Generative vs. Discriminative models

Introduction

- So far we've looked at "generative models"
  - Language models, Naive Bayes
- But there is now much use of conditional or discriminative probabilistic models in NLP

Conditional vs. Joint Likelihood

- A *joint* model gives probabilities P(*d,c*) and tries to maximize this joint likelihood.

$$= \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- A *conditional* model directly comptes P(*c*|*d*).

Features

- In these slides and most maxent work: *features* f are elementary pieces of evidence that link aspects of what we observe d with a category c that we want to predict
- A feature is a function with a bounded real value

Example features

- $f_1(c, d) \equiv [c = \text{LOCATION} \wedge w_{-1} = \text{“in”} \wedge \text{isCapitalized}(w)]$
- $f_2(c, d) \equiv [c = \text{LOCATION} \wedge \text{hasAccentedLatinChar}(w)]$
- $f_3(c, d) \equiv [c = \text{DRUG} \wedge \text{ends}(w, \text{“c”})]$

LOCATION	LOCATION	DRUG	PERSON
in Arcadia	*in Québec*	*taking Zantac*	*saw Sue*

- Models will assign to each feature a *weight:*
  - A positive weight votes that this configuration is likely correct
  - A negative weight votes that this configuration is likely incorrect

Example features

- $f_1(c, d) \equiv [c = \text{LOCATION} \wedge w_{-1} = \text{“in”} \wedge \text{isCapitalized}(w)]$
- $f_2(c, d) \equiv [c = \text{LOCATION} \wedge \text{hasAccentedLatinChar}(w)]$
- $f_3(c, d) \equiv [c = \text{DRUG} \wedge \text{ends}(w, \text{“c”})]$

LOCATION	LOCATION	DRUG	PERSON
in Arcadia	*in Québec*	*taking Zantac*	*saw Sue*

- Models will assign to each feature a *weight:*
  - A positive weight votes that this configuration is likely correct
  - A negative weight votes that this configuration is likely incorrect

Features

- In NLP uses, usually a feature specifies (1) an indicator function – a yes/no boolean matching function – of properties of the input and (2) a particular class
  - $f_i(c, d) \equiv [\Phi(d) \wedge c = c_j]$ [Value is 0 or 1]
  - They pick out a data subset and suggest a label for it.
- We will say that $\Phi(d)$ is a feature of the data d, when, for each c_j, the conjunction $\Phi(d) \wedge c = c_j$ is a feature of the data-class pair (c, d)

Features

- In NLP uses, usually a feature specifies
  1. an indicator function – a yes/no boolean matching function – of properties of the input and
  2. a particular class

$$f_i(c, d) \equiv [\Phi(d) \wedge c = c_j]$$

[Value is 0 or 1]

- Each feature picks out a data subset and suggests a label for it

Feature-Based Models

- The decision about a data point is based only on the features active at that point.

Text Categorization	Word-Sense Disambiguation	POS Tagging
Data BUSINESS: Stocks hit a yearly low …	Data … to restructure bank:MONEY debt.	Data DT JJ NN … The previous fall …
Label: BUSINESS Features {…, stocks, hit, a, yearly, low, …}	Label: MONEY Features {…, w_{-1}=restructure, w_{+1}=debt, L=12, …}	Label: NN Features {w=fall, t_{-1}=JJ w_{-1}=previous}

Feature-Based Linear Classifiers

- Linear classifiers at classification time:
  - Linear function from feature sets $\{f_i\}$ to classes $\{c\}$.
  - Assign a weight λ_i to each feature f_i.
  - We consider each class for an observed datum d
  - For a pair (c,d), features vote with their weights:
    - vote(c) = $\Sigma\lambda_i f_i(c,d)$

