COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Linear Models: Perceptron, Logistic Regression

CMSC 470

Marine Carpuat

Slides credit: Jacob Eisenstein

Linear Models for Multiclass Classification

Feature function representation

$$\hat{y} = \arg\max_{y} \boldsymbol{\theta}^{\top} \mathbf{f}(\mathbf{x}, y)$$

Weights

Multiclass perceptron

Algorithm 3 Perceptron learning algorithm

procedure PERCEPTRON($\boldsymbol{x}^{(1:N)}, y^{(1:N)}$)
    $t \leftarrow 0$
    $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$
    repeat
        $t \leftarrow t + 1$
        Select an instance i
        $\hat{y} \leftarrow \operatorname{argmax}_y \boldsymbol{\theta}^{(t-1)} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$
        if $\hat{y} \neq y^{(i)}$ **then**
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y})$
        else
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
    until tired
    return $\boldsymbol{\theta}^{(t)}$

Properties of Linear Models we've seen so far

Naïve Bayes

- Batch learning
- Generative model p(x,y)
- Grounded in probability
- Assumes features are independent given class
- Learning = find parameters that maximize likelihood of training data

Perceptron

- Online learning
- Discriminative model score(y|x)
- Guaranteed to converge if data is linearly separable
- But might overfit the training set
- Error-driven learning

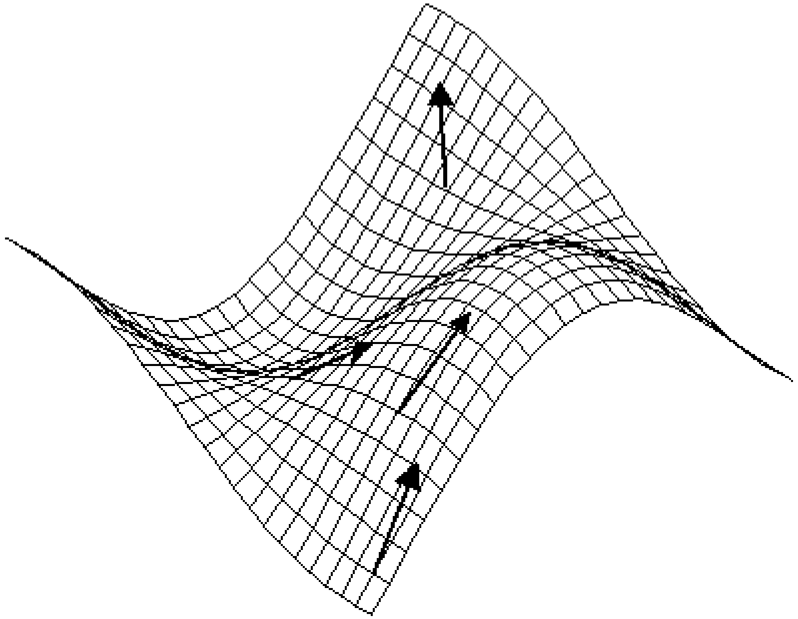
Averaged Perceptron improves generalization

Algorithm 4 Averaged perceptron learning algorithm

procedure AVG-PERCEPTRON($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}$)
    $t \leftarrow 0$
    $\boldsymbol{\theta}^{(0)} \leftarrow 0$
    repeat
        $t \leftarrow t + 1$
        Select an instance i
        $\hat{y} \leftarrow \text{argmax}_y \, \boldsymbol{\theta}^{(t-1)} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$
        if $\hat{y} \neq y^{(i)}$ **then**
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y})$
        else
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
        $\boldsymbol{m} \leftarrow \boldsymbol{m} + \boldsymbol{\theta}^{(t)}$
    until tired
    $\overline{\boldsymbol{\theta}} \leftarrow \frac{1}{t}\boldsymbol{m}$
    return $\overline{\boldsymbol{\theta}}$

Differential Calculus Refresher

- Derivatives
- Chain rule
- Convex functions
- Gradients



Gradient Vectors Shown at Several Points on the Surface of cos(x) sin(y)

