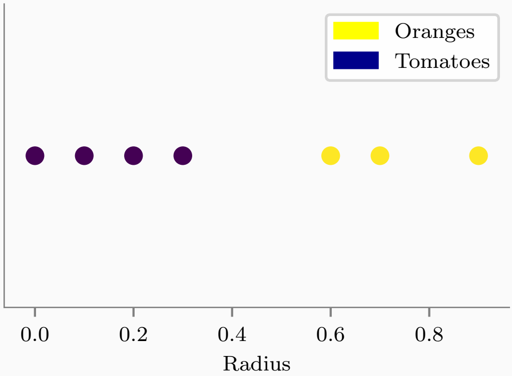
Logistic Regression

Nipun Batra

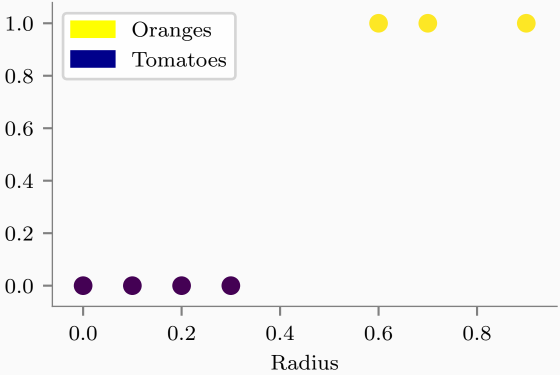
March 14, 2023

IIT Gandhinagar

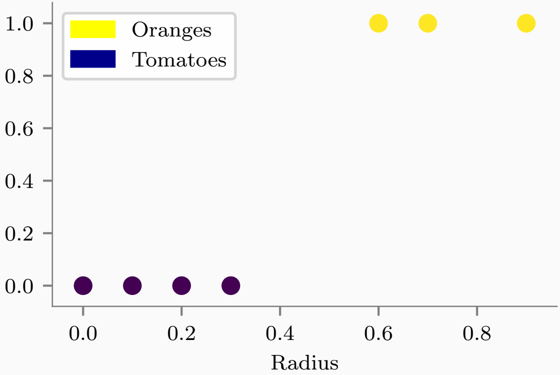
Classification Technique



Classification Technique

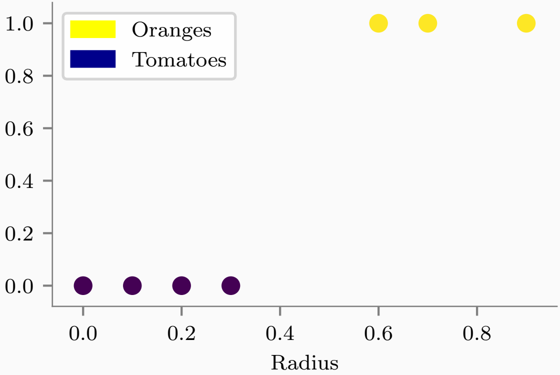


Classification Technique



Aim: Probability(Orange | Radius) ? or

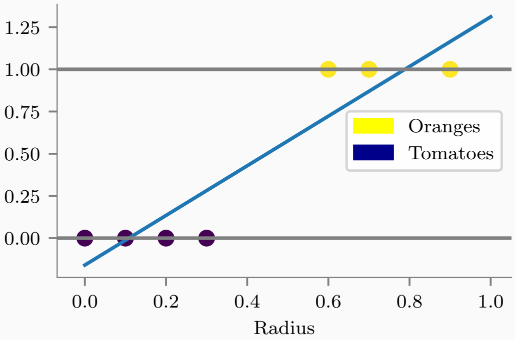
Classification Technique



Aim: Probability(Orange | Radius) ? or

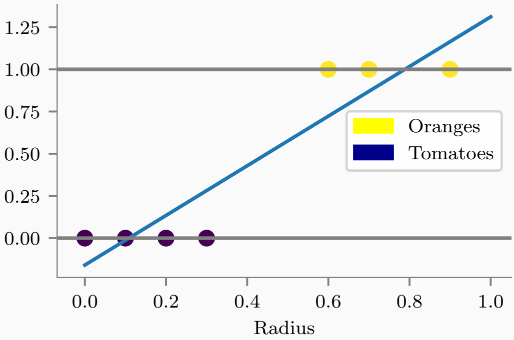
$\boxed{\text{More generally, } P(y = 1|X = x)?}$

Idea: Use Linear Regression



$$P(y = Orange|Radius = r) = \theta_0 + \theta_1 \times r$$

Idea: Use Linear Regression

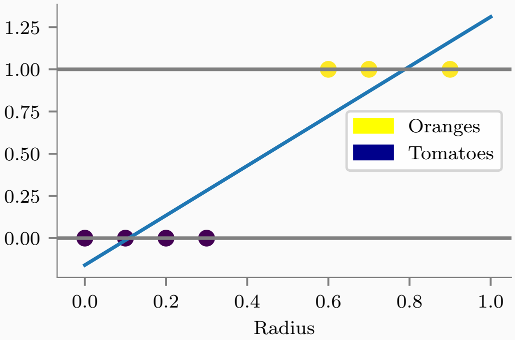


$$P(y = Orange|Radius = r) = \theta_0 + \theta_1 \times r$$

Generally,

$$P(y = 1|x) = x^T\theta$$

Idea: Use Linear Regression



$$P(y = Orange|Radius = r) = \theta_0 + \theta_1 \times r$$

Generally,

$$P(y = 1|x) = x^T\theta$$

For N examples

$$P(\mathbf{y} = 1|X) = X\theta$$

Idea: Use Linear Regression

Prediction:

If $\theta_0 + \theta_1 \times \mathit{Radius} > 0.5 \rightarrow$ Orange

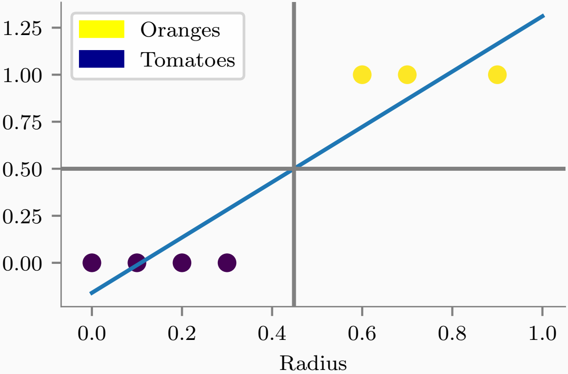
Else $\rightarrow$ Tomato

Problem:

