

# Logistic Regression

HW 2 DUE  
**Tomorrow**

# Naïve Bayes Recap

- Bag of words (order independent)
- Features are assumed independent given class

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \dots P(x_n | c)$$

**Q: Is this really true?**

# The problem with assuming conditional independence

- Correlated features -> double counting evidence
  - Parameters are estimated independently
- This can hurt classifier accuracy and calibration

# Logistic Regression

- (Log) Linear Model – similar to Naïve Bayes
- Doesn't assume features are independent
- Correlated features don't “double count”

# What are “Features”?

- A feature function,  $f$ 
  - Input: Document,  $D$  (a string)
  - Output: Feature Vector,  $X$

# What are “Features”?

$$f(d) = \begin{pmatrix} \text{count}(\text{“boring”}) \\ \text{count}(\text{“not boring”}) \\ \text{length of document} \\ \text{author of document} \\ \vdots \end{pmatrix}$$

Doesn't have to be just “bag of words”

# Feature Templates

- Typically “feature templates” are used to generate many features at once
- For each word:
  - $\text{\$}\{w\}_\text{count}$
  - $\text{\$}\{w\}_\text{lowercase}$
  - $\text{\$}\{w\}_\text{with\_NOT\_before\_count}$



# Logistic Regression: Example

- Compute Features:

$$f(d_i) = x_i = \begin{pmatrix} \text{count}(\text{"won"}) \\ \text{count}(\text{"choose"}) \\ \text{count}(\text{"\$1,00,00,00,000"}) \end{pmatrix}$$

- Assume we are given some weights:

$$w = \begin{pmatrix} -1.0 \\ -1.0 \\ 4.0 \end{pmatrix}$$

# Logistic Regression: Example

- Compute Features
- We are given some weights
- Compute the dot product:

$$z = \sum_{i=0}^{|X|} w_i x_i$$

# Logistic Regression: Example

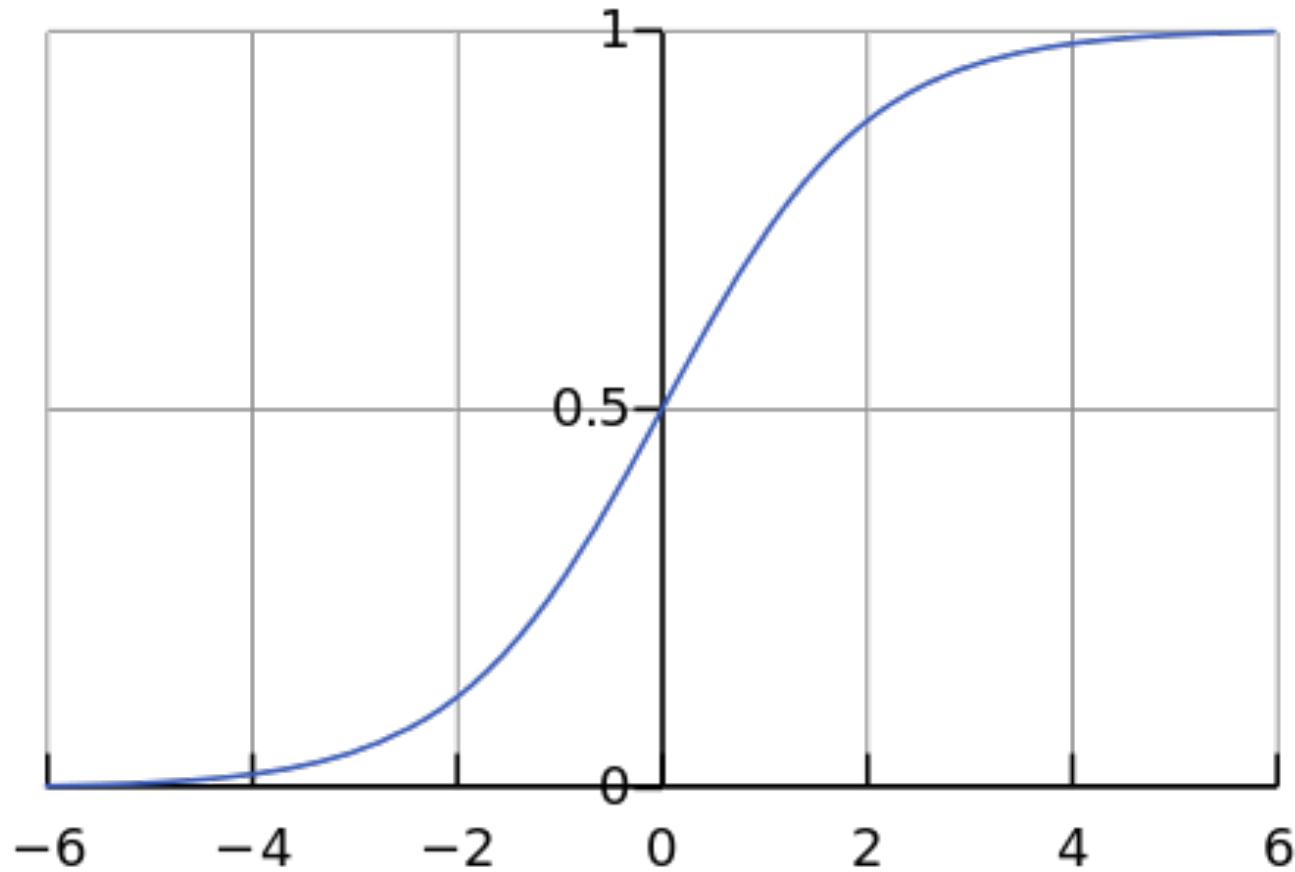
- Compute the dot product:

$$z = \sum_{i=0}^{|X|} w_i x_i$$

- Compute the logistic function:

$$P(\text{spam}|x) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

# The Logistic function

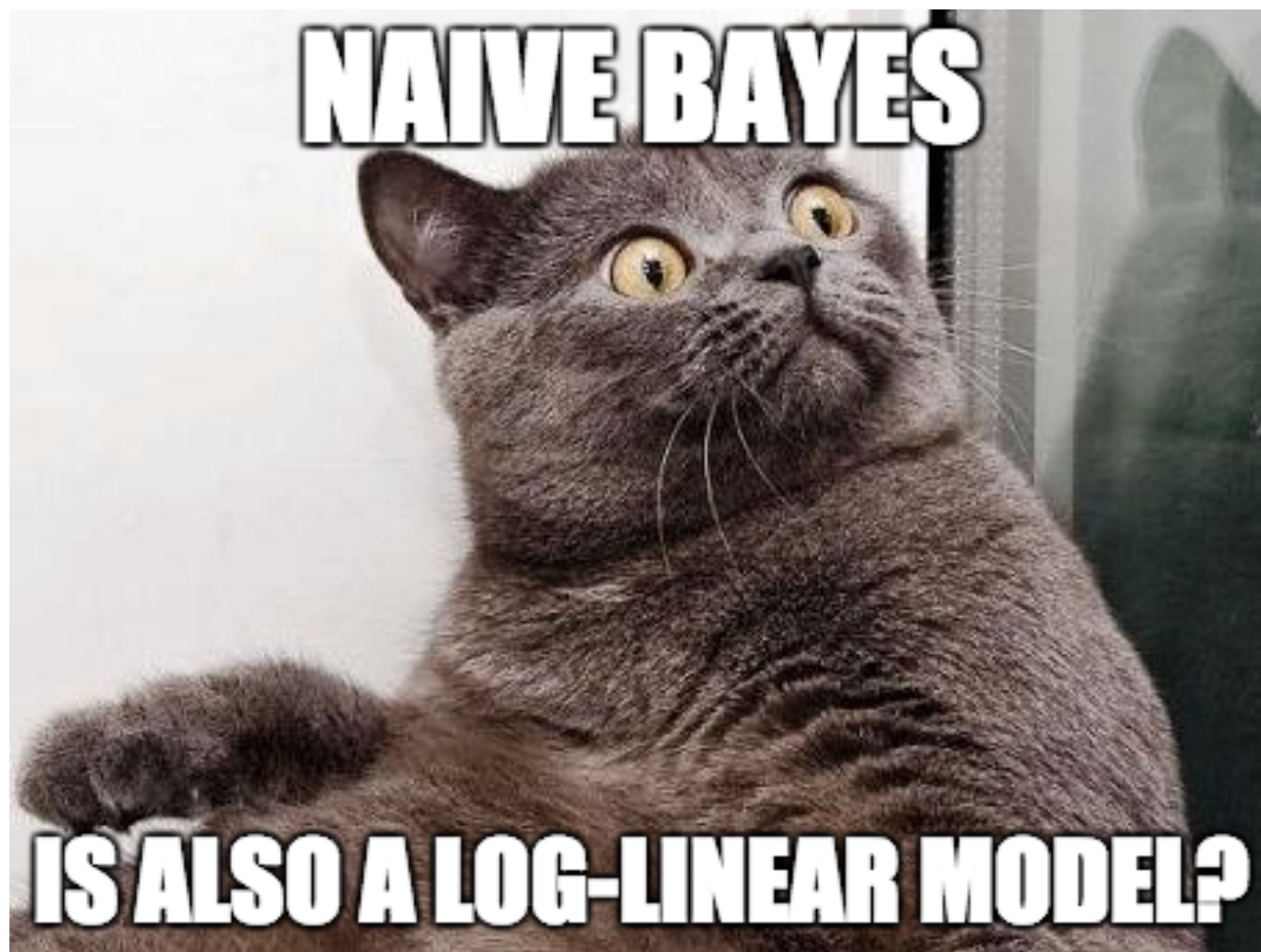


$$P(\text{spam}|x) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

# The Dot Product

$$z = \sum_{i=0}^{|X|} w_i x_i$$

- Intuition: weighted sum of features
- All Linear models have this form



# Naïve Bayes as a log-linear model

- Q: what are the features?
- Q: what are the weights?