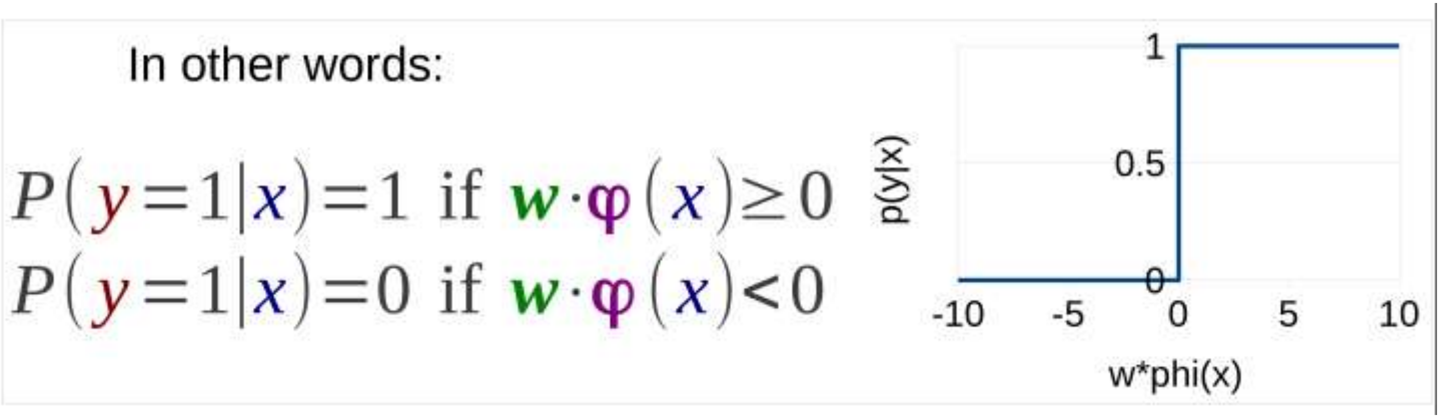
Logistic Regression for **Binary** Classification

Examples & illustrations: Graham Neubig

Perceptron & Probabilities

- What if we want a probability p(y|x)?

- The perceptron gives us a prediction y
  - Let's illustrate this with binary classification

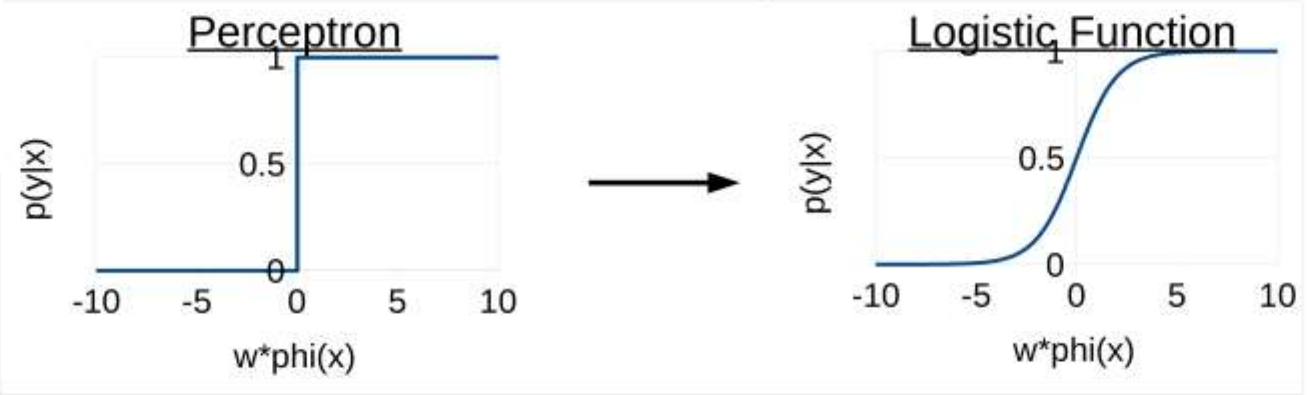


Illustrations: Graham Neubig

The logistic function

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

- x: the input
- **φ(x)**: vector of feature functions $\{\varphi_1(x), \varphi_2(x), \ldots, \varphi_I(x)\}$
- **w**: the weight vector $\{w_1, w_2, \ldots, w_I\}$
- y: the prediction, +1 if "yes", -1 if "no"



- "Softer" function than in perceptron
- Can account for uncertainty
- Differentiable

Logistic regression: how to train?

