Supervised Classification with Logistic Regression

CMSC 470
Marine Carpuat
The Perceptron
What you should know

• What is the underlying function used to make predictions
• Perceptron test algorithm
• Perceptron training algorithm
• How to improve perceptron training with the averaged perceptron
• Fundamental Machine Learning Concepts:
  • train vs. test data; parameter; hyperparameter; generalization; overfitting; underfitting.
• How to define features
Logistic Regression for **Binary** Classification

Images and examples: Jurafsky & Martin, SLP 3 Chapter 5
From Perceptron to Probabilities: the Logistic Regression classifier

- The perceptron gives us a prediction $y$, and the activation can take any real value

- What if we want a probability $p(y|x)$ instead?
The sigmoid function
(aka the logistic function)

\[ y = \frac{1}{1 + e^{-z}} \]
From Perceptron to Probabilities for Binary Classification

\[
P(y = 1) = \sigma(w \cdot x + b)
\]
\[
= \frac{1}{1 + e^{-(w \cdot x + b)}}
\]

\[
P(y = 0) = 1 - \sigma(w \cdot x + b)
\]
\[
= 1 - \frac{1}{1 + e^{-(w \cdot x + b)}}
\]
\[
= \frac{e^{-(w \cdot x + b)}}{1 + e^{-(w \cdot x + b)}}
\]
Making Predictions with the Logistic Regression Classifier

• Given a test instance $x$, predict class 1 if $P(y=1|x) > 0.5$, and 0 otherwise

$$\hat{y} = \begin{cases} 1 \text{ if } P(y = 1|x) > 0.5 \\ 0 \text{ otherwise} \end{cases}$$

• Inputs $x$ for which $P(y=1|x) = 0.5$ constitute the **decision boundary**
Example: Sentiment Classification with Logistic Regression

• 2 classes: 1 (positive sentiment) or 0 (negative sentiment)
• Examples are movie reviews
• Features:

<table>
<thead>
<tr>
<th>Var</th>
<th>Definition</th>
<th>Value in Fig. 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>count(positive lexicon $\in$ doc)</td>
<td>3</td>
</tr>
<tr>
<td>$x_2$</td>
<td>count(negative lexicon $\in$ doc)</td>
<td>2</td>
</tr>
</tbody>
</table>
| $x_3$ | \[
\begin{cases}
1 & \text{if "no" } \in \text{ doc} \\
0 & \text{otherwise}
\end{cases}
\] | 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) | $\ln(64) = 4.15$ |
Constructing the feature vector $\mathbf{x}$ for one example

<table>
<thead>
<tr>
<th>Var</th>
<th>Definition</th>
<th>Value in Fig. 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>count(positive lexicon) $\in$ doc</td>
<td>3</td>
</tr>
<tr>
<td>$x_2$</td>
<td>count(negative lexicon) $\in$ doc</td>
<td>2</td>
</tr>
</tbody>
</table>
| $x_3$ | \( \begin{cases} 
1 & \text{if "no" } \in \text{doc} \\
0 & \text{otherwise} 
\end{cases} \) | 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) | $\ln(64) = 4.15$ |

It's hokey. There are virtually no surprises, and the writing is second-rate. So why was it so enjoyable? For one thing, the cast is great. Another nice touch is the music. I was overcome with the urge to get off the couch and start dancing. It sucked me in, and it'll do the same to you.
Example: Sentiment Classification with Logistic Regression

• Assume we are given the parameters of the classifier
  \[ w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \]
  \[ b = 0.1 \]

• On this example:
  \[ P(y=1 | x) = 0.69 \]
  \[ P(y=0 | x) = 0.31 \]

It's hokey. There are virtually no surprises, and the writing is second-rate. So why was it so enjoyable? For one thing, the cast is great. Another nice touch is the music. I was overcome with the urge to get off the couch and start dancing. It sucked me in, and it'll do the same to you.
Learning in Logistic Regression

• How are parameters of the model \(w\) and \(b\) learned?

