Logistic Regression

Logisitc regression defines the conditional probability directly and is a **discriminative model** rather than a generative model.

- Start with the scoring function $\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y)$ that measures the comptability between the features $\boldsymbol{x}$ and y.
- To make sure it's not negative, we exponentiate it and get $\exp \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y)$
- We then normalize it by dividing it over all possible labels $y \in \mathcal{Y}$ and get a probability.

$$p(y|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\exp \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y)}{\sum_{y' \in \mathcal{Y}} \exp \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y')}$$

Logistic Regression

The weights are estimated by **maximum conditional likelihood**. Give a data set $D = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, the maximum conditional likelihood is:

$$
\begin{aligned}
\log p(y^{(1:N)} | \boldsymbol{x}^{(1:N)}; \boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \log \sum_{y' \in \mathcal{Y}} \exp \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')
\end{aligned}
$$

Or they can be estimated by minimizing the logistic regression loss (or cross-entropy loss):

$$
\mathcal{L}_{\text{LOGREG}} = - \sum_{i=1}^{N} \left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \log \sum_{y' \in \mathcal{Y}} \exp \left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y') \right) \right)
$$

Logistic Regression Objective

Loss of a single sample:

Assignment Project Exam Help

$$\ell_{\text{LOGREG}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\right)$$

https://powcoder.com

Add WeChat powcoder

Gradient of Logistic Regression

The gradient with respect to the loss of a single example

$$
\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}} &= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\right)} \\
&\quad \times \sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\right) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y') \\
&= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \frac{exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\right)}{\sum_{y'' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\right)} \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y') \\
&= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} P(y' | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y') \\
&= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + E_{Y|X}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]
\end{aligned}
$$

Application of the chain rule in calculus, expectation

Gradient of Logistic Regression

$$
\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}} &= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} P(y' | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y') \\
&= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + E_{y|x}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]
\end{aligned}
$$

- This is a very nice intuitive result
  - The gradient equals to the difference between the expected feature counts under the current model $E_{y|x}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]$ and the observed feature counts $\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)})$
  - The loss is minimized if the feature counts under the current model and the observed feature counts are the same
- The power of Logistic Regression model is that you can use arbitrary number of features without making any independence assumptions. The allows creative feature engineering to improve the performance of the model.

Digression: Expectation

Expectation is the mean or average of a random variable, for discrete case. Let X be a random variable:

$$\mathbb{E}[X] = \sum_{x \in X} xP(x) = \mu$$

$$\mathbb{E}[g(X)] = \sum_{x} g(x)p(x)$$

$$\mathbb{E}[g(X, Y)] = \sum_{x,y} g(x, y)p(x, y)$$

Let X be the random variable which is the value of rolling a single dice:

$$\mathbb{E}[x] = \sum_{x=1}^{6} xP(y) = \frac{1}{6} \sum_{x=1}^{6} x = \frac{21}{6} = 3.5$$

Digression: Linearity of Expectations

$$
\begin{aligned}
\mathbb{E}[X+Y] &= \sum_{x,y} P(x,y)(x+y) \\
&= \sum_{x,y} P(x,y)x + \sum_{x,y} P(x,y)y \\
&= \sum_{x} \left(\sum_{y} P(x,y) \right) x \sum_{y} \left(\sum_{x} P(x,y) \right) y \\
&= \sum_{x} P(x)x + \sum_{y} P(y)y \\
&= \mathbb{E}[X] + \mathbb{E}[Y]
\end{aligned}
$$

Let the Y be the random variable for the sum of two dice rolled. Expected value of Y:

$$\mathbb{E}[Y] = \mathbb{E}[X] + \mathbb{E}[X]$$

When two random variables are independent:

$$\mathbb{E}[X, Y] = \mathbb{E}[X] \times \mathbb{E}[X]$$

Digression: Variance

- Variance of a random variable is a measure of whether the values of the random variable tend to be consistent over trials/experiments or whether they vary a lot

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X]^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

$$\begin{aligned}
\mathbb{E}[(X - \mathbb{E}[X]^2] &= \sum_x (x - \mathbb{E}[X])^2 P(x) \\
&= \sum_x (x^2 - 2\mathbb{E}[X]x + (\mathbb{E}[X])^2)P(x) \\
&= \sum_x x^2 P(x) - 2\mathbb{E}[X] \sum_x xP(x) + (\mathbb{E}[X])^2 \sum_x P(x) \\
&= \mathbb{E}[X^2] - 2(\mathbb{E}[X])^2 + (\mathbb{E}[X])^2 \\
&= \mathbb{E}[X^2] - (\mathbb{E}[X])^2
\end{aligned}$$

- σ^2 denotes the variance, and σ is the standard deviation.

