Multi-Class Logistic Regression and Perceptron

Instructor: Wei Xu

Some slides adapted from Dan Jurfasky, Brendan O'Connor and Marine Carpuat

MultiClass Classification

- Q: what if we have more than 2 categories?
  - Sentiment: Positive, Negative, Neutral
  - Document topics: Sports, Politics, Business, Entertainment, …

Q: How to easily do Multi-label classification?

Two Types of MultiClass Classification

- Multi-label Classification
  - each instance can be assigned more than one labels

- Multinominal Classification
  - each instance appears in exactly one class (classes are exclusive)

Multinominal Classification

- Pretty straightforward with Naive Bayes.

$$P(\text{spam}|D) \propto P(\text{spam}) \prod_{w \in D} P(w|\text{spam})$$

Log-Linear Models

$$P(y|x) \propto e^{w \cdot f(x,y)}$$

$$P(y|x) = \frac{1}{Z(w)} e^{w \cdot f(x,y)}$$

Multinominal Logistic Regression

$$P(y|x) \propto e^{w \cdot f(x,y)}$$

$$P(y|x) = \frac{1}{Z(w)} e^{w \cdot f(x,y)}$$

$$\boxed{P(y|x) = \frac{e^{w \cdot f(x,y)}}{\sum_{y' \in Y} e^{w \cdot f(x,y')}}}$$

Multinominal Logistic Regression

- Binary (two classes):
  - We have one feature vector that matches the size of the vocabulary
- Multi-class in practice:
  - one weight vector for each category

$$w_{\text{pos}} \qquad w_{\text{neg}} \qquad w_{\text{neut}}$$

Can represent this in practice with one giant weight vector and repeated features for each category.

Maximum Likelihood Estimation

$$w_{\mathrm{MLE}} = \mathrm{argmax}_w \log P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$$

$$= \mathrm{argmax}_w \sum_i \log P(y_i | x_i; w)$$

$$= \mathrm{argmax}_w \sum_i \log \frac{e^{w \cdot f(x_i, y_i)}}{\sum_{y' \in Y} e^{w \cdot f(x_i, y')}}$$

Multiclass LR Gradient

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} \sum_{y \in Y} f_j(y, d_i) P(y|d_i)$$

(a.k.a) Softmax Regression

Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Article Talk Read Edit View history Search Wikipedia

Softmax function

From Wikipedia, the free encyclopedia

In mathematics, the **softmax function**, or **normalized exponential function**,[1]:198 is a generalization of the logistic function that "squashes" a K-dimensional vector $\mathbf{z}$ of arbitrary real values to a K-dimensional vector $\sigma(\mathbf{z})$ of real values in the range (0, 1) that add up to 1. The function is given by

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

(a.k.a) Maximum Entropy Classifier

- or MaxEnt

- Math proof of “LR=MaxEnt”:
  - [Klein and Manning 2003]
  - [Mount 2011]

http://www.win-vector.com/dfiles/LogisticRegressionMaxEnt.pdf

Perceptron Algorithm

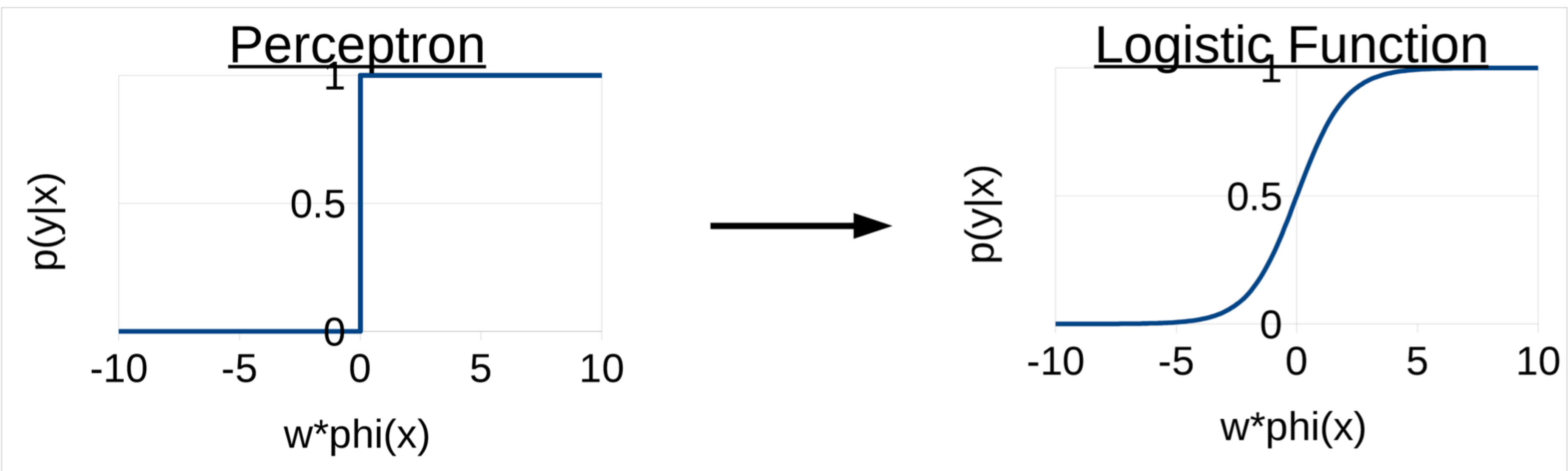
- Very similar to logistic regression
- Not exactly computing gradient