PERSON *in Québec* | LOCATION *in Québec* | DRUG *in Québec*

  - Choose the class c which maximizes $\Sigma\lambda_i f_i(c,d)$

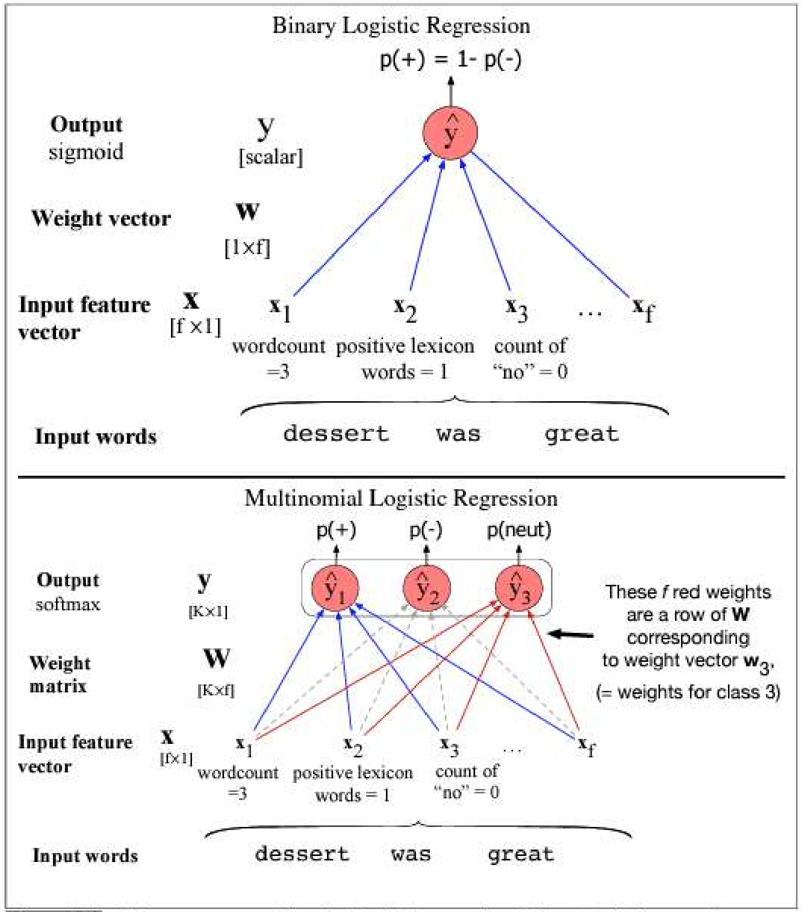

Figure 5.3 Binary versus multinomial logistic regression. Binary logistic regression uses a single weight vector **w**, and has a scalar output $\hat{y}$. In multinomial logistic regression we have K separate weight vectors corresponding to the K classes, all packed into a single weight matrix **W**, and a vector output $\hat{\mathbf{y}}$.

Feature-Based Linear Classifiers

- Linear classifiers at classification time:
  - Linear function from feature sets $\{f_i\}$ to classes $\{c\}$.
  - Assign a weight λ_i to each feature f_i.
  - We consider each class for an observed datum d
  - For a pair (c,d), features vote with their weights:
    - $\text{vote(c)} = \Sigma\lambda_i f_i(c,d)$

PERSON *in Québec*

1.8 LOCATION *in Québec* –0.6

0.3 DRUG *in Québec*

  - Choose the class c which maximizes $\Sigma\lambda_i f_i(c,d)$ = LOCATION

Multinomial Linear Classifiers

- Exponential (log-linear, maxent, logistic, Gibbs) models:
  - Make a probabilistic model from the linear combination $\Sigma\lambda_i f_i(c,d)$

$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c',d)}$$

Makes votes positive

Normalizes votes

- P(LOCATION|*in Québec*) = $e^{1.8}e^{-0.6}/(e^{1.8}e^{-0.6} + e^{0.3} + e^{0})$ = 0.586
- P(DRUG|*in Québec*) = $e^{0.3}/(e^{1.8}e^{-0.6} + e^{0.3} + e^{0})$ = 0.238
- P(PERSON|*in Québec*) = $e^{0}/(e^{1.8}e^{-0.6} + e^{0.3} + e^{0})$ = 0.176

  - The weights are the parameters of the probability model, combined via a "soft max" function

Feature-Based Linear Classifiers

- Exponential (log-linear, maxent, logistic, Gibbs) models:
  - Given this model form, we will choose parameters $\{\lambda_i\}$ that *maximize the conditional likelihood* of the data according to this model.
  - We construct not only classifications, but probability distributions over classifications.
    - There are other (good!) ways of discriminating classes – SVMs, boosting, even perceptrons – but these methods are not as trivial to interpret as distributions over classes.

Building a Maxent Model

- We define features (indicator functions) over data points
  - Features represent sets of data points which are distinctive enough to deserve model parameters.
    - Words, but also "word contains number", "word ends with *ing*", etc.
- We will simply encode each Φ feature as a unique String
  - A datum will give rise to a set of Strings: the active Φ features
  - Each feature $f_i(c, d) \equiv [\Phi(d) \wedge c = c_j]$ gets a real number weight
- We concentrate on Φ features but the math uses i indices of f_i