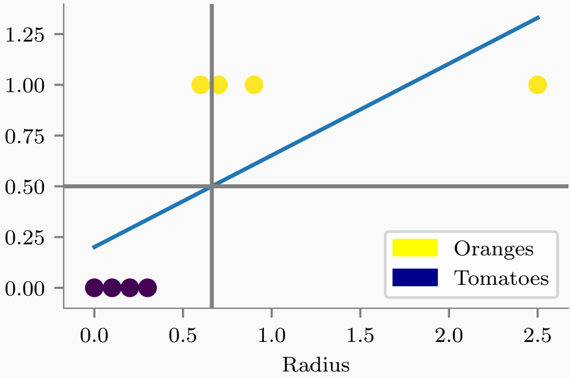
Range of $X\theta$ is $(-\infty, \infty)$

But $P(y = 1|\ldots) \in [0, 1]$

Idea: Use Linear Regression

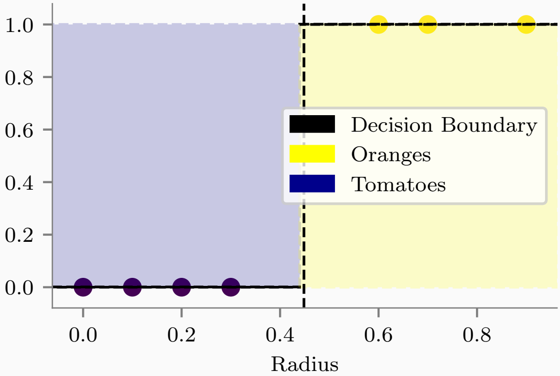


Idea: Use Linear Regression



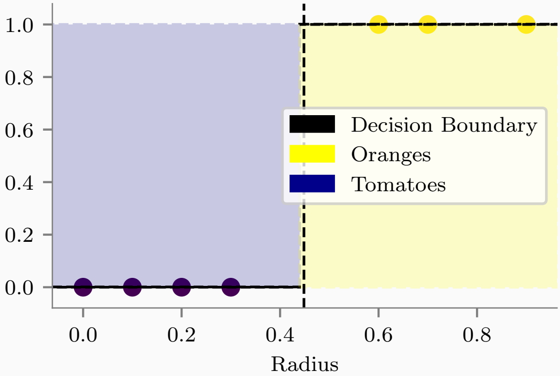
Linear regression for classification gives a poor prediction!

Ideal boundary



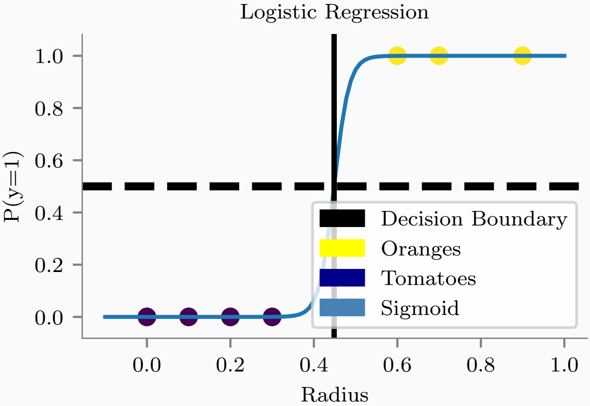
- Have a decision function similar to the above (but not so sharp and discontinuous)

Ideal boundary



- Have a decision function similar to the above (but not so sharp and discontinuous)
- Aim: use linear regression still!

Idea: Use Linear Regression



Question. Can we still use Linear Regression?

Answer. Yes! Transform $x^T\theta \to [0, 1]$

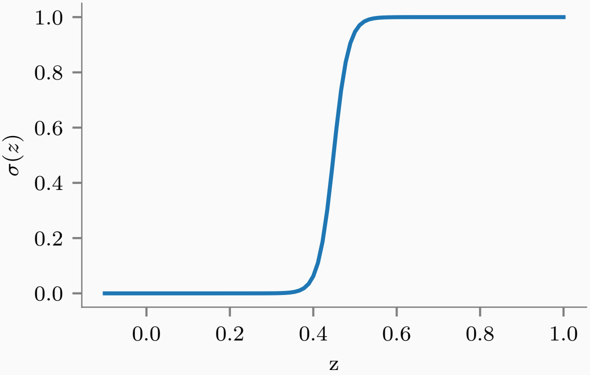
Logistic / Sigmoid Function

$x^T\theta \in (-\infty, \infty)$

$\phi =$ Sigmoid / Logistic Function (σ)

$\phi(x^T\theta) \in [0, 1]$

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



Logistic / Sigmoid Function

$z \to \infty$

Logistic / Sigmoid Function

$z \to \infty$

$\sigma(z) \to 1$

Logistic / Sigmoid Function

$z \to \infty$

$\sigma(z) \to 1$

$z \to -\infty$

Logistic / Sigmoid Function

$z \to \infty$

$\sigma(z) \to 1$

$z \to -\infty$

$\sigma(z) \to 0$

Logistic / Sigmoid Function

$z \to \infty$

$\sigma(z) \to 1$

$z \to -\infty$

$\sigma(z) \to 0$

$z = 0$

Logistic / Sigmoid Function

$$z \to \infty$$
$$\sigma(z) \to 1$$
$$z \to -\infty$$
$$\sigma(z) \to 0$$
$$z = 0$$
$$\sigma(z) = 0.5$$

Logistic / Sigmoid Function

Question. Could you use some other transformation (ϕ) of $x^T\theta$ s.t.

$$\phi(x^T\theta) \in [0, 1]$$

Yes! But Logistic Regression works.

Logistic / Sigmoid Function

$$P(y = 1|X) = \sigma(X\theta) = \frac{1}{1 + e^{-X\theta}}$$

Q. Write $X\theta$ in a more convenient form (as $P(y = 1|X)$, $P(y = 0|X)$)

Logistic / Sigmoid Function

$$P(y=1|X) = \sigma(X\theta) = \frac{1}{1+e^{-X\theta}}$$

Q. Write $X\theta$ in a more convenient form (as $P(y=1|X)$, $P(y=0|X)$)

Logistic / Sigmoid Function

$$P(y=1|X) = \sigma(X\theta) = \frac{1}{1+e^{-X\theta}}$$

Q. Write $X\theta$ in a more convenient form (as $P(y=1|X)$, $P(y=0|X)$)

$$P(y=0|X) = 1 - P(y=1|X) = 1 - \frac{1}{1+e^{-X\theta}} = \frac{e^{-X\theta}}{1+e^{-X\theta}}$$

