Logistic Regression

EECS 349

Discriminative vs. Generative training

- Say our distribution has variables ***X*** , ***Y***
- Naïve Bayes learning learns P(***X*** , ***Y***)
- But often, the only inferences we care about are of form P(***Y*** | ***X***)
  - P(*Disease* | *Symptoms* = **e**)
  - P(*StockMarketCrash* | *RecentPriceActivity* = **e**)

Discriminative vs. Generative training

- Learning P(***X*** , ***Y***): **generative** training
  - Learned model can "generate" the full data ***X, Y***
- Learning only P(***Y*** | ***X***): **discriminative** training
  - Model can't assign probs. to ***X***. Only ***Y*** given ***X***
- Idea: Only model what we care about
  - Don't "waste data" on params irrelevant to task
  - Side-step false independence assumptions in training (example to follow)

Generative Model Example

- Naïve Bayes model
  - Y binary {1=spam, 0=not spam}
    $\boldsymbol{X}$ an n-vector: message has word (1) or not (0)
  - Re-write P(Y | $\boldsymbol{X}$) using Bayes Rule, apply Naïve Bayes assumption
  - $2n + 1$ parameters, for n observed variables

Generative => Discriminative (1 of 3)

- But $P(Y \mid \boldsymbol{X})$ can be written more compactly

$$P(Y \mid \boldsymbol{X}) = \frac{1}{1 + \exp(w_0 + w_1 x_1 + \ldots + w_n x_n)}$$

- Total of n + 1 parameters w_i

Generative => Discriminative (2 of 3)

- One way to do conversion (vars binary):

$$\exp(w_0) = \frac{P(Y=0)\, P(X_1=0|Y=0)\, P(X_2=0|Y=0)\ldots}{P(Y=1)\, P(X_1=0|Y=1)\, P(X_2=0|Y=1)\ldots}$$

for $i > 0$:

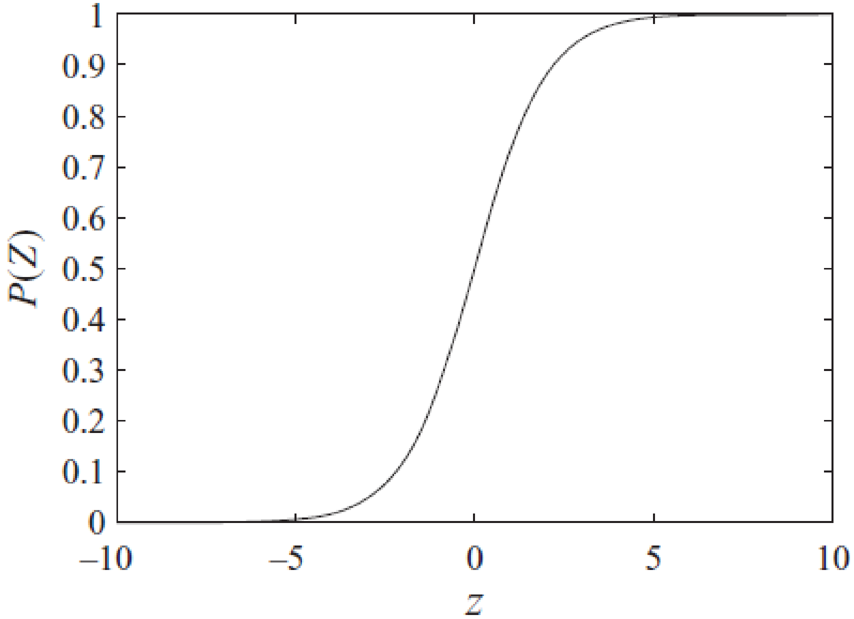
$$\exp(w_i) = \frac{P(X_i=0|Y=1)\, P(X_i=1|Y=0)}{P(X_i=0|Y=0)\, P(X_i=1|Y=1)}$$

Generative => Discriminative (3 of 3)