Regularization

- $L2$ regularization: add a multiple of the squared norm $\frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$ to the minimization objective
- Regularization forces the estimator to trade off performance on the training data against the norm of the weights, and thus helps relieve overfitting.

The overall loss of a training set with a regularization term:

$$\mathcal{L}_{\text{LOGREG}} = \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 - \sum_{i=1}^{N}\left(\boldsymbol{\theta}\cdot\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \log\sum_{y'\in\mathcal{Y}}\exp\left(\boldsymbol{\theta}\cdot\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)\right)\right)$$

Regularized gradient

Derivative of the $L2$ term:

$$\frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 = \frac{\lambda}{2}\left(\left(\sum_{j=1}^{N}\theta_j^2\right)^{\frac{1}{2}}\right)^2 = \frac{\lambda}{2}\sum_{j=1}^{N}\theta_j^2$$

$$\frac{\partial}{\partial\theta_k}\frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 = \frac{\partial}{\partial\theta_k}\frac{\lambda}{2}\sum_{j=1}^{N}\theta_j^2 = \frac{\lambda}{2}\sum_{j=1}^{N}\frac{\partial}{\partial\theta_k}\theta_j^2 = \lambda\theta_k$$

$$\frac{\partial}{\partial\theta_k}\theta_j^2 = \begin{cases} 2\theta_k & \text{if } j=k \\ 0 & \text{otherwise} \end{cases}$$

Gradient of regularized loss:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{LOGREG}} = \lambda\boldsymbol{\theta} - \sum_{i=1}^{N}\left(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - E_{Y|X}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y')]\right)$$

Batch Optimization vs online optimization

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Gradient Descent vs Stochastic Gradient Descent. In **batch optimization**, each update to the weights is based on the entire dataset. One such algorithm is **gradient descent**, which iteratively updates the weights,

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(t)} \nabla_{\boldsymbol{\theta}} \mathcal{L}$$

where $\nabla_{\boldsymbol{\theta}} \mathcal{L}$ is the gradient computed over the entire training set, $\eta^{(t)}$ is the **learning rate** at iteration t.

Variations of Gradient Descent

- **Online learning** algorithms make updates as going through the data. In **stochasttic gradient descent**, the approximate gradient is computed by sampling a single instance, and an update is made immediately.
- In **mini-batch** stochastic gradient descent, the gradient is computed over a small subset of instances.

Generalized gradient descent algorithm

The function BATCHER partitions the training set into B batches such that each instance appears in exactly one batch. In stochastic gradient descent, $B = N$; in gradient descent, $B = 1$; in mini-batch stochastic gradient descent, $1 < B < N$.

procedure GRDNTDSCNT($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}, \ell, \eta^{(1:\infty)}$, BATCHER, T_{MAX})

    $t \leftarrow 0$

    $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$

    repeat

        $(\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \cdots, \boldsymbol{b}^{(B)}) \leftarrow$ BATCHER(N)

        for $n \in \{1, 2, \cdots, B\}$ **do**

            $t \leftarrow t + 1$

            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} - \eta^{(t)} \nabla \mathcal{L}(\boldsymbol{\theta}^{(t-1)}; \boldsymbol{x}^{(b_1^{(n)}, b_2^{(n)}, \cdots)}, \boldsymbol{y}^{(b_1^{(n)}, b_2^{(n)}, \cdots)})$

            if Converged($\boldsymbol{\theta}^{(1,2,\cdots,t)}$) **then return** $\boldsymbol{\theta}^{(t)}$