- Train based on **conditional likelihood**
- Find parameters w that maximize conditional likelihood of all answers y_i given examples x_i

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_i P(y_i | x_i ; \mathbf{w})$$

Stochastic gradient ascent (or descent)

- Online training algorithm

```
create map w
for I iterations
    for each labeled pair x, y in the data
        w += α * dP(y|x)/dw
```

- Update weights for every training example
- Move in direction given by gradient
- Size of update step scaled by learning rate

Gradient of the logistic function

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

$$= \varphi(x)\frac{e^{w\cdot\varphi(x)}}{(1+e^{w\cdot\varphi(x)})^2}$$

$$\frac{d}{dw}P(y=-1|x) = \frac{d}{dw}\left(1-\frac{e^{w\cdot\varphi(x)}}{1+e^{w\cdot\varphi(x)}}\right)$$

$$= -\varphi(x)\frac{e^{w\cdot\varphi(x)}}{(1+e^{w\cdot\varphi(x)})^2}$$

Example: Person/not-person classification problem

Given an introductory sentence in Wikipedia

predict whether the article is about a person

Example: initial update

- Set α=1, initialize **w=0**

x = A site , located in Maizuru , Kyoto $\quad$ y = -1

$$\mathbf{w}\cdot\varphi(x)=0 \qquad \frac{d}{dw}P(y=-1|x) = -\frac{e^0}{(1+e^0)^2}\varphi(x)$$

$$= -0.25\varphi(x)$$

$$\mathbf{w} \leftarrow \mathbf{w} + -0.25\varphi(x)$$

$w_{unigram\ \text{"Maizuru"}} = -0.25$

$w_{unigram\ \text{","}} = -0.5$

$w_{unigram\ \text{"in"}} = -0.25$

$w_{unigram\ \text{"Kyoto"}} = -0.25$

$w_{unigram\ \text{"A"}} = -0.25$

$w_{unigram\ \text{"site"}} = -0.25$

$w_{unigram\ \text{"located"}} = -0.25$

Example: second update

x = Shoken , monk born in Kyoto   y = 1

-0.5   -0.25   -0.25

$$\mathbf{w}\cdot\varphi(x) = -1 \qquad \frac{d}{d\mathbf{w}}P(y=1|x) = \frac{e^{1}}{(1+e^{1})^{2}}\varphi(x)$$

$$= 0.196\,\varphi(x)$$

$$\mathbf{w} \leftarrow \mathbf{w} + 0.196\,\varphi(x)$$

$w_{\text{unigram “Maizuru”}} = -0.25$
$w_{\text{unigram “,”}} = -0.304$
$w_{\text{unigram “in”}} = -0.054$
$w_{\text{unigram “Kyoto”}} = -0.054$

$w_{\text{unigram “A”}} = -0.25$
$w_{\text{unigram “site”}} = -0.25$
$w_{\text{unigram “located”}} = -0.25$

$w_{\text{unigram “Shoken”}} = 0.196$
$w_{\text{unigram “monk”}} = 0.196$
$w_{\text{unigram “born”}} = 0.196$

How to set the learning rate?

- Various strategies
  - decay over time

$$\alpha = \frac{1}{C + t}$$

Parameter

Number of samples

  - Use held-out test set, increase learning rate when likelihood increases

What you should know about linear models

- Standard supervised learning set-up for text classification
  - Difference between train vs. test data
  - How to evaluate
- 3 examples of linear classifiers
  - Naïve Bayes, Perceptron, Logistic Regression
    - How to make predictions, how to train, strengths and weaknesses
  - Learning as optimization: loss functions and their properties
  - Difference between generative vs. discriminative classifiers
- General machine learning concepts
  - Smoothing, overfitting, underfitting, regularization