Logistic / Sigmoid Function

$$P(y = 1|X) = \sigma(X\theta) = \frac{1}{1 + e^{-X\theta}}$$

Q. Write $X\theta$ in a more convenient form (as $P(y = 1|X)$, $P(y = 0|X)$)

$$P(y = 0|X) = 1 - P(y = 1|X) = 1 - \frac{1}{1 + e^{-X\theta}} = \frac{e^{-X\theta}}{1 + e^{-X\theta}}$$

$$\therefore \frac{P(y = 1|X)}{1 - P(y = 1|X)} = e^{X\theta} \implies X\theta = \log \frac{P(y = 1|X)}{1 - P(y = 1|X)}$$

Odds (Used in betting)

$$\frac{P(win)}{P(loss)}$$

Here,

$$Odds = \frac{P(y = 1)}{P(y = 0)}$$

$$\boxed{\text{logits} = \text{log-odds} = \log \frac{P(y=1)}{P(y=0)} = X\theta}$$

Logits Usage

Docs > torch.nn > CrossEntropyLoss

CROSSENTROPYLOSS

CLASS torch.nn.CrossEntropyLoss(*weight=None*, *size_average=None*, *ignore_index=- 100*, *reduce=None*, *reduction='mean'*, *label_smoothing=0.0*) [SOURCE]

This criterion computes the cross entropy loss between input logits and target.

Logits Usage

Computes the cross-entropy loss between true labels and predicted labels.

Inherits From: `Loss`

View aliases

```
tf.keras.losses.BinaryCrossentropy(
    from_logits=False,
    label_smoothing=0.0,
    axis=-1,
    reduction=losses_utils.ReductionV2.AUTO,
    name='binary_crossentropy'
)
```

Used in the notebooks

Used in the guide	Used in the tutorials
• The Functional API • Transfer learning and fine-tuning • Distributed training with TensorFlow • Estimators • Making new Layers and Models via subclassing	• Load a pandas DataFrame • Transfer learning and fine-tuning • Basic text classification • Warm-start embedding layer matrix • Parameter server training with ParameterServerStrategy

Use this cross-entropy loss for binary (0 or 1) classification applications. The loss function requires the following inputs:

- `y_true` (true label): This is either 0 or 1.
- `y_pred` (predicted value): This is the model's prediction, i.e, a single floating-point value which either represents a logit, (i.e, value in [-inf, inf] when `from_logits=True`) or a probability (i.e, value in [0., 1.] when `from_logits=False`).

Recommended Usage: (set `from_logits=True`)

Logits Usage

TensorFlow > Resources > Probability > API

Was this helpful?

tfp.distributions.Bernoulli

View source on GitHub

Bernoulli distribution.

Inherits From: `Distribution`, `AutoCompositeTensor`

```
tfp.distributions.Bernoulli(
    logits=None,
    probs=None,
    dtype=tf.int32,
    validate_args=False,
    allow_nan_stats=True,
    name='Bernoulli'
)
```

The Bernoulli distribution with `probs` parameter, i.e., the probability of a `1` outcome (vs a `0` outcome).

Args	
`logits`	An N-D `Tensor` representing the log-odds of a `1` event. Each entry in the `Tensor` parameterizes an independent Bernoulli distribution where the probability of an event is sigmoid(logits). Only one of `logits` or `probs` should be passed in.
`probs`	An N-D `Tensor` representing the probability of a `1` event. Each entry in the `Tensor` parameterizes an independent Bernoulli distribution. Only one of `logits` or `probs` should be passed in.

Q. What is decision boundary for Logistic Regression?

Q. What is decision boundary for Logistic Regression?

Decision Boundary: $P(y=1|X) = P(y=0|X)$

$$\text{or } \frac{1}{1+e^{-X\theta}} = \frac{e^{-X\theta}}{1+e^{-X\theta}}$$

$$\text{or } e^{X\theta} = 1$$

$$\text{or } X\theta = 0$$

Example with 2 input features

Notebook:
https://nipunbatra.github.io/ml2023/notebooks/posts/logistic.html

Learning Parameters

Could we use cost function as:

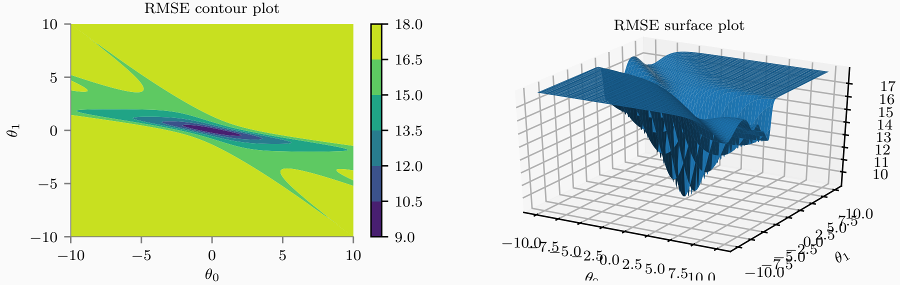
$$J(\theta) = \sum (y_i - \hat{y}_i)^2$$

$$\hat{y}_i = \sigma(X\theta)$$

Answer: No (Non-Convex)

(See Jupyter Notebook)

Cost function convexity



Learning Parameters

Likelihood $= P(D|\theta)$

$P(y|X,\theta) = \prod_{i=1}^{n} P(y_i|x_i,\theta)$
where y = 0 or 1

Learning Parameters

Likelihood $= P(D|\theta)$

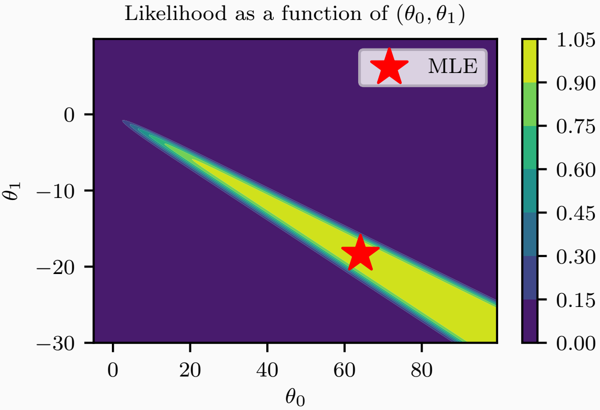
$$
\begin{aligned}
P(y|X,\theta) &= \prod_{i=1}^{n} P(y_i|x_i,\theta) \\
&= \prod_{i=1}^{n} \left\{\frac{1}{1+e^{-x_i^T\theta}}\right\}^{y_i} \left\{1 - \frac{1}{1+e^{-x_i^T\theta}}\right\}^{1-y_i}
\end{aligned}
$$

[Above: Similar to $P(D|\theta)$ for Linear Regression;

Difference Bernoulli instead of Gaussian]

$$
\begin{aligned}
-\log P(y|X,\theta) &= \text{Negative Log Likelihood} \\
&= \text{Cost function will be minimising} \\
&= J(\theta)
\end{aligned}
$$

Likelihood Visualisation



Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).

Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).
- What is p(H)?

Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).
- What is p(H)?
- We might think it to be: $4/10 = 0.4$. But why?

Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).
- What is p(H)?
- We might think it to be: $4/10 = 0.4$. But why?
- Answer 1: Probability defined as a measure of long running frequencies

Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).
- What is p(H)?
- We might think it to be: $4/10 = 0.4$. But why?
- Answer 1: Probability defined as a measure of long running frequencies
- Answer 2: What is likelihood of seeing the above sequence when the p(Head)$=\theta$?

Aside on Bernoulli Likelihood

- Assume you have a coin and flip it ten times and get (H, H, T, T, T, H, H, T, T, T).
- What is p(H)?
- We might think it to be: $4/10 = 0.4$. But why?
- Answer 1: Probability defined as a measure of long running frequencies
- Answer 2: What is likelihood of seeing the above sequence when the p(Head)$=\theta$?
- Idea find MLE estimate for θ

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$
- What is $P(D_1, D_2, ..., D_n|\theta)$?

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$
- What is $P(D_1, D_2, ..., D_n|\theta)$?
- $P(D_1, D_2, ..., D_n|\theta) = P(D_1|\theta)P(D_2|\theta)...P(D_n|\theta)$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$
- What is $P(D_1, D_2, ..., D_n|\theta)$?
- $P(D_1, D_2, ..., D_n|\theta) = P(D_1|\theta)P(D_2|\theta)...P(D_n|\theta)$
- $P(D_1, D_2, ..., D_n|\theta) = \theta^{n_h}(1-\theta)^{n_t}$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$
- What is $P(D_1, D_2, ..., D_n|\theta)$?
- $P(D_1, D_2, ..., D_n|\theta) = P(D_1|\theta)P(D_2|\theta)...P(D_n|\theta)$
- $P(D_1, D_2, ..., D_n|\theta) = \theta^{n_h}(1-\theta)^{n_t}$
- Log-likelihood $= \mathcal{LL}(\theta) = n_h \log(\theta) + n_t \log(1-\theta)$

Aside on Bernoulli Likelihood

- $p(H) = \theta$ and $p(T) = 1 - \theta$
- What is the PMF for first observation $P(D_1 = x|\theta)$, where x = 0 for Tails and x = 1 for Heads?
- $P(D_1 = x|\theta) = \theta^x(1-\theta)^{(1-x)}$
- Verify the above: if x = 0 (Tails), $P(D_1 = x|\theta) = 1 - \theta$ and if x = 1 (Heads), $P(D_1 = x|\theta) = \theta$
- What is $P(D_1, D_2, ..., D_n|\theta)$?
- $P(D_1, D_2, ..., D_n|\theta) = P(D_1|\theta)P(D_2|\theta)...P(D_n|\theta)$
- $P(D_1, D_2, ..., D_n|\theta) = \theta^{n_h}(1-\theta)^{n_t}$
- Log-likelihood $= \mathcal{LL}(\theta) = n_h \log(\theta) + n_t \log(1-\theta)$
- $\frac{\partial \mathcal{LL}(\theta)}{\partial \theta} = 0 \implies \frac{n_h}{\theta} + \frac{n_t}{1-\theta} = 0 \implies \theta_{MLE} = \frac{n_h}{n_h + n_t}$

Learning Parameters

$$J(\theta) = -\log \left\{ \prod_{i=1}^{n} \left\{ \frac{1}{1 + e^{-x_i^T \theta}} \right\}^{y_i} \left\{ 1 - \frac{1}{1 + e^{-x_i^T \theta}} \right\}^{1-y_i} \right\}$$

$$J(\theta) = -\left\{ \sum_{i=1}^{n} y_i \log(\sigma_\theta(x_i)) + (1 - y_i) \log(1 - \sigma_\theta(x_i)) \right\}$$

Learning Parameters

$$J(\theta) = -\log\left\{\prod_{i=1}^{n}\left\{\frac{1}{1+e^{-x_i^T\theta}}\right\}^{y_i}\left\{1-\frac{1}{1+e^{-x_i^T\theta}}\right\}^{1-y_i}\right\}$$

$$J(\theta) = -\left\{\sum_{i=1}^{n} y_i\log(\sigma_\theta(x_i)) + (1-y_i)\log(1-\sigma_\theta(x_i))\right\}$$

This cost function is called cross-entropy.

Learning Parameters

$$J(\theta) = -\log\left\{\prod_{i=1}^{n}\left\{\frac{1}{1+e^{-x_i^T\theta}}\right\}^{y_i}\left\{1-\frac{1}{1+e^{-x_i^T\theta}}\right\}^{1-y_i}\right\}$$

$$J(\theta) = -\left\{\sum_{i=1}^{n} y_i\log(\sigma_\theta(x_i)) + (1-y_i)\log(1-\sigma_\theta(x_i))\right\}$$

This cost function is called cross-entropy.

Why?

Interpretation of Cross-Entropy Cost Function

Interpretation of Cross-Entropy Cost Function

What is the interpretation of the cost function?

Interpretation of Cross-Entropy Cost Function

What is the interpretation of the cost function?

Let us try to write the cost function for a single example:

Interpretation of Cross-Entropy Cost Function

What is the interpretation of the cost function?

Let us try to write the cost function for a single example:

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

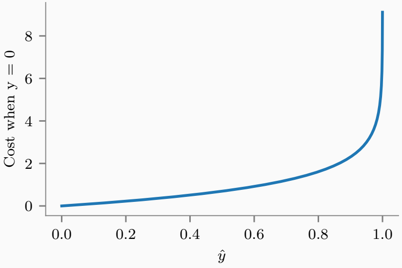
Interpretation of Cross-Entropy Cost Function

What is the interpretation of the cost function?

Let us try to write the cost function for a single example:

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

First, assume y_i is 0, then if $\hat{y}_i$ is 0, the loss is 0; but, if $\hat{y}_i$ is 1, the loss tends towards infinity!



Interpretation of Cross-Entropy Cost Function

Interpretation of Cross-Entropy Cost Function

What is the interpretation of the cost function?

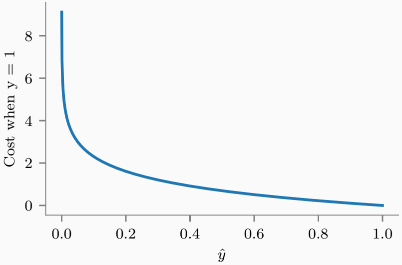
$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

Interpretation of Cross-Entropy Cost Function

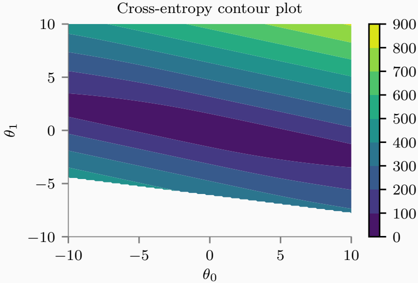
What is the interpretation of the cost function?