            end if

        end for

    until $t = T_{MAX}$

end procedure

Binary logistic regression is a special case of multinominal logistic regression

$$
\begin{aligned}
P(y=1|\boldsymbol{x}) &= \frac{\exp(\sum_k \theta_k f_k(y=1,\boldsymbol{x}))}{\exp(\sum_k \theta_k f_k(y=1,\boldsymbol{x})) + \exp(\sum_k \theta_k f_k(y=0,\boldsymbol{x}))} \\
&= \frac{\exp(\sum_k \theta_k f_k(y=1,\boldsymbol{x}))}{\exp(\sum_k \theta_k f_k(y=1,\boldsymbol{x})) + \exp(\sum_k \theta_k f_k(y=0,\boldsymbol{x}))} \\
&\quad \times \frac{\exp(-\sum_k \theta_k f_k(y=1,\boldsymbol{x}))}{\exp(-\sum_k \theta_k f_k(y=1,\boldsymbol{x}))} \\
&= \frac{1}{1+\exp(\sum_k \theta_k f_k(y=0,\boldsymbol{x})) \times \exp(-\sum_k \theta_k f_k(y=1,\boldsymbol{x}))} \\
&= \frac{1}{1+\exp(\sum_k \theta_k (f_k(y=0,\boldsymbol{x}) - f_k(y=1,\boldsymbol{x})))} \\
&= \frac{1}{1+\exp(\sum_k -\theta_k f'_k(\boldsymbol{x}))} = \sigma(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}))
\end{aligned}
$$

Note: For binary classification, you only need to pay attention to the positive class.

Logistic Regression: features and weights

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

e.g., $f_1(x, y) = 1$ if x= "not" $\wedge$ y= "POS", $\theta_1 = -1$

Note: The features $\boldsymbol{f}(\boldsymbol{x}, y)$ and $\boldsymbol{\theta}$ are presented in a table rather than a vector due to limitation of space. Mathematically they should still be viewed as vectors

Logistic Regression: Inference

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Test instance: "funny, smart, and visually stunning"

$$p(y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y))}{\sum_y' \exp(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}, y'))} = \frac{\exp(\sum_k \theta_k f_k(\boldsymbol{x}, y))}{\sum_{y'} \exp(\sum_k \theta_k f_k(\boldsymbol{x}, y'))}$$

Logistic Regression: Inference

$$f(\boldsymbol{x}, y)$$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$$\boldsymbol{\theta}$$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Test instance: “funny, smart, and visually stunning”

$$p(y = POS|\boldsymbol{x})$$

$$= \frac{\exp(\theta_4 f_4 + \theta_{25} f_{25})}{\exp(\theta_4 f_4 + \theta_{25} f_{25}) + \exp(\theta_5 f_5 + \theta_{26} f_{26}) + \exp(\theta_6 f_6 + \theta_{27} f_{27})}$$

$$= \frac{\exp(2 + 1.2)}{\exp(2 + 1.2) + \exp(-2 + 0.8) + \exp(-0.9 + 0.4)} = 0.9643$$

Logistic Regression: Inference

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Test instance: "funny, smart, and visually stunning"

$$p(y = NEG|\boldsymbol{x})$$

$$= \frac{\exp(\theta_5 f_5 + \theta_{26} f_{26})}{\exp(\theta_4 f_4 + \theta_{25} f_{25}) + \exp(\theta_5 f_5 + \theta_{26} f_{26}) + \exp(\theta_6 f_6 + \theta_{27} f_{27})}$$

$$= \frac{\exp(-2 + 0.8)}{\exp(2 + 1.2) + \exp(-2 + 0.8) + \exp(-0.9 + 0.4)} = 0.0118$$

Logistic Regression: Inference

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	-0.8	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Test instance: "funny, smart, and visually stunning"

$$p(y = NEU|\boldsymbol{x})$$

$$= \frac{\exp(\theta_6 f_6 + \theta_{27} f_{27})}{\exp(\theta_4 f_4 + \theta_{25} f_{25}) + \exp(\theta_5 f_5 + \theta_{26} f_{26}) + \exp(\theta_6 f_6 + \theta_{27} f_{27})}$$

$$= \frac{\exp(-9 + 0.4)}{\exp(2 + 1.2) + \exp(-2 + 0.8) + \exp(-0.9 + 0.4)} = 0.0238$$

Logistic Regression: Parameter estimation

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{LOGREG}} = -\sum_{i=1}^{N} \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{i=1}^{N} \sum_{y' \in Y} p(y' | \boldsymbol{x}^{(i)}) \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

Logistic Regression: parameter estimation

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = “not funny at all”

$$p(y = NEG|x)$$

$$= \frac{\exp(\theta_2 f_2 + \theta_5 f_5 + \theta_{26} f_{26})}{\exp(\theta_1 f_1 + \theta_4 f_4 + \theta_{25} f_{25})) + \exp(\theta_2 f_2 + \theta_5 f_5 + \theta_{26} f_{26})) + \exp(\theta_3 f_3 + \theta_6 f_6 + \theta_{27} f_{27}))}$$

$$= \frac{\exp(2 - 2 + 0.8)}{\exp(2 - 1 + 1.2) + \exp(2 - 2 + 0.8) + \exp(-0.9 - 0.4 + 0.4)} = 0.1909$$

