School of CIT
Social Computing Research Group

Technical
University
of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

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: $p(y = 1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{x}^T \boldsymbol{w} + w_0)$

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

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

$$p(\mathbf{y}_k = 1|\mathbf{x}) = \frac{\exp(\mathbf{w}_k \cdot \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j \cdot \mathbf{x} + b_j)}$$

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

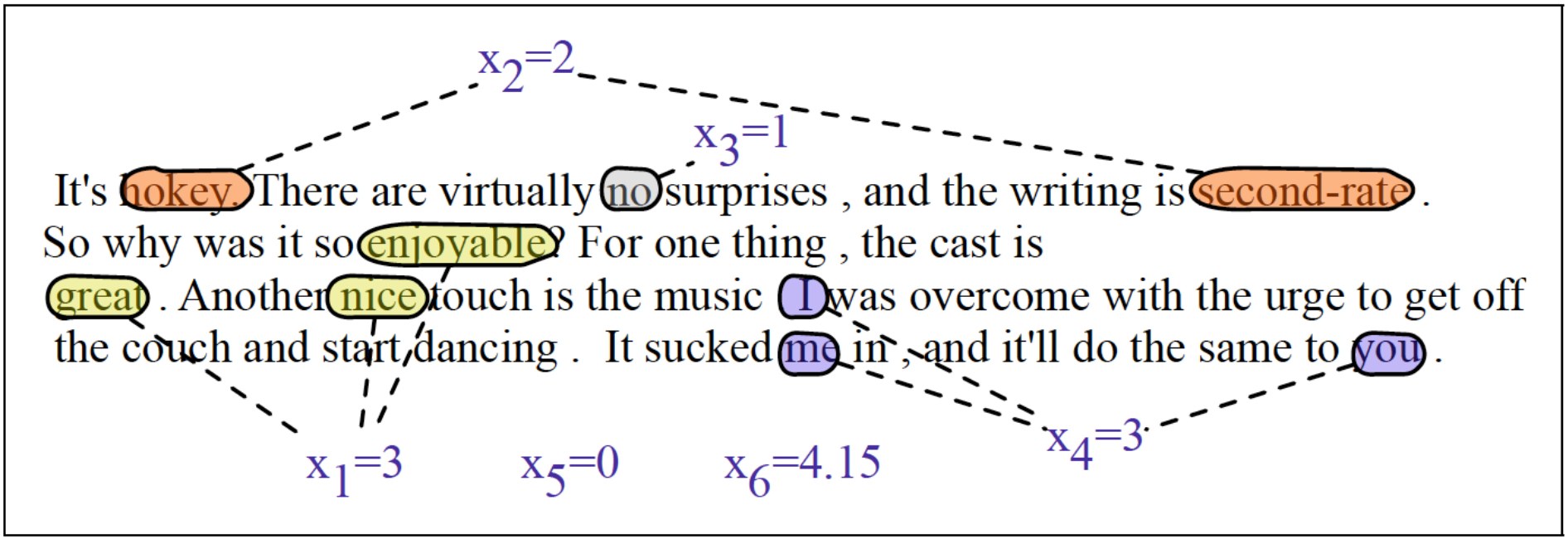

Figure 5.2 A sample mini test document showing the extracted features in the vector x.

Features: Example Period Disambiguation

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

- example features:

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

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

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

feature interaction (complex feature)

Repetition from ML1: Training Logistic Regression

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

- Cross Entropy Loss:

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

$$L_{CE}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$$

$$= -[y \log \sigma(w \cdot x + b) + (1-y) \log (1 - \sigma(w \cdot x + b))]$$

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

$$\log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)})$$

$$= -\sum_{i=1}^{m} L_{CE}(\hat{y}^{(i)}, y^{(i)})$$

Repetition from ML1: Training Logistic Regression

- learning $\theta = w, b$

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

- 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), ..., x^(n)
    #        y is the set of training outputs (labels) y^(1), y^(2), ..., y^(n)

θ ← 0
repeat T times
   For each training tuple (x^(i), y^(i)) (in random order)
   Compute ŷ^(i) = f(x^(i); θ)     # What is our estimated output ŷ?
   Compute the loss L(ŷ^(i), y^(i))  # How far off is ŷ^(i) from the true output y^(i)?
   g ← ∇_θ L(f(x^(i); θ), y^(i))     # How should we move θ to maximize loss ?
   θ ← θ − η 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)

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

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

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

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)
  - naïve conditional independence assumption is pretty unrealistic in most cases;
  - cannot deal well with correlated features ("naïve"←→independence assumption on features, will add up evidences from correlated features)
  - fast training

- Logistic Regression:
  - can robustly deal well with correlated features (just distribute weight elements accordingly)
  - may work better on large documents

- main contribution: good features!

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Jan 2022); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL Oct 2022) (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 2022)

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