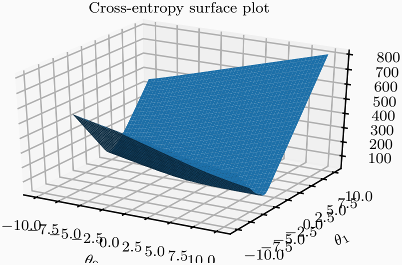
$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

Now, assume y_i is 1, then if $\hat{y}_i$ is 0, the loss is huge; but, if $\hat{y}_i$ is 1, the loss is zero!



Cost function convexity





Learning Parameters

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= -\frac{\partial}{\partial \theta_j}\left\{\sum_{i=1}^{n} y_i log(\sigma_\theta(x_i)) + (1 - y_i) log(1 - \sigma_\theta(x_i))\right\} \\
&= -\sum_{i=1}^{n}\left[y_i \frac{\partial}{\partial \theta_j}\log(\sigma_\theta(x_i)) + (1 - y_i)\frac{\partial}{\partial \theta_j} log(1 - \sigma_\theta(x_i))\right]
\end{aligned}
$$

Learning Parameters

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{i=1}^{n} \left[y_i \frac{\partial}{\partial \theta_j} \log(\sigma_\theta(x_i)) + (1 - y_i) \frac{\partial}{\partial \theta_j} log(1 - \sigma_\theta(x_i)) \right]$$

$$= -\sum_{i=1}^{n} \left[\frac{y_i}{\sigma_\theta(x_i)} \frac{\partial}{\partial \theta_j} \sigma_\theta(x_i) + \frac{1 - y_i}{1 - \sigma_\theta(x_i)} \frac{\partial}{\partial \theta_j} (1 - \sigma_\theta(x_i)) \right] \quad (1)$$

Aside:

$$\frac{\partial}{\partial z} \sigma(z) = \frac{\partial}{\partial z} \frac{1}{1 + e^{-z}} = -(1 + e^{-z})^{-2} \frac{\partial}{\partial z} (1 + e^{-z})$$

$$= \frac{e^{-z}}{(1 + e^{-z})^2} = \left(\frac{1}{1 + e^{-z}} \right) \left(\frac{e^{-z}}{1 + e^{-z}} \right) = \sigma(z) \left\{ \frac{1 + e^{-z}}{1 + e^{-z}} - \frac{1}{1 + e^{-z}} \right\}$$

$$= \sigma(z)(1 - \sigma(z))$$

Learning Parameters

Resuming from (1)

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{i=1}^{n}\left[\frac{y_i}{\sigma_\theta(x_i)}\frac{\partial}{\partial \theta_j}\sigma_\theta(x_i) + \frac{1-y_i}{1-\sigma_\theta(x_i)}\frac{\partial}{\partial \theta_j}(1-\sigma_\theta(x_i))\right]$$

$$= -\sum_{i=1}^{n}\left[\frac{y_i\sigma_\theta(x_i)}{\sigma_\theta(x_i)}(1-\sigma_\theta(x_i))\frac{\partial}{\partial \theta_j}(x_i{}^T\theta) + \frac{1-y_i}{1-\sigma_\theta(x_i)}(1-\sigma_\theta(x_i))\frac{\partial}{\partial \theta_j}(1-\sigma_\theta(x_i))\right]$$

$$= -\sum_{i=1}^{n}\left[y_i(1-\sigma_\theta(x_i))x_i^j - (1-y_i)\sigma_\theta(x_i)x_i^j\right]$$

$$= -\sum_{i=1}^{n}\left[(y_i - y_i\sigma_\theta(x_i) - \sigma_\theta(x_i) + y_i\sigma_\theta(x_i))x_i^j\right]$$

$$= \sum_{i=1}^{n}\left[\sigma_\theta(x_i) - y_i\right]x_i^j$$

Learning Parameters

$$\boxed{\frac{\partial J(\theta)}{\theta_j} = \sum_{i=1}^{N} \left[\sigma_\theta(x_i) - y_i\right] x_i^j}$$

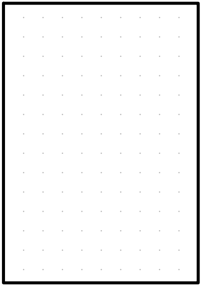
$$\boxed{\frac{\partial J(\theta)}{\theta_j} = \sum_{i=1}^{N} \left[\hat{y}_i - y_i\right] x_i^j}$$

Now, just use Gradient Descent!

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j$$

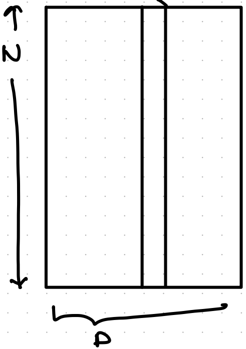
$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i)\, x_i^j$$

MATRIX X



$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j$$

MATRIX X



j^{th} column of X

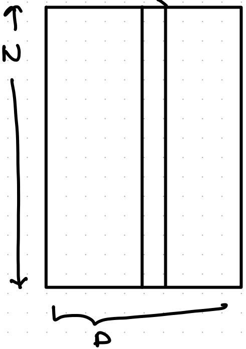
$= X^j$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j$$

MATRIX X

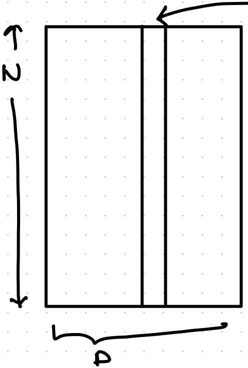
j^{th} column of X

$= X^j$



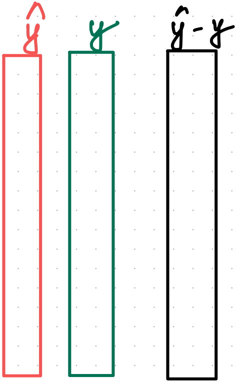
$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j$$

