Logistic Regression

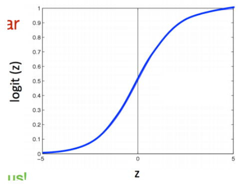
Type	Lecture Notes
Reviewed	☐
Available Summary?	In progress
Week	6

▼ Logistic Regression

- a technique for classification, Y is discrete
- Assumes the following form for $P(Y|X)$:
  - $P(Y=1|X) = \frac{1}{1+e^{w_0+\sum_i w_i X_i}}$
- logistic function applied to linear function of the data

  Logistic func. (sigmoid):

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



- features can be discrete or continuous

▼ Discriminative vs Generative Model

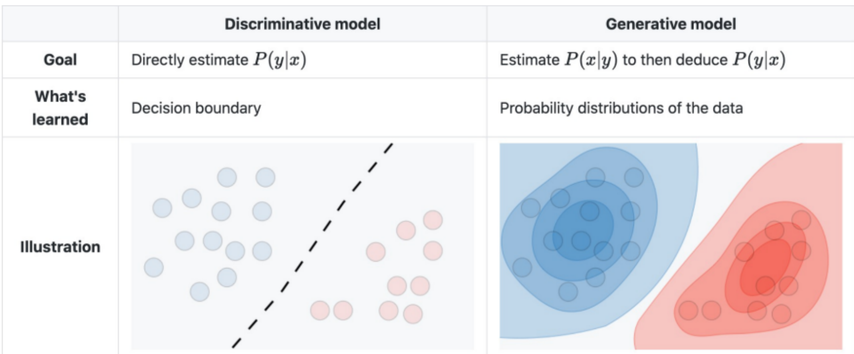
▼ Discriminative model:

- models decision boundary between the classes

▼ Generative model:

- explicitly models the actual feature distribution of each class

- at the end both of them is predicting the conditional probability $P(Y|X)$

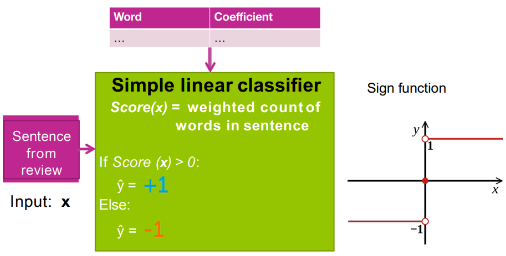


	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		

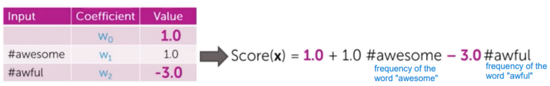
Logistic Regression

Naïve Bayes

▼ Linear Classifier



- if the output is weighted sum of input, then it is a linear classifier
- uses training data to learn a weight or coefficient for each word

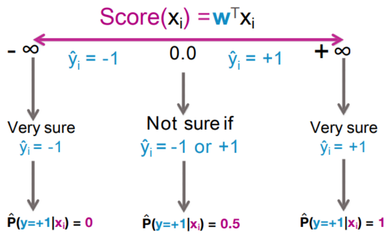


Input	Coefficient	Value
	w_0	1.0
#awesome	w_1	1.0
#awful	w_2	-3.0

▼ Decision boundary

- Separates positive and negative predictors
- For linear classifiers:
  - when 2 coefficients are non-zero line
  - when 3 coefficients are non-zero plane
  - when many coefficients are non-zero hyperplane
- For more general classifiers more complicated shapes

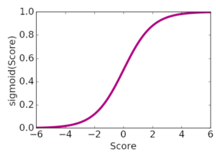
▼ Interpreting Score



▼ Sigmoid

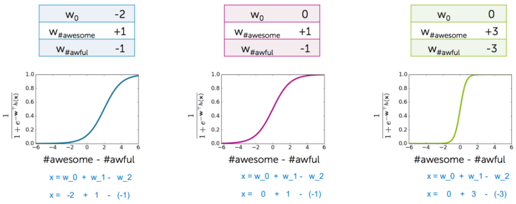
$$sigmoid(\text{Score}) = \frac{1}{1 + e^{-\text{Score}}}$$

Score	$-\infty$	-2	0.0	+2	$+\infty$
sigmoid(Score)	0	0.12	0.5	0.88	1



- sigmoid(score) is always bounded between [0,1]
- as score increases g(z) approaches to 1
- as score decreases g(z) approaches to 0
- it is differentiable

