Naïve Bayes Classifiers and Logistic Regression

Doug Downey
Northwestern EECS 349
Winter 2014

Naïve Bayes Classifiers

- Combines all ideas we've covered
  - Conditional Independence
  - Bayes' Rule
  - Statistical Estimation
  - Bayes Nets
- …in a simple, yet accurate classifier
  - Classifier: Function f($\boldsymbol{x}$) from $\boldsymbol{X}$ = {<x_1, …, x_d>} to *Class*
  - E.g., $\boldsymbol{X}$ = {<GRE, GPA, Letters>}, *Class* = {yes, no, wait}

Probability => Classification (1 of 2)

- Classification Task:
  - Learn function f($\boldsymbol{x}$) from $\boldsymbol{X} = \{<x_1, \ldots, x_d>\}$ to *Class*
  - Given: Examples $D=\{(\boldsymbol{x}, y)\}$

- Probabilistic Approach
  - Learn P(*Class* = y | $\boldsymbol{X} = \boldsymbol{x}$) from D
  - Given $\boldsymbol{x}$, pick the maximally probable y

Probability => Classification (2 of 2)

- More formally
  - $f(\boldsymbol{x}) = \arg\max_y P(\mathit{Class} = y \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{MAP}})$
  - $\boldsymbol{\theta}_{\mathrm{MAP}}$ *:* MAP parameters, learned from data
    - That is, parameters of $P(\mathit{Class} = y \mid \boldsymbol{X} = \boldsymbol{x})$
  - …we'll focus on using MAP estimate, but can also use ML or Bayesian
- Predict next coin flip? Instance of this problem
  - X = null
  - Given *D*= hhht…tht, estimate $P(\theta \mid D)$, find MAP
  - Predict *Class* = heads iff $\theta_{\mathrm{MAP}} > ½$

Example: Text Classification

Dear Sir/Madam,
We are pleased to inform you of the result of the Lottery Winners International programs held on the 30/8/2004. Your e-mail address attached to ticket number: EL-23133 with serial Number: EL-123542, batch number: 8/163/EL-35, lottery Ref number: EL-9318 and drew lucky numbers 7-1-8-36-4-22 which consequently won in the 1st category, you have therefore been approved for a lump sum pay out of US$1,500,000.00 (One Million, Five Hundred Thousand United States dollars)

- SPAM

NOT SPAM?

Representation

- ***X*** = document
- Estimate P(*Class* = {spam, non-spam} | ***X***)
- Question: how to represent ***X***?
  - One dimension for each possible e-mail, i.e. possible permutation of words?
    - No.
  - Lots of possibilities, common choice: “bag of words”

Dear Sir/Madam,
We are pleased to inform you of the result of the Lottery Winners International programs held on the 30/8/2004. Your e-mail address attached to ticket number: EL-23133 with serial Number: EL-123542, batch number: 8/163/EL-35, lottery Ref number: EL-9318 and drew lucky numbers 7-1-8-36-4-22 which consequently won in the 1st category, you have therefore been approved for a lump sum pay out of US$1,500,000.00 (One Million, Five Hundred Thousand United States dollars)
…

→

Sir	1
Lottery	10
Dollars	7
With	38
…	

Bag of Words

- Ignores Word Order, i.e.
  - No emphasis on title
  - No compositional meaning (“Cold War” -> “cold” and “war”)
  - Etc.
  - But, massively reduces dimensionality/complexity
- Still and all…
  - Recording presence or absence of a 100,000-word vocab entails 2^100,000 distinct vectors

Naïve Bayes Classifiers

- $P(Class \mid \boldsymbol{X})$ for $|\mathrm{Val}(\boldsymbol{X})| = 2$^100,000 requires 2^100,000 parameters
  - Problematic.
- Bayes' Rule:
  $P(Class \mid \boldsymbol{X}) = \mathrm{P}(\boldsymbol{X} \mid Class)\, \mathrm{P}(Class) \,/\, \mathrm{P}(\boldsymbol{X})$
- Assume presence of word *i* is independent of all other words given *Class*:
  $P(Class \mid \boldsymbol{X}) = \Pi_i \, \mathrm{P}(w_i \mid Class)\, \mathrm{P}(Class) \,/\, \mathrm{P}(\boldsymbol{X})$
- Now only 200,001 parameters for $P(Class \mid \boldsymbol{X})$