MATRIX X

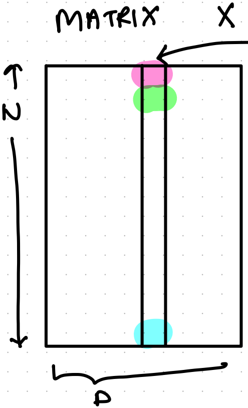


j^{th} column of X

$= X^j$

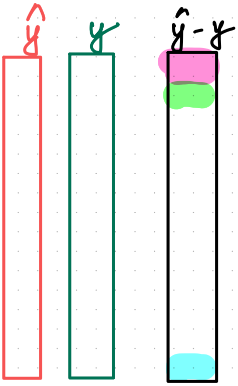


$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j$$

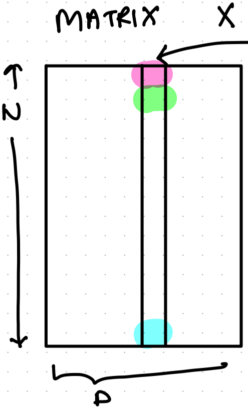


j^{th} column of X

$= X^j$

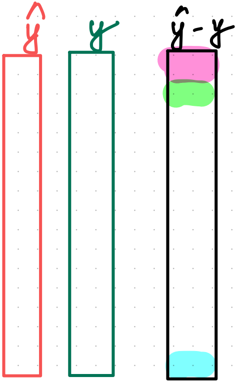


$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j = \boxed{ {x^j}^T_{1\times N} (\hat{y} - y) }$$



j^{th} column of X

$= X^j$



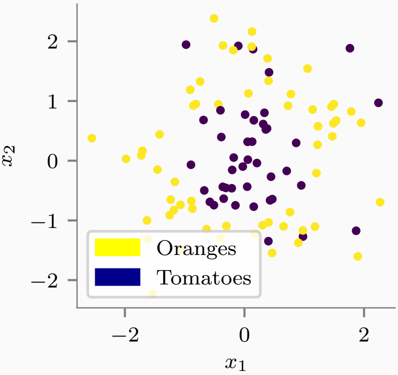
$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j = \underset{1\times N}{x^{j^T}} (\hat{y} - y)$$

$$\begin{bmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \frac{\partial J(\theta)}{\partial \theta_2} \\ \vdots \\ \\ \frac{\partial J(\theta)}{\partial \theta_D} \end{bmatrix} = \begin{pmatrix} x^{1^T} (\hat{y} - y) \\ x^{2^T} (\hat{y} - y) \\ \vdots \\ \\ x^{D^T} (\hat{y} - y) \end{pmatrix}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (\hat{y}_i - y_i) x_i^j = \boxed{\underset{1\times N}{x^{j^T}} (\hat{y} - y)}$$

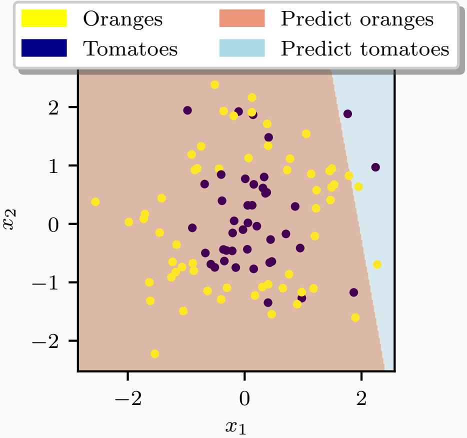
$$\begin{bmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \frac{\partial J(\theta)}{\partial \theta_2} \\ \vdots \\ \\ \frac{\partial J(\theta)}{\partial \theta_D} \end{bmatrix} = \begin{pmatrix} x^{1^T} (\hat{y} - y) \\ x^{2^T} (\hat{y} - y) \\ \vdots \\ x^{D^T} (\hat{y} - y) \end{pmatrix} = X^T (\hat{y} - y)$$

Logistic Regression with feature transformation



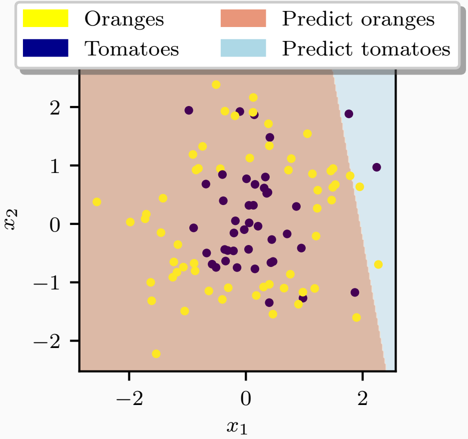
What happens if you apply logistic regression on the above data?

Logistic Regression with feature transformation



Linear boundary will not be accurate here. What is the technical name of the problem?

Logistic Regression with feature transformation

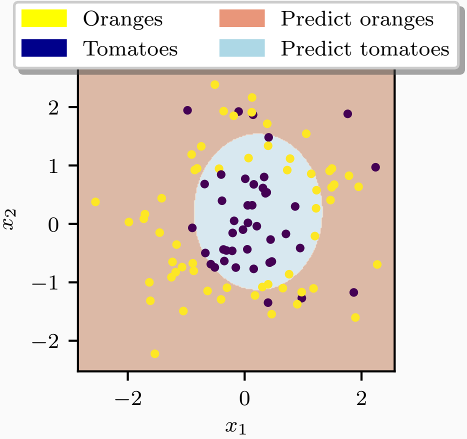


Linear boundary will not be accurate here. What is the technical name of the problem? Bias!

Logistic Regression with feature transformation

$$\phi(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K-1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K$$

Logistic Regression with feature transformation



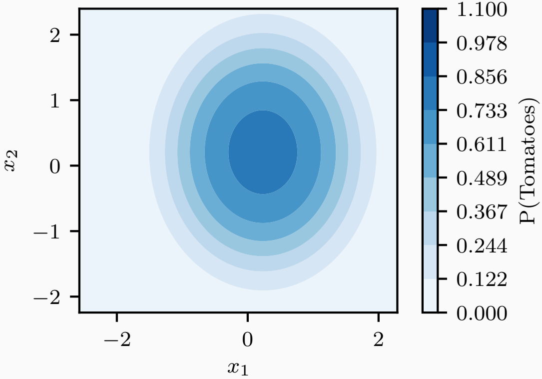
Using x_1^2, x_2^2 as additional features, we are able to learn a more accurate classifier.

Logistic Regression with feature transformation

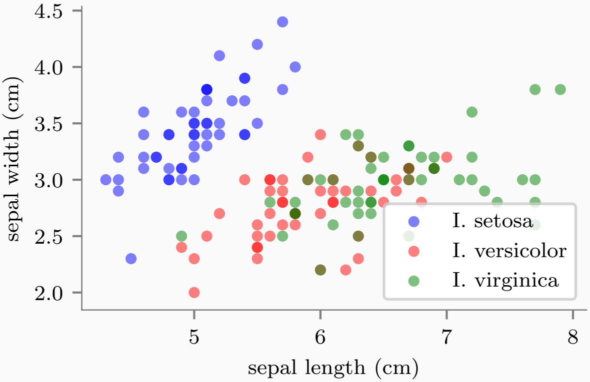
How would you expect the probability contours look like?

