

Faculty for Informatics

Technical
University
of Munich



Natural Language Processing

IN2361

PD Dr. Georg Groh

Social Computing
Research Group

Chapter 7

Logistic Regression

- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

Repetition from ML1: Logistic Regression

- **Generative** classifier: $\hat{y} = \underset{y}{\operatorname{argmax}} P(y|x) = \underset{y}{\operatorname{argmax}} P(x|y)P(y)$
vs. **discriminative** classifier: $\hat{y} = \underset{y}{\operatorname{argmax}} P(y|x)$

- **Logistic Regression**: $p(y = 1 | x) = \sigma(x^T w + w_0)$

where x are pattern-vectors or feature-vectors of pattern-vectors
and σ is the Sigmoid fct.

- **Multi-class Logistic Regression** (using softmax fct.):

$$p(y = c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^k e^{w_j \cdot x + b_j}}$$

Features: Example Sentiment

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) \in doc)	3
x_2	count(negative lexicon) \in doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns \in doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(64) = 4.15$

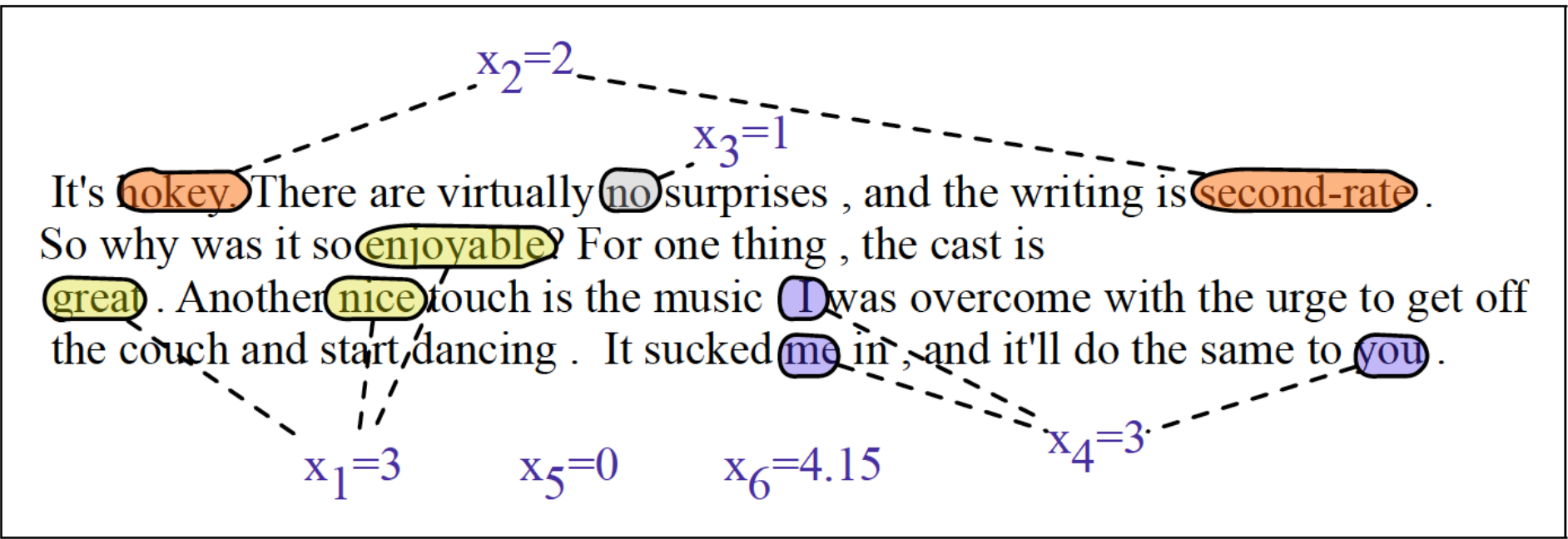


Figure 5.2 A sample mini test document showing the extracted features in the vector x .

