

# CS 584 – MACHINE LEARNING

## TOPIC: LOGISTIC REGRESSION



**Mustafa Bilgic**



<http://www.cs.iit.edu/~mbilgic>



<https://twitter.com/bilgicm>

# LOGISTIC REGRESSION



# LOGISTIC REGRESSION

$$s = w_0 + \sum w_i x_i$$

- Learns  $P(Y|\mathbf{X})$  directly, without going through  $P(\mathbf{X}|Y)$  and  $P(Y)$
- Assumes  $P(Y|\mathbf{X})$  follows the logistic function

$$P(Y = \text{false} \mid X_1, X_2, \dots, X_n) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^n w_i X_i}} = \frac{1}{1 + e^s}$$

$$P(Y = \text{true} \mid X_1, X_2, \dots, X_n) = \frac{e^{w_0 + \sum_{i=1}^n w_i X_i}}{1 + e^{w_0 + \sum_{i=1}^n w_i X_i}} = \frac{e^s}{1 + e^s}$$

- Learning: estimate the weights  $w_0, w_1, \dots, w_n$



## LEARNING – PARAMETER ESTIMATION

- Maximize (conditional) log-likelihood

$$W \leftarrow \operatorname{argmax}_W \prod P(Y[d] \mid \mathbf{X}[d])$$

$$W \leftarrow \operatorname{argmax}_W \sum \ln P(Y[d] \mid \mathbf{X}[d])$$



# TAKE DERIVATIVE OF CLL WRT W

- See OneNote



# OPTIMIZATION

- No closed-form solution for  $W$
- One solution: gradient ascent
- Good news: log-likelihood for logistic regression is concave



# GRADIENT OPTIMIZATION



# MOTIVATION

CUL

- Maximize / minimize a function  $f(x)$
- Typical approach
  - Take gradient of  $f(x)$  wrt  $x$  and set it to 0
  - That is, solve  $\nabla f(x) = 0$
- What if there is no analytical solution to  $\nabla f(x) = 0$ ?
- One approach is gradient ascent (for maximization of  $f$ ) and gradient descent (for minimization of  $f$ )





## TWO SIMPLE EXAMPLES

1. Maximize  $f(x) = -2x^2 + 8x + 10$
2. Maximize  $f(x) = -x^5 - 2x^4 + 13x^3 + 14x^2 - 24x$



# GRADIENT ASCENT

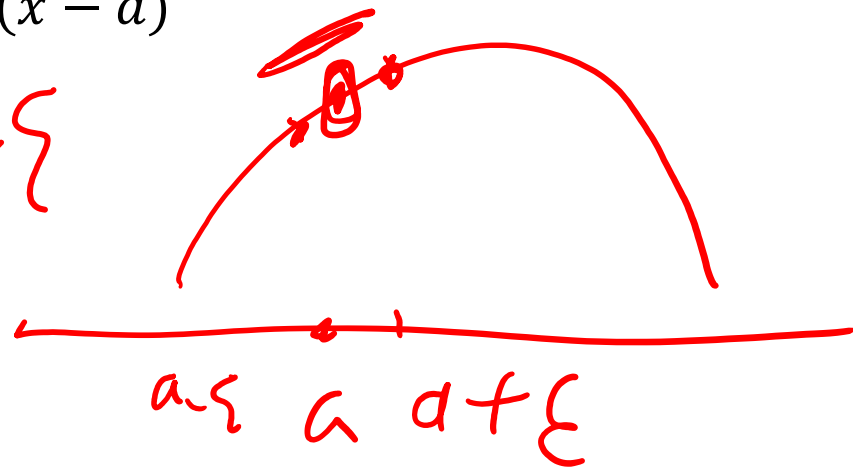
- Using the Taylor expansion, a function  $f(x)$  can be approximated at around  $a$  as
  - $f(x) \approx f(a) + \nabla f(a) * (x - a)$



# GRADIENT ASCENT

- Using the Taylor expansion, a function  $f(x)$  can be approximated at around  $a$  as
  - $f(x) \approx f(a) + \nabla f(a) * (x - a)$

$$a + \xi \approx f(a) + f'(a) \cdot \xi$$



$$f(a + \xi) \approx f(a) + f'(a) \cdot \xi$$



# GRADIENT ASCENT

- Find maximum of  $f(x)$  where there is no closed form solution
- Start with some initial guess  $x_0$
- While change is not much
  - $x_{i+1} = x_i + \eta * \nabla f(x_i)$
- $\eta$  is called the learning rate and it is a user specified parameter



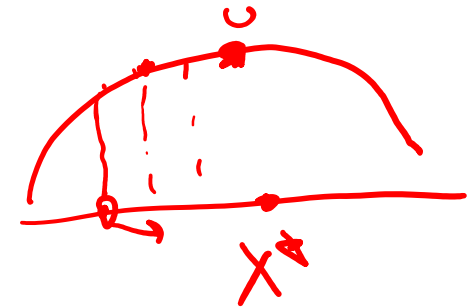
# GRADIENT ASCENT

- Find maximum of  $f(x)$  where there is no closed form solution
- Start with some initial guess  $x_0$
- While change is not much

- $x_{i+1} = x_i + \eta * \nabla f(x_i)$

- $\eta$  is called the learning rate and it is a user specified parameter

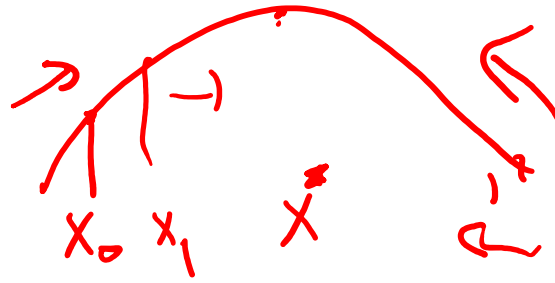
$$x_1 = x_0 + \eta \nabla f(x_0)$$
$$x_2 = x_1 + \eta \nabla f(x_1)$$



$$x^* = x^0 + \eta \cdot b$$



# GRADIENT ASCENT



- Find maximum of  $f(x)$  where there is no closed form solution
- Start with some initial guess  $x_0$
- While change is not much
  - $x_{i+1} = x_i + \eta * \nabla f(x_i)$
- $\eta$  is called the learning rate and it is a user specified parameter



## LET'S SEE A SIMPLE EXAMPLE

- Maximize  $f(x) = -2x^2 + 8x + 10$
- Note that  $\nabla f(x) = 0$  has a closed form solution for this example. We'll use this simple example just for illustration purposes
- See OneNote and Jupyter Notebook



# LOGISTIC REGRESSION GRADIENT EXAMPLE

- See OneNote





# REGULARIZATION



# REGULARIZATION

- Prefer smaller weights
  - Why?



# $L_2$ REGULARIZATION

- Objective function

- $W \leftarrow \operatorname{argmax}_W \left( \sum \ln P(Y[d] | \mathbf{X}[d]) - \frac{\lambda}{2} \|W\|^2 \right)$
- Trade-off between fit to the data vs model complexity

- Assuming  $n$  features

- $W \leftarrow \operatorname{argmax}_W \left( \sum \ln P(Y[d] | \mathbf{X}[d]) - \frac{\lambda}{2} \sum_{i=1}^n w_i^2 \right)$

- Take derivate of the objective function with respect to  $w_i$ .



# $L_2$ REGULARIZATION & BAYESIAN ESTIMATION

- Unregularized version corresponds to maximum likelihood estimate of the parameters
- Bayesian means we put a prior on what we