Logistic Regression with feature transformation

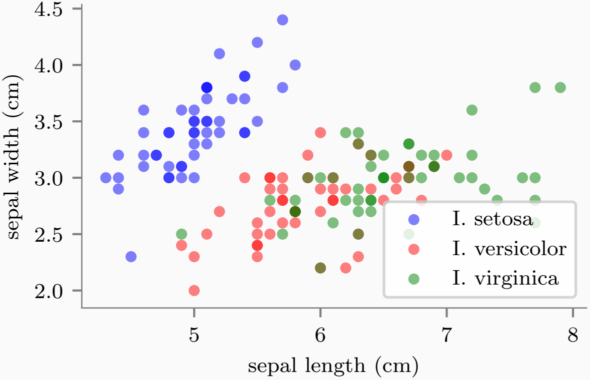
How would you expect the probability contours look like?



Multi-Class Prediction

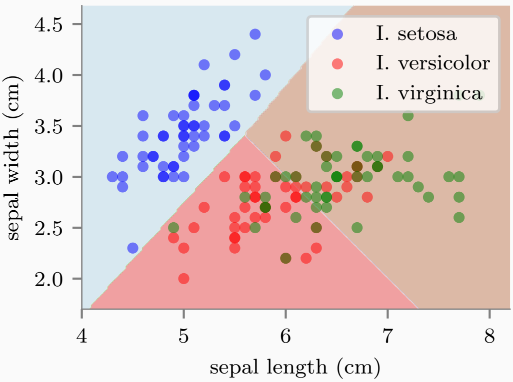


Multi-Class Prediction

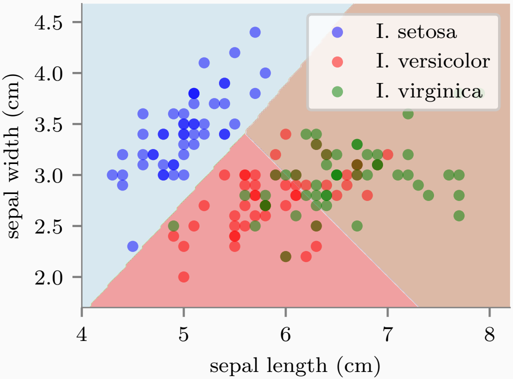


How would you learn a classifier? Or, how would you expect the classifier to learn decision boundaries?

Multi-Class Prediction

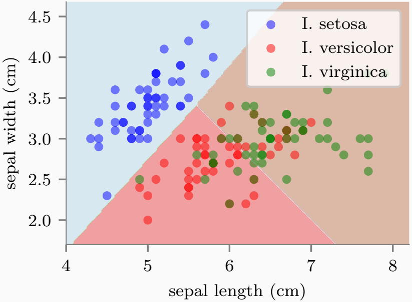


Multi-Class Prediction

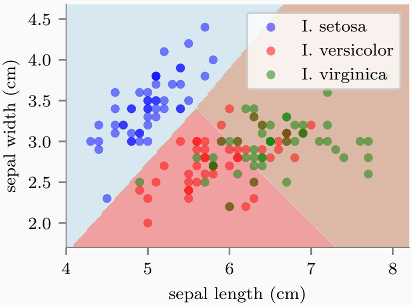


1. Use one-vs.-all on Binary Logistic Regression
2. Use one-vs.-one on Binary Logistic Regression
3. Extend Binary Logistic Regression to Multi-Class Logistic Regression

Multi-Class Prediction

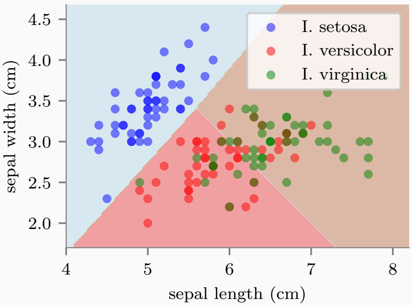


Multi-Class Prediction

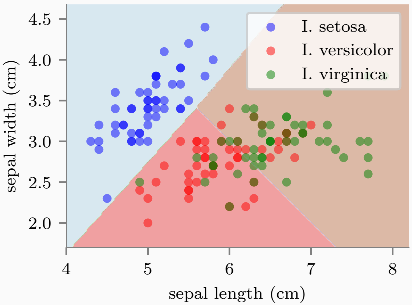


1. Learn P(setosa (class 1)) $= \mathcal{F}(X\theta_1)$
2. P(versicolor (class 2)) $= \mathcal{F}(X\theta_2)$
3. P(virginica (class 3)) $= \mathcal{F}(X\theta_3)$
4. Goal: Learn $\theta_i \forall i \in \{1, 2, 3\}$
5. Question: What could be an $\mathcal{F}$?

Multi-Class Prediction



Multi-Class Prediction



1. Question: What could be an $\mathcal{F}$?
2. Property: $\sum_{i=1}^{3} \mathcal{F}(X\theta_i) = 1$
3. Also $\mathcal{F}(z) \in [0, 1]$
4. Also, $\mathcal{F}(z)$ has squashing proprties: $R \mapsto [0, 1]$

Softmax

$$Z \in \mathbb{R}^d$$

$$\mathcal{F}(z_i) = \frac{e^{z_i}}{\sum_{i=1}^{d} e^{z_i}}$$

$$\therefore \sum \mathcal{F}(z_i) = 1$$

$\mathcal{F}(z_i)$ refers to <u>probability</u> of class <u>i</u>

Softmax for Multi-Class Logistic Regression

$$k = \{1, \ldots, K\}\text{classes}$$

$$\theta = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \theta_1 & \theta_2 & \cdots & \theta_K \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\boxed{\hat{y}^k = P(y = k | X, \theta) = \frac{e^{X\theta_k}}{\sum_{k=1}^{K} e^{X\theta_K}}}$$

Softmax for Multi-Class Logistic Regression

For K = 2 classes,

$$P(y = k|X, \theta) = \frac{e^{X\theta_k}}{\sum_{k=1}^{K} e^{X\theta_k}}$$

$$P(y = 0|X, \theta) = \frac{e^{X\theta_0}}{e^{X\theta_0} + e^{X\theta_1}}$$

$$P(y = 1|X, \theta) = \frac{e^{X\theta_1}}{e^{X\theta_0} + e^{X\theta_1}} = \frac{e^{X\theta_1}}{e^{X\theta_1}\{1 + e^{X(\theta_0 - \theta_1)}\}}$$

$$= \frac{1}{1 + e^{-X\theta'}}$$

$$= \text{Sigmoid!}$$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.1 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.1 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.1 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