Logistic Regression: parameter estimation

$$\boldsymbol{f}(\boldsymbol{x}, y)$$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$$\boldsymbol{\theta}$$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$\begin{aligned} & p(y = POS|x) \\ &= \frac{\exp(\theta_1 f_1 + \theta_4 f_4 + \theta_{25} f_{25}))}{\exp(\theta_1 f_1 + \theta_4 f_4 + \theta_{25} f_{25})) + \exp(\theta_2 f_2 + \theta_5 f_5 + \theta_{26} f_{26})) + \exp(\theta_3 f_3 + \theta_6 f_6 + \theta_{27} f_{27}))} \\ &= \frac{\exp(2 - 1 + 1.2)}{\exp(2 - 1 + 1.2) + \exp(2 - 2 + 0.8) + \exp(-0.9 - 0.4 + 0.4)} = 0.7742 \end{aligned}$$

Logistic Regression: Parameter estimation

$\boldsymbol{f}(\boldsymbol{x}, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$p(y = NEU|x)$$

$$= \frac{\exp(\theta_3 f_3 + \theta_6 f_6 + \theta_{27} f_{27}))}{\exp(\theta_1 f_1 + \theta_4 f_4 + \theta_{25} f_{25})) + \exp(\theta_2 f_2 + \theta_5 f_5 + \theta_{26} f_{26})) + \exp(\theta_3 f_3 + \theta_6 f_6 + \theta_{27} f_{27}))}$$

$$= \frac{\exp(-0.9 - 4 + 0.4)}{\exp(2 - 1 + 1.2) + \exp(2 - 2 + 0.8) + \exp(-0.9 - 0.4 + 0.4)} = 0.0349$$

Logistic Regression: Parameter estimation

$f(x, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}

$\boldsymbol{\theta}$

POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$gradient[\theta_1] = -0 + 0.7742, gradient[\theta_4] = -0 + 0.7742$$
$$gradient[\theta_2] = -1 + 0.1909, gradient[\theta_5] = -1 + 0.1909$$
$$gradient[\theta_3] = -0 + 0.0349, gradient[\theta_6] = -0 + 0.0349$$

Logistic Regression: Parameter estimation

$f(x, y)$

	not	funny	painful	ok	overall	story	good	jokes	*bias*
POS	f_1	f_4	f_7	f_{10}	f_{13}	f_{16}	f_{19}	f_{22}	f_{25}
NEG	f_2	f_5	f_8	f_{11}	f_{14}	f_{17}	f_{20}	f_{23}	f_{26}
NEU	f_3	f_6	f_9	f_{12}	f_{15}	f_{18}	f_{21}	f_{24}	f_{27}
					$\boldsymbol{\theta}$				
POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$gradient[\theta_{25}] = -0 + 0.7742$$
$$gradient[\theta_{26}] = -1 + 0.1909$$
$$gradient[\theta_{27}] = -0 + 0.0349$$

Note: The "bias" is the feature that always fires.

Some observations about the gradient for Logistic Regression

- The gradient of all features that predict the same label is updated by the same amount.
- A feature is penalized or rewarded by an amount that is proportional to how badly off the prediction is.

Discussion question:

- Assuming that the weights $\boldsymbol{\theta}$ are initialized as $\mathbf{0}$, how would the first iteration of the Logistic Regression Training proceed?

Adding a regularization term

	θ								
POS	-1	2	-2.5	0.5	0.2	.08	1.5	0.8	1.2
NEG	2	-2	1.8	-0.5	0.1	-0.6	-2	-1.2	0.8
NEU	-0.4	-0.9	-1.5	2	1	-0.2	-1.2	-0.3	0.4

Training instance: y = NEG, x = "not funny at all"

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{LOGREG}} = \lambda \boldsymbol{\theta} - \sum_{i=1}^{N} \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{i=1}^{D} \sum_{y' \in Y} p(y' | \boldsymbol{x}^{(i)}) \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

Let $\lambda = 0.5$:

$\text{gradient}[\theta_1] = -0.5 - 0 + 0.7742, \text{gradient}[\theta_4] = 1 - 0 + 0.7742$

$\text{gradient}[\theta_2] = 1 - 1 + 0.1909, \text{gradient}[\theta_5] = -1 - 1 + 0.1909$

$\text{gradient}[\theta_3] = -0.2 - 0 + 0.0349, \text{gradient}[\theta_6] = -0.45 - 0 + 0.0349$