[Rosenblatt 1957]

http://www.peterasaro.org/writing/neural_networks.html

Perceptron Algorithm

- Very similar to logistic regression
- Not exactly computing gradient



$$P(y=1|x)=1 \text{ if } \mathbf{w}\cdot\boldsymbol{\varphi}(x)\geq 0$$
$$P(y=1|x)=0 \text{ if } \mathbf{w}\cdot\boldsymbol{\varphi}(x)<0$$

$$P(y=1|x)=\frac{e^{\mathbf{w}\cdot\boldsymbol{\varphi}(x)}}{1+e^{\mathbf{w}\cdot\boldsymbol{\varphi}(x)}}$$

Online Learning

- Update parameters for each training example (when predication is wrong)

```
for I iterations
    for each labeled pair x, y in the data
        phi = CREATE_FEATURES(x)
        y' = PREDICT_ONE(w, phi)
        if y' != y
            UPDATE_WEIGHTS(w, phi, y)
```

Online Learning

- The Perceptron is an online learning algorithm.
- Logistic Regression is not:

$$w_{\mathrm{MLE}} = \mathrm{argmax}_w \log P(y_1, \ldots, y_d | x_1, \ldots, x_d; w)$$

Perceptron Algorithm

- Very similar to logistic regression
- Not exactly computing gradient

```
Initalize weight vector w = 0
Loop for K iterations
    Loop For all training examples x_i
        if sign(w * x_i) != y_i
            w += (y_i - sign(w * x_i)) * x_i
```

Perceptron Notes

- Guaranteed to converge if the data is linearly separable
- Only hyperparameter is maximum number of iterations
- Parameter averaging will greatly improve performance

Differences between LR and Perceptron

- Online learning vs. Batch

- Perceptron doesn’t always make updates

MAP-based learning (perceptron)

$$\frac{\partial \mathcal{L}}{\partial w_j} \approx \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} f_j(\arg\max_{y \in Y} P(y|d_i), d_i)$$

Online Learning (perceptron)

- Rather than making a full pass through the data, compute gradient and update parameters after each training example.

- Gradients will be less accurate, but the overall effect is to move in the right direction

- Often works well and converges faster than batch learning

MultiClass Perceptron Algorithm

```
Initalize weight vector w = 0
Loop for K iterations
    Loop For all training examples x_i
        y_pred = argmax_y w_y * x_i
        if y_pred != y_i
            w_y_gold += x_i
            w_y_pred -= x_i
```

Q: what if there are only 2 categories?

$$P(y = j | x_i) = \frac{e^{w_j \cdot x_i}}{\sum_k e^{w_k \cdot x_i}}$$

Q: what if there are only 2 categories?

$$P(y=1|x) = \frac{e^{w_1 \cdot x}}{e^{w_0 \cdot x + w_1 \cdot x - w_1 \cdot x} + e^{w_1 \cdot x}}$$

Q: what if there are only 2 categories?

$$P(y=1|x) = \frac{e^{w_1 \cdot x}}{e^{w_0 \cdot x - w_1 \cdot x} e^{w_1 \cdot x} + e^{w_1 \cdot x}}$$

Q: what if there are only 2 categories?

$$P(y=1|x) = \frac{e^{w_1 \cdot x}}{e^{w_1 \cdot x}(e^{w_0 \cdot x - w_1 \cdot x} + 1)}$$

Q: what if there are only 2 categories?

$$P(y = 1|x) = \frac{1}{e^{w_0 \cdot x - w_1 \cdot x} + 1}$$

Q: what if there are only 2 categories?

$$P(y = 1|x) = \frac{1}{e^{-w' \cdot x} + 1}$$

Sigmoid (logistic) function

Regularization

- Combating over fitting
- Intuition: don't let the weights get very large

$$w_{\mathrm{MLE}} = \mathrm{argmax}_w \log P(y_1, \ldots, y_d | x_1, \ldots, x_d; w)$$

$$\mathrm{argmax}_w \log P(y_1, \ldots, y_d | x_1, \ldots, x_d; w) - \delta \sum_{i=1}^{V} w_i^2$$

Regularization in the Perceptron Algorithm

- Can’t directly include regularization in gradient

- # of iterations

- Parameter averaging