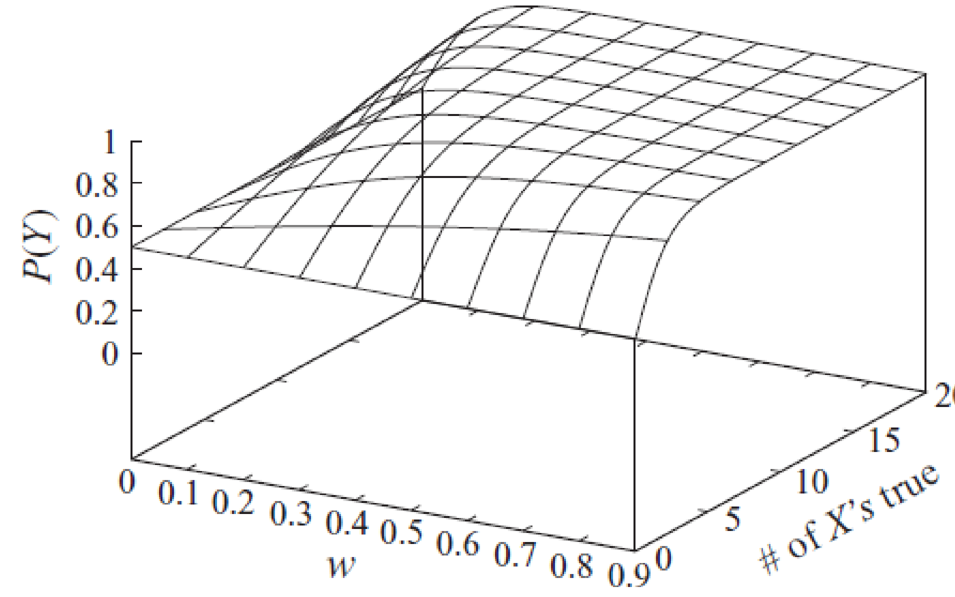
- We reduced $2n + 1$ parameters to $n + 1$
  - This must be *better*, right?
- Not exactly. If we construct $P(Y \mid \boldsymbol{X})$ to be equivalent to Naïve Bayes (as on prev. slide)
  - then it's…equivalent to Naïve Bayes
- Idea: optimize the $n + 1$ parameters directly, using training data

Discriminative Training

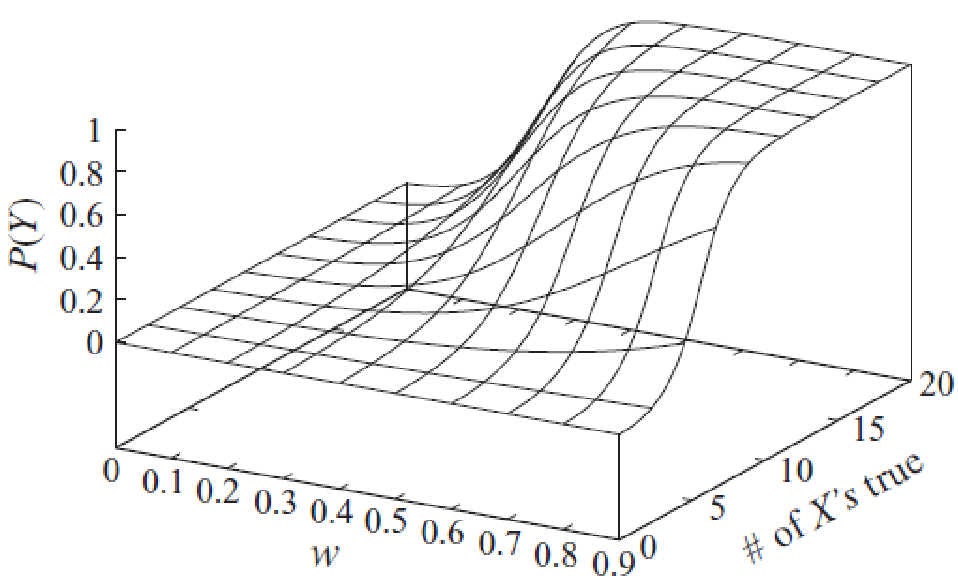
- In our example:
  $$P(Y \mid \boldsymbol{X}) = \frac{1}{1 + \exp(w_0 + w_1 x_1 + \ldots + w_n x_n)}$$
- Goal: find w_i that maximize likelihood of training data Ys given training data $\boldsymbol{X}$s
  - Known as "logistic regression"
  - Solved with gradient ascent techniques
  - A convex optimization problem