Naïve Bayes Assumption

- Features are conditionally independent given class
  - *Not* $P(\text{"Republican"}, \text{"Democrat"}) = P(\text{"Republican"})P(\text{"Democrat"})$ but instead
    $P(\text{"Republican"}, \text{"Democrat"} \mid Class = \text{Politics}) =$
    $P(\text{"Republican"} \mid Class = \text{Politics})P(\text{"Democrat"} \mid Class = \text{Politics})$
- Still, an absurd assumption
  - ("Lottery" $\perp$ "Winner" | SPAM)? ("lunch" $\perp$ "noon" | Not SPAM)?
- But: offers massive tractability advantages and works quite well in practice
  - Lesson: Overly strong independence assumptions sometimes allow you to build an accurate model where you otherwise couldn't

Getting the parameters from data

- Parameters $\boldsymbol{\theta}$ = < θ_{ij} =P(w_i | *Class = j*) >
- Maximum Likelihood: Estimate P(w_i | *Class = j*) from *D* by counting
  - Fraction of documents in class *j* containing word *i*
  - But if word *i* never occurs in class *j* ?
- Commonly used MAP estimate:
  - $\dfrac{(\#\text{ docs in class } j \text{ with word } i) + 1}{(\#\text{ docs in class } j) + |V|}$

Caveats

- Naïve Bayes effective as a *classifier*

- **Not** as effective in producing probability estimates
  - Π_i P(w_i | *Class*) pushes estimates toward 0 or 1

- In practice, numerical underflow is typical at classification time
  - Compare sum of logs instead of product

Discriminative vs. Generative training

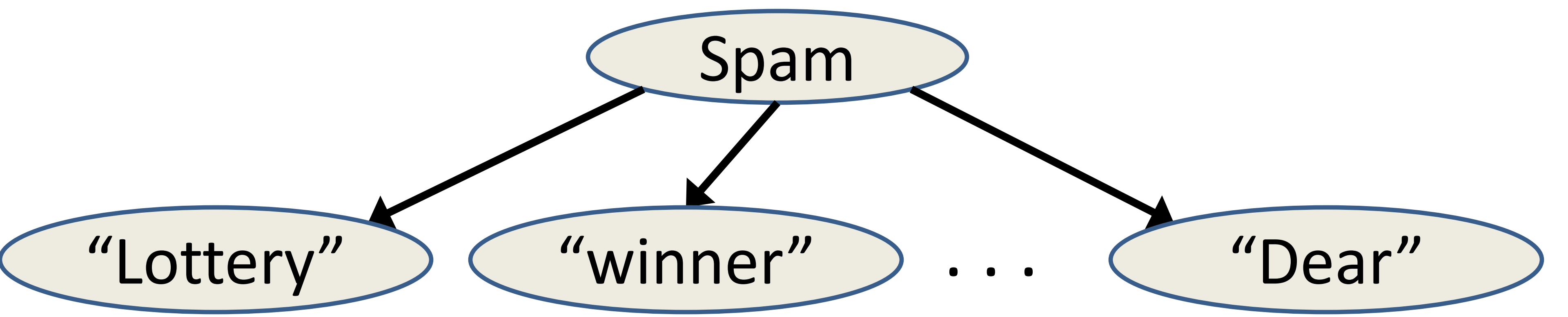
- Say our graph *G* has variables ***X*** , ***Y***
- Previous method learns P(***X*** , ***Y***)
- But often, the only inferences we care about are of form P(***Y*** | ***X***)
  - P(*Disease* | *Symptoms* = ***e***)
  - P(*StockMarketCrash* | *RecentPriceActivity* = ***e***)

