ e
◯ f ◯ g ◯ h ◯ i ◯ j
◯ k ◯ l ◉ m ✓ ◯ n ◯ o
◯ p ◯ q ◉ r ✗ ◯ s ◉ t ✓
◯ u ◯ v ◯ w ◯ x ◯ y
◯ z 25/26 items right 👍 Try again!

Which options are suitable evaluation metrics?

Q5. Evaluating Logistic Regression

Evaluation metrics:

(a) Accuracy
Gauging a model's overall performance.

(b) Precision, Recall
Quantifies the model's ability to distinguish between positive and negative classes effectively.

Loss functions:

(c) Binary Cross Entropy Loss
More suitable for **classification** tasks (assumes the binomial distribution).

(d) Mean Squared Error
More suitable for **regression** tasks (assumes the normal distribution).

Answer: (b) $>$ (a) $>$ (c) $>$ (d).

Bonus. Sigmoid vs Softmax

We use the sigmoid function for logistic regression in lecture:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

In multi-class logistic regression, we often use the softmax function instead:

$$\text{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

(a) Show that sigmoid function is a special case of the softmax function.

(b) Express the derivative $\text{softmax}'(\boldsymbol{z})_{ik}$ in terms of $\text{softmax}(\boldsymbol{z})$.

(c) Under what scenarios would you consider using softmax function instead of the sigmoid function?

Bonus. Sigmoid vs Softmax

Extra Slide

Solution.

(a) When $K = 2$,

$$
\begin{aligned}
\operatorname{softmax}(\boldsymbol{z}) &= \begin{bmatrix} \dfrac{e^{z_1}}{e^{z_1}+e^{z_2}} & \dfrac{e^{z_2}}{e^{z_1}+e^{z_2}} \end{bmatrix}^\top \\
&= \begin{bmatrix} \dfrac{e^{z_1-z_2}}{e^{z_1-z_2}+1} & \dfrac{e^{z_2-z_1}}{1+e^{z_2-z_1}} \end{bmatrix}^\top \\
&= \begin{bmatrix} \sigma(z_1-z_2) & \sigma(z_2-z_1) \end{bmatrix}^\top
\end{aligned}
$$

which can be replaced by logistic regression where $z = z_1 - z_2$, predicting the probability of class 1.

```
def mysoftmax(z):
    softmax_class0 = torch.sigmoid(z[:, 0:1] - z[:, 1:2])
    return torch.hstack([softmax_class0, 1 - softmax_class0])
```

Bonus. Sigmoid vs Softmax

Extra Slide

(b) Define δ_{ik} as 1 if $i = k$, 0 otherwise.

$$
\begin{aligned}
\text{softmax}'(\boldsymbol{z})_{ik} &= \frac{\delta_{ik} e^{z_i} \left(\sum_{j=1}^{K} e^{z_j} \right) - e^{z_i} e^{z_k}}{\left(\sum_{j=1}^{K} e^{z_j} \right)^2} \\
&= \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \cdot \left(\delta_{ik} - \frac{e_{z_k}}{\sum_{j=1}^{K} e^{z_j}} \right) \\
&= \text{softmax}(\boldsymbol{z})_i \cdot (\delta_{ik} - \text{softmax}(\boldsymbol{z})_k)
\end{aligned}
$$

Notice the similarity with the sigmoid function!

```python
def mysoftmax_grad(z):
    n, m = z.shape
    z = torch.softmax(z, dim=1)
    return z.reshape(n, m, 1)
            * (torch.eye(m).reshape(1, m, m) - z.reshape(n, 1, m))
```

Bonus. Sigmoid vs Softmax

(c) The softmax function ensures that all output probabilities sums up to 1. It is a good idea to use the softmax function if the classes are mutually exclusive. On the other hand, use the sigmoid function if the classes are independent events.