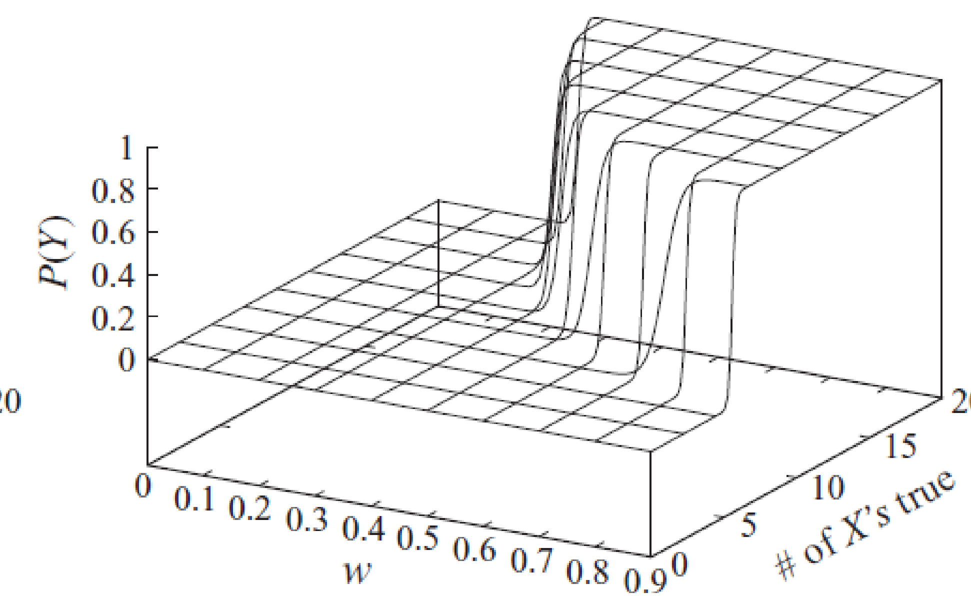

P(Z)
z
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
−10
−5
0
5
10

(a)


P(Y)
w
# of X's true
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20

(b)


P(Y)
w
# of X's true
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20


P(Y)
w
# of X's true
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20

Naïve Bayes vs. LR

- Both models operate over the same hypothesis space

- So what's the difference? Training method.
  - Naïve Bayes "trusts its assumptions" in training
  - Logistic Regression doesn't – recovers better when assumptions violated

NB vs. LR: Example

Training Data

SPAM	Lottery	Winner	Lunch	Noon
1	1	1	0	0
1	1	1	1	1
0	0	0	1	1
0	1	1	0	1

- Naïve Bayes will classify the last example incorrectly, even after training on it!
- Whereas Logistic Regression is perfect with e.g.,
  $w_0 = 0.1 \quad w_{lottery} = w_{winner} = w_{lunch} = -0.2 \quad w_{noon} = 0.4$

Logistic Regression in practice

- Can be employed for any numeric variables X_i
  - or for other variable types, by converting to numeric (e.g. indicator) functions
- "Regularization" plays the role of priors in Naïve Bayes
- Optimization tractable, but (way) more expensive than counting (as in Naïve Bayes)

Discriminative Training

- Naïve Bayes vs. Logistic Regression one illustrative case
- Applicable more broadly, whenever queries P(***Y*** | ***X***) known *a priori*