▼ Effect of coefficient



▼ *an example*

$$w_0^{\hat{MLE}} = -10.65$$
$$\hat{w}_1^{MLE} = 0.0055$$

What is the probability that an individual with a balance of $1000 defaults?

$$P(\text{default} = 1|\text{balance} = 1000) = \frac{e^{-10.65+0.0055\times1000}}{1 + e^{-10.65+0.0055\times1000}} \approx 0.0058 = 0.58\%$$

▼ Finding best coefficients

- Likelihood $l(w)$: measures quality of fit for model with coefficients w
- To find the best classifier → maximize likelihood over all possible w values

$$max_{all\_possible\_w} \prod_{i=1}^{N} P(y_i|x_i, w)$$
$$= ln \prod_{i=1}^{N} P(y_i|x_i, w)$$
$$= \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - ln(1 + e^{(w_0 + \sum_i^d w_i x_i^j)}) \right]$$

▼ Good news:

- $l(w)$ is concave function of w →no locally solutions
- concave functions are easy to optimize (unique maximum)

▼ Bad news:

- no closed-form solution to maximize $l(w)$

▼ Convex vs Concave

Maximum of a concave function = minimum of a convex function

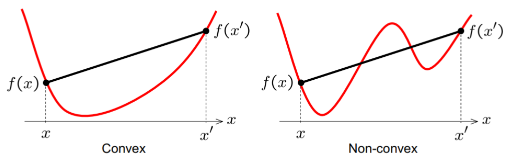
▼ Convex vs Non-convex

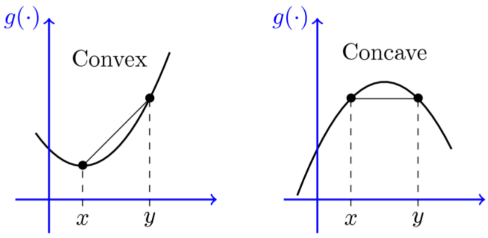
A function $f : A \subseteq \mathbb{X} \to \mathbb{R}$ defined on a convex set A is called convex if

$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

for any $x, x' \in \mathbb{X}$ and $\lambda \in [0, 1]$

For convex function local minimum = global minimum





▼ Training Logistic Regression

- Maximum (Conditional) Likelihood Estimates:

  $\hat{w}_{MLE} = argmax_w \prod_{j=1}^{n} P(X^{(j)}, Y^{(j)}|w)$

> 💡 ***Discriminative philosophy:***
>
> Don't waste effort learning $P(X)$, focus on $P(Y|X)$ - that's all that matters for classification!

▼ Gradients

- The gradient of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

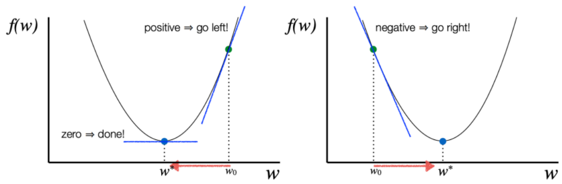
> 💡 ***Gradient Descent:***
> Find the gradient of the function at the current point and move in the opposite direction.

▼ Steps:

1. Set a random point
2. Determine a descent direction
   a. Negative slope is direction of descent!!
3. Choose a step size
4. Apply update rule
   a. Update rule:$w_{t+1} \leftarrow w_t - \eta \nabla E_d(w_t)$

      η: step size

Repeat the steps until stopping criterion is satisfied.



▼ Effect of step size:

Large η → fast convergence but larger residual error, also possible oscillations

Small η → Slow convergence but small residual error

Derivation on the board.

Gradient ascent algorithm: iterate until change < ε

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For i=1,…,d,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \underbrace{\hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})}_{\text{Predict what current weight thinks label Y should be}}]$$

repeat

> 💡 ***Derivative of the sigmoid:***
>
> $\sigma(x).(1 - \sigma(x))$

> 💡 Maximizing log-likelihood is equivalent to minimizing -log-likelihood and equivalent to minimizing cross-entrpy loss

▼ Binary Cross-Entropy Loss (Log-loss)

$$\overbrace{-\sum_{i=1}^{N}}^{\text{Sum over examples}} \overbrace{\underbrace{y_i}_{\text{Label}} log(\underbrace{p(y_i)}_{\text{Prob of positive class}}) + (1 - \underbrace{y_i}_{\text{Label}}) log(1 - \underbrace{p(y_i)}_{\text{Prob of positive class}})}^{\text{Sum over classes}}$$

left at pg .72