$= -(0 \times \log(0.1) + 1 \times \log(0.8) + 0 \times \log(0.1))$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.1 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

$= -(0 \times \log(0.1) + 1 \times \log(0.8) + 0 \times \log(0.1))$

Tends to zero

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class #2

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class #2

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

$= -(0 \times \log(0.1) + 1 \times \log(0.4) + 0 \times \log(0.1))$

Multi-Class Logistic Regression Cost

Assume our prediction and ground truth for the three classes for i^{th} point is:

$$\hat{y}_i = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix} = \begin{bmatrix} \hat{y}_i^1 \\ \hat{y}_i^2 \\ \hat{y}_i^3 \end{bmatrix}$$

$$y_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix}$$

meaning the true class is Class $\#2$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

$= -(0 \times \log(0.1) + 1 \times \log(0.4) + 0 \times \log(0.1))$

High number! Huge penalty for misclassification!

Multi-Class Logistic Regression Cost

For 2 class we had:

$$J(\theta) = -\left\{ \sum_{i=1}^{n} y_i \log(\sigma_\theta(x_i)) + (1 - y_i) \log(1 - \sigma_\theta(x_i)) \right\}$$

Multi-Class Logistic Regression Cost

For 2 class we had:

$$J(\theta) = -\left\{ \sum_{i=1}^{n} y_i \log(\sigma_\theta(x_i)) + (1 - y_i) \log(1 - \sigma_\theta(x_i)) \right\}$$

More generally,

Multi-Class Logistic Regression Cost

For 2 class we had:

$$J(\theta) = -\left\{ \sum_{i=1}^{n} y_i \log(\sigma_\theta(x_i)) + (1 - y_i) \log(1 - \sigma_\theta(x_i)) \right\}$$

More generally,

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

Extend to K-class:

$$J(\theta) = -\left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k) \right\}$$

Multi-Class Logistic Regression Cost Gradient

$$J(\theta) = -\left\{ \sum_{i=1}^{n} \sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k) \right\}$$

$$\nabla_{\theta k} J(\theta) = \sum_{i=1}^{N} (\hat{y}_i^k - y_i^k) x_i$$

Hessian Matrix

The Hessian matrix of f(.) with respect to θ, written $\nabla_\theta^2 f(\theta)$ or simply as $\mathbb{H}$, is the $d \times d$ matrix of partial derivatives,

$$\nabla_\theta^2 f(\theta) = \begin{bmatrix} \frac{\partial^2 f(\theta)}{\partial \theta_1^2} & \frac{\partial^2 f(\theta)}{\partial \theta_1 \partial \theta_2} & \dots & \frac{\partial^2 f(\theta)}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2 f(\theta)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 f(\theta)}{\partial \theta_2^2} & \dots & \frac{\partial^2 f(\theta)}{\partial \theta_2 \partial \theta_n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(\theta)}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 f(\theta)}{\partial \theta_n \partial \theta_2} & \dots & \frac{\partial^2 f(\theta)}{\partial \theta_n^2} \end{bmatrix}$$

Newton's Algorithm

The most basic second-order optimization algorithm is Newton's algorithm, which consists of updates of the form,

$$\theta_{k+1} = \theta_k - \mathbb{H}_k^1 g_k$$

where g_k is the gradient at step k. This algorithm is derived by making a second-order Taylor series approximation of $f(\theta)$ around θ_k:

$$f_{quad}(\theta) = f(\theta_k) + g_k^T(\theta - \theta_k) + \frac{1}{2}(\theta - \theta_k)^T \mathbb{H}_k(\theta - \theta_k)$$

differentiating and equating to zero to solve for θ_{k+1}.

Learning Parameters

Now assume:

$$g(\theta) = \sum_{i=1}^{n} \Big[\sigma_\theta(x_i) - y_i\Big] x_i^j = \mathbf{X^T}(\sigma_\theta(\mathbf{X}) - \mathbf{y})$$

$$\pi_i = \sigma_\theta(x_i)$$

Let $\mathbb{H}$ represent the Hessian of $J(\theta)$

$$\begin{aligned}
\mathbb{H} = \frac{\partial}{\partial\theta} g(\theta) &= \frac{\partial}{\partial\theta} \sum_{i=1}^{n} \Big[\sigma_\theta(x_i) - y_i\Big] x_i^j \\
&= \sum_{i=1}^{n} \Big[\frac{\partial}{\partial\theta}\sigma_\theta(x_i) x_i^j - \frac{\partial}{\partial\theta} y_i x_i^j\Big] \\
&= \sum_{i=1}^{n} \sigma_\theta(x_i)(1 - \sigma_\theta(x_i)) x_i x_i^T \\
&= \mathbf{X^T} diag(\sigma_\theta(x_i)(1 - \sigma_\theta(x_i)))\mathbf{X}
\end{aligned}$$

Iteratively reweighted least squares (IRLS)

For binary logistic regression, recall that the gradient and Hessian of the negative log-likelihood are given by:

$$
\begin{aligned}
g(\theta)_k &= \mathbf{X}^{\mathbf{T}}(\pi_{\mathbf{k}} - \mathbf{y}) \\
\mathbf{H}_k &= \mathbf{X}^T S_k \mathbf{X} \\
\mathbf{S}_k &= diag(\pi_{1k}(1-\pi_{1k}), \ldots, \pi_{nk}(1-\pi_{nk})) \\
\pi_{ik} &= sigm(\mathbf{x_i}\theta_{\mathbf{k}})
\end{aligned}
$$

The Newton update at iteraion $k+1$ for this model is as follows:

$$
\begin{aligned}
\theta_{k+1} &= \theta_k - \mathbb{H}^{-1} g_k \\
&= \theta_k + (X^T S_k X)^{-1} X^T (y - \pi_k) \\
&= (X^T S_k X)^{-1} [(X^T S_k X)\theta_k + X^T (y - \pi_k)] \\
&= (X^T S_k X)^{-1} X^T [S_k X \theta_k + y - \pi_k]
\end{aligned}
$$

Regularized Logistic Regression

Unregularised:

$$J_1(\theta) = -\left\{ \sum_{i=1}^{n} y_i \log(\sigma_\theta(x_i)) + (1 - y_i) \log(1 - \sigma_\theta(x_i)) \right\}$$

L2 Regularization:

$$J(\theta) = J_1(\theta) + \lambda \theta^T \theta$$

L1 Regularization:

$$J(\theta) = J_1(\theta) + \lambda |\theta|$$