Features: Example Period Disambiguation

- goal: using features of words before “.”, decide End of Sentence (EOS) or not EOS
- example features:

$$x_1 = \begin{cases} 1 & \text{if “}Case(w_i) = \text{Lower”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if “}w_i \in \text{AcronymDict”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if “}w_i = \text{St. \& } Case(w_{i-1}) = \text{Cap”} \\ 0 & \text{otherwise} \end{cases}$$

feature interaction
(complex feature)

Multiple Classes and Class Specific Features

- **Multi-class** Logistic Regression (using **class specific features**):

$$p(y = c|x) = p(c|x) = \frac{\exp\left(\sum_{i=1}^N w_i f_i(c, x)\right)}{\sum_{c' \in C} \exp\left(\sum_{i=1}^N w_i f_i(c', x)\right)} \quad \text{softmax}$$

where $f_i(c, x)$ is the i -th element of the class-specific feature-vector of x

- using **class-specific features** \leftrightarrow using a separate set of weights for each class

$$p(y = c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^k e^{w_j \cdot x + b_j}}$$

Class-Specific Features $f_i(c, x)$

- **example case:** ternary sentiment analysis + binary features

Var	Definition	Wt
$f_1(0, x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	-4.5
$f_1(+, x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	2.6
$f_1(-, x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1.3

- in NLP possible: **feature templates**: example: next word prediction (language modelling) $\rightarrow |V|$ classes

$$f_4(c, x) = \begin{cases} 1 & \text{if “}w_{i-1}\text{ends in -an; \& } c = \text{breakfast”} \\ 0 & \text{otherwise} \end{cases}$$

Repetition from ML1: Training Logistic Regression

- Loss function: $L(\hat{y}, y)$ = How much \hat{y} differs from the true y
- Cross Entropy Loss:

$$p(y|x) = \hat{y}^y (1 - \hat{y})^{1-y} \quad \hat{y} = \sigma(w \cdot x + b)$$

$$\begin{aligned} L_{CE}(\hat{y}, y) &= -\log p(y|x) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \\ &= -[y \log \sigma(w \cdot x + b) + (1 - y) \log(1 - \sigma(w \cdot x + b))] \end{aligned}$$

- Cross Entropy Loss for a dataset: $\mathcal{D} = ((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)}))$

$$\begin{aligned} \log \prod_{i=1}^m p(y^{(i)} | x^{(i)}) &= \sum_{i=1}^m \log p(y^{(i)} | x^{(i)}) \\ &= -\sum_{i=1}^m L_{CE}(\hat{y}^{(i)}, y^{(i)}) \end{aligned}$$

Repetition from ML1: Training Logistic Regression

- learning $\theta = w, b$

$$\hat{\theta} = \operatorname{argmin}_{\theta} \frac{1}{m} \sum_{i=1}^m L_{CE}(y^{(i)}, x^{(i)}; \theta)$$

- stochastic gradient descent

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x; \theta), y)$$

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x , y) **returns** θ

where: L is the loss function

f is a function parameterized by θ

x is the set of training inputs $x^{(1)}, x^{(2)}, \dots, x^{(n)}$

y is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \dots, y^{(n)}$

$\theta \leftarrow 0$

repeat T times

For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)

Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output \hat{y} ?

Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$ from the true output $y^{(i)}$?

$g \leftarrow \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize loss ?

$\theta \leftarrow \theta - \eta g$ # go the other way instead

return θ

Repetition from ML1: Regularization

- Avoid overfitting → switch from MLE for w to MAP: assume priors for w :
Regularization

- L2-Regularization (Gaussian prior)

$$\hat{w} = \operatorname{argmax}_w \sum_j \log P(y^{(j)} | x^{(j)}) - \alpha \sum_{i=1}^N w_i^2$$

- L1-(Lasso)-Regularization (Laplace prior): enforces sparse w -vector

$$\hat{w} = \operatorname{argmax}_w \sum_j \log P(y^{(j)} | x^{(j)}) - \alpha \sum_{i=1}^N |w_i|$$

Choosing a Classifier

- General **discriminative vs. generative** discussion (see ML1 or Murphy 8.6)
- Naïve Bayes: surprisingly good for small documents (better than Log.Regr. or SVM) but naïve conditional independence assumption is pretty unrealistic in most cases
- Logistic Regression: can robustly deal well with correlated features (just distribute weight elements accordingly); may work better on large documents
- **main contribution: good features!**



- (1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Oct. 2019); Online: <https://web.stanford.edu/~jurafsky/slp3/> (URL Oct 2019) (this slideset is especially based on chapter 5)
- (2) Powerpoint slides from Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft); Online: <https://web.stanford.edu/~jurafsky/slp3/> (URL, Oct 2018)

Recommendations for Studying

- minimal approach:

work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

== standard approach