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}
    X an *n*-vector: message has word (1) or not (0)
  - Re-write P(Y | ***X***) using Bayes Rule, apply Naïve Bayes assumption
  - 2*n* + 1 parameters, for *n* observed variables

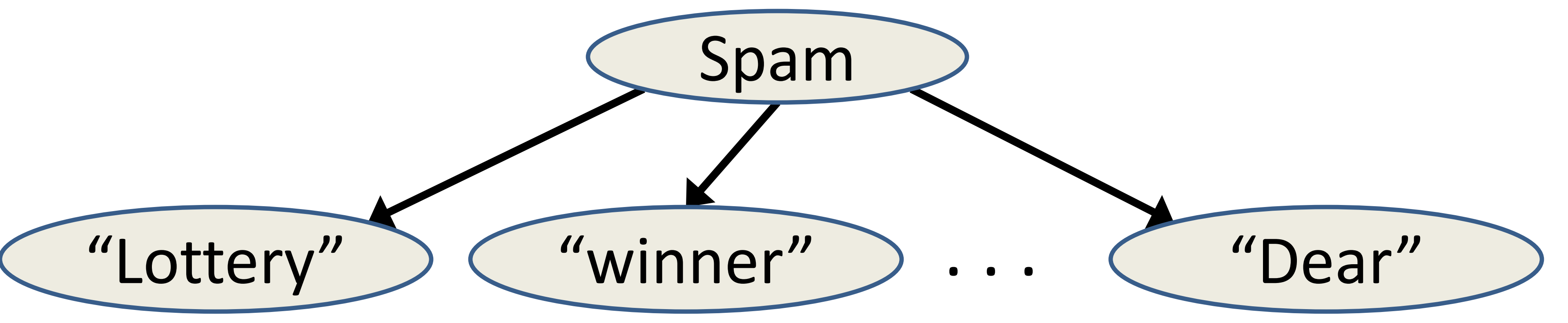


Generative => Discriminative (1 of 3)

- But P(Y | $\boldsymbol{X}$) 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):

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

for $i > 0$:

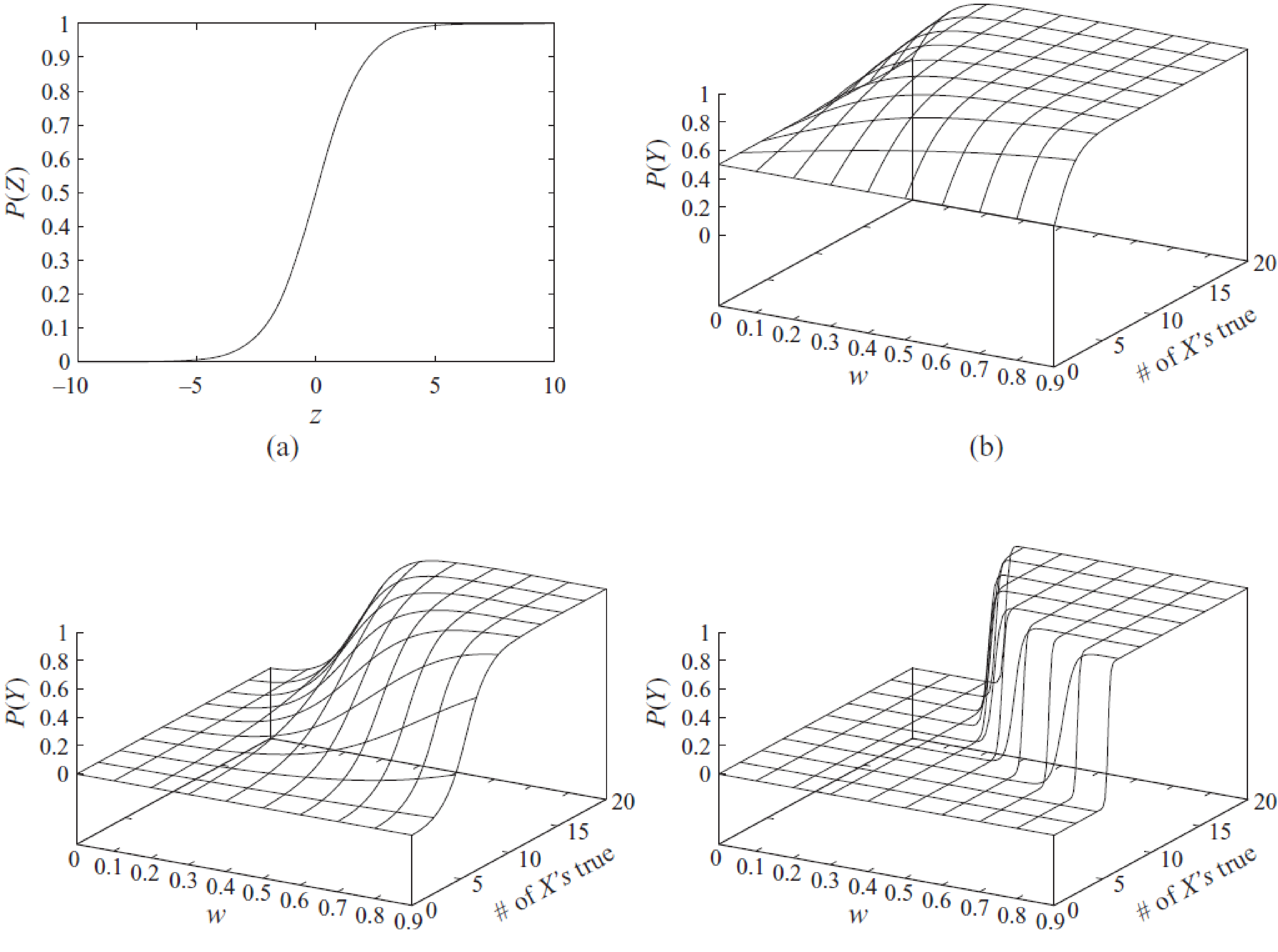
$$\exp(w_i) = \frac{P(X_i{=}0 \mid Y{=}1)\ P(X_i{=}1 \mid Y{=}0)}{P(X_i{=}0 \mid Y{=}0)\ P(X_i{=}1 \mid Y{=}1)}$$

Generative => Discriminative (3 of 3)

- We reduced $2n + 1$ parameters to $n + 1$
  - Bias vs. Variance arguments says this must be better, right?
- Not exactly. If we construct P(Y | $\boldsymbol{X}$) to be equivalent to Naïve Bayes (as before)
  - 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 *Y*s given training data ***X***s
  - Known as "logistic regression"
  - Solved with gradient ascent techniques
  - A convex (actually concave) optimization problem



P(Z)
z
−10
−5
0
5
10
(a)
P(Y)
w
# of X's true
(b)

Naïve Bayes vs. LR

- 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.,
  $w_0 = 0.1$
  $w_{lottery} = w_{winner} = w_{lunch} = -0.2$
  $w_{noon} = 0.4$

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*

Data Set	MNB-FM	SFE	MNB	NBEM	LProp	Logist
Apte (10)	0.306	0.271	**0.336**	0.306	0.245	0.208
Apte (100)	**0.554**	0.389	0.222	0.203	0.263	0.330
Apte (1k)	**0.729**	0.614	0.452	0.321	0.267	0.702
Amzn (10)	**0.542**	0.524	0.508	0.475	0.470*	0.499
Amzn (100)	**0.587**	0.559	0.456	0.456	0.498*	0.542
Amzn (1k)	0.687	0.611	0.465	0.455	0.539*	**0.713**
RCV1 (10)	**0.494**	0.477	0.387	0.485	-	0.272
RCV1 (100)	**0.677**	0.613	0.337	0.470	-	0.518
RCV1 (1k)	0.772	0.735	0.408	0.491	-	**0.774**

* Limited to 5 of 10 Amazon categories