• This is an instance of supervised learning
  • We have labeled training examples

• We want model parameters such that
  • For training examples \(x\)
  • The prediction of the model \(\hat{y}\)
  • is as close as possible to the true \(y\)
Learning in Logistic Regression

• How are parameters of the model \( (w \text{ and } b) \) learned?

• This is an instance of supervised learning
  • We have labeled training examples

• We want model parameters such that
  • For training examples \( x \), the prediction of the model \( \hat{y} \) is as close as possible to the true \( y \)
  • Or equivalently so that the distance between \( \hat{y} \) and \( y \) is small
Ingredients required for training

• Loss function or cost function
  • A measure of distance between classifier prediction and true label for a given set of parameters
    
    \[ L(\hat{y}, y) = \text{How much } \hat{y} \text{ differs from the true } y \]

• An algorithm to minimize this loss
  • Here we’ll introduce stochastic gradient descent
The cross-entropy loss function

• Loss function used for logistic regression and often for neural networks

• Defined as follows:

\[ L_{CE}(w, b) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))] \]
Deriving the cross-entropy loss function

• Conditional maximum likelihood
  • Choose parameters that maximize the log probability of true labels $y$ given inputs $x$

\[
\log p(y|x) = \log \left[ \hat{y}^y (1 - \hat{y})^{1-y} \right] \\
= y \log \hat{y} + (1 - y) \log (1 - \hat{y})
\]

• Cross-entropy loss is defined as

\[
L_{CE}(\hat{y}, y) = - \log p(y|x) = - [y \log \hat{y} + (1 - y) \log (1 - \hat{y})]
\]
Example: Sentiment Classification with Logistic Regression

• Assume we are given the parameters of the classifier

\[ w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \]

\[ b = 0.1 \]

• On this example:

\[ P(y=1|x) = 0.69 \]

\[ P(y=0|x) = 0.31 \]

\[ \text{Loss}(w,b) = -\log(0.69) = 0.37 \]
Example: Sentiment Classification with Logistic Regression

• Assume we are given the parameters of the classifier
  \[ w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \]
  \[ b = 0.1 \]

• If the example was negative (\( y=0 \))
  \[ \text{Loss}(w,b) = -\log(0.31) = 1.17 \]
Gradient Descent

• Goal:
  • find parameters $\theta = w, b$
  • Such that

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

• For logistic regression, the loss is \textbf{convex}
Illustrating Gradient Descent

The gradient indicates the direction of greatest increase of the cost/loss function.

Gradient descent finds parameters \((w,b)\) that decrease the loss by taking a step in the opposite direction of the gradient.

Figure 5.4 Visualization of the gradient vector in two dimensions \(w\) and \(b\).
function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, $x$, $y$) returns $\theta$

# where: $L$ is the loss function
# $f$ is a function parameterized by $\theta$
# $x$ is the set of training inputs $x^{(1)}$, $x^{(2)}$, ..., $x^{(n)}$
# $y$ is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$, ..., $y^{(n)}$

$\theta \leftarrow 0$

repeat til done  # see caption
    For each training tuple $(x^{(i)}$, $y^{(i)})$ (in random order)
    1. Optional (for reporting):  # How are we doing on this tuple?
        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 $y^{(i)}$?
    2. $g \leftarrow \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$  # How should we move $\theta$ to maximize loss?
    3. $\theta \leftarrow \theta - \eta g$  # Go the other way instead

return $\theta$

Figure 5.5  The stochastic gradient descent algorithm. Step 1 (computing the loss) is used to report how well we are doing on the current tuple. The algorithm can terminate when it converges (or when the gradient $< \varepsilon$), or when progress halts (for example when the loss starts going up on a held-out set).
The gradient for logistic regression

\[ L_{CE}(w, b) = - [y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))] \]

\[ \frac{\partial L_{CE}(w, b)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j \]

Note: the detailed derivation is available in the reading (SLP3 Chapter 5, section 5.8)
Logistic Regression
What you should know
How to make a prediction with logistic regression classifier
How to train a logistic regression classifier

Machine learning concepts:
Loss function
Gradient Descent Algorithm