# Naïve Bayes as a Log-Linear Model

$$P(\text{spam}|D) \propto P(\text{spam}) \prod_{w \in D} P(w|\text{spam})$$

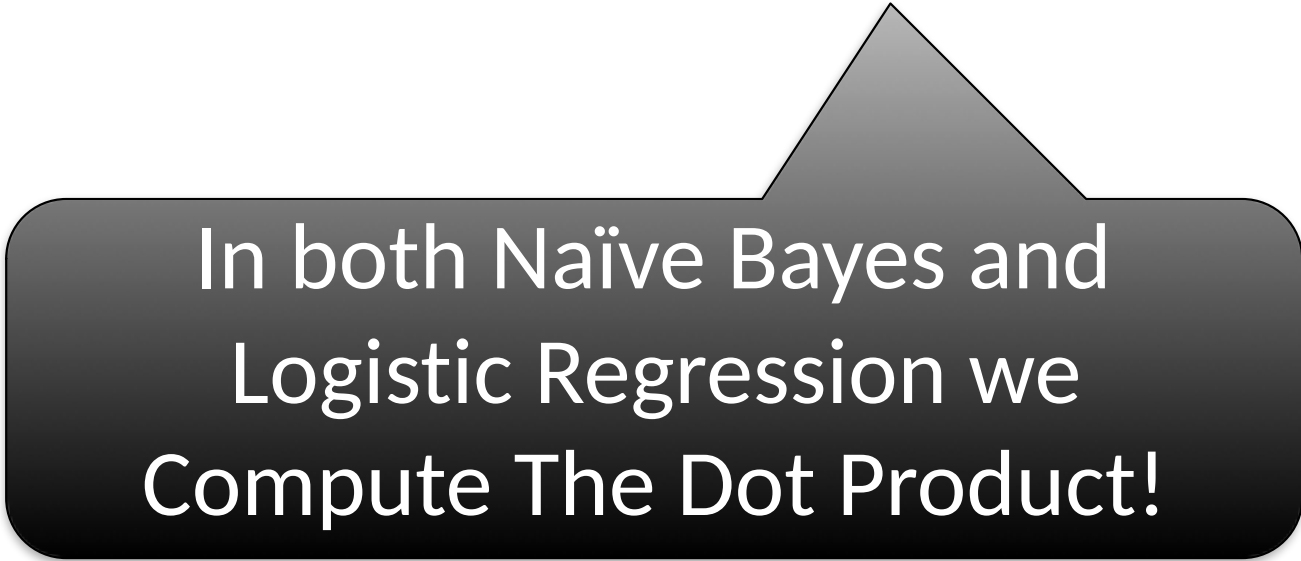
$$P(\text{spam}|D) \propto P(\text{spam}) \prod_{w \in \text{Vocab}} P(w|\text{spam})^{x_i}$$

$$\log P(\text{spam}|D) \propto \log P(\text{spam}) + \sum_{w \in \text{Vocab}} x_i \cdot \log P(w|\text{spam})$$



# Naïve Bayes as a Log-Linear Model

$$\log P(\text{spam}|D) \propto \log P(\text{spam}) + \sum_{w \in \text{Vocab}} x_i \cdot \log P(w|\text{spam})$$



In both Naïve Bayes and  
Logistic Regression we  
Compute The Dot Product!

# NB vs. LR

- Both compute the dot product
- NB: sum of log probabilities
- LR: logistic function

# NB vs. LR:

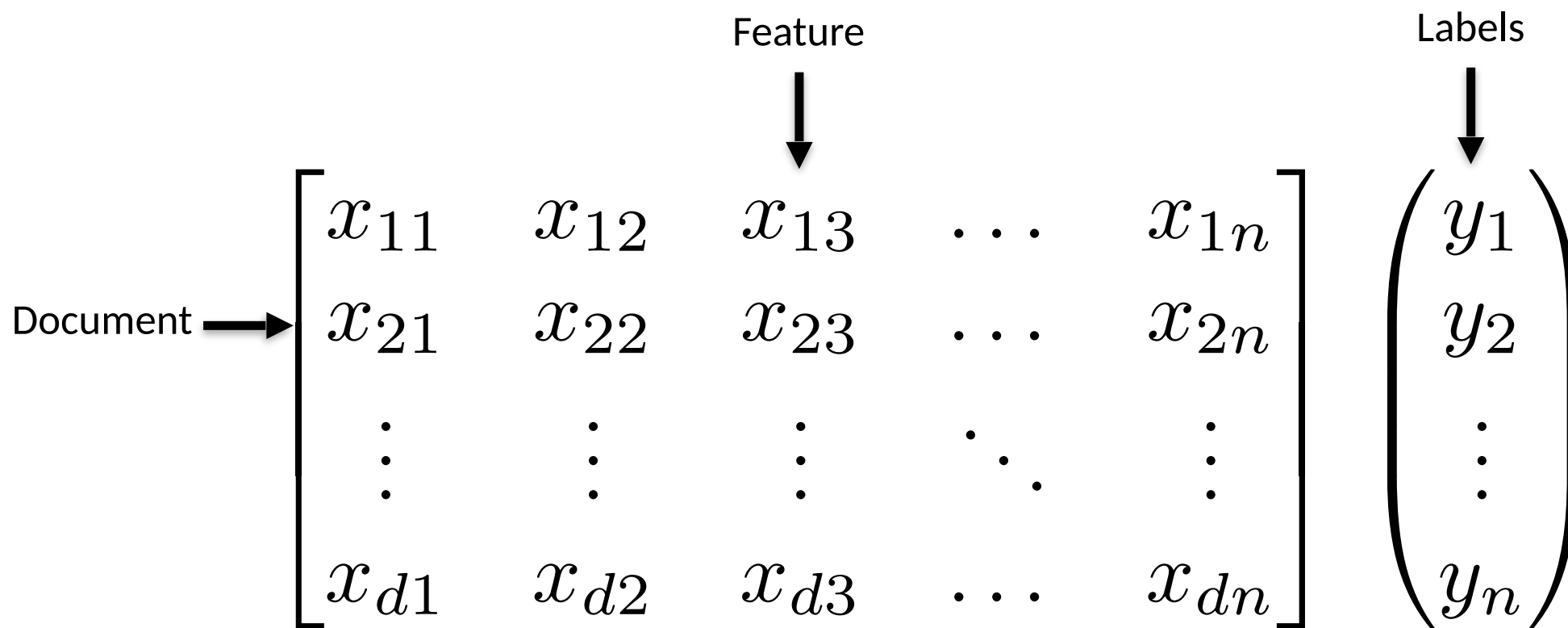
## Parameter Learning

- Naïve Bayes:
  - Learn conditional probabilities **independently** by counting
- Logistic Regression:
  - Learn weights **jointly**

# LR: Learning Weights

- Given: a set of feature vectors and labels
- Goal: learn the weights

# Learning Weights



# Q: what parameters should we choose?

- What is the right value for the weights?
- Maximum Likelihood Principle:
  - Pick the parameters that maximize the probability of the data

# Maximum Likelihood Estimation

$$\begin{aligned}w_{\text{MLE}} &= \operatorname{argmax}_w \log P(y_1, \dots, y_d | x_1, \dots, x_d; w) \\&= \operatorname{argmax}_w \sum_i \log P(y_i | x_i; w) \\&= \operatorname{argmax}_w \sum_i \log \begin{cases} p_i, & \text{if } y_i = 1 \\ 1 - p_i, & \text{if } y_i = 0 \end{cases} \\&= \operatorname{argmax}_w \sum_i \log p_i^{\mathbb{I}(y_i=1)} (1 - p_i)^{\mathbb{I}(y_i=0)}\end{aligned}$$

# Maximum Likelihood Estimation

$$= \operatorname{argmax}_w \sum_i \log p_i^{\mathbb{I}(y_i=1)} (1 - p_i)^{\mathbb{I}(y_i=0)}$$

$$= \operatorname{argmax}_w \sum_i y_i \log p_i + (1 - y_i) \log(1 - p_i)$$



# Maximum Likelihood Estimation

- Unfortunately there is no closed form solution
  - (like there was with naïve bayes)
- Solution:
  - Iteratively climb the log-likelihood surface through the derivatives for each weight
- Luckily, the derivatives turn out to be nice

# Logistic Regression: Pros and Cons

- Doesn't assume conditional independence of features
  - Better calibrated probabilities
  - Can handle highly correlated overlapping features
- NB is faster to train, less likely to overfit

# NB & LR

- Both are linear models

$$z = \sum_{i=0}^{|X|} w_i x_i$$

- Training is different:
  - NB: weights are trained independently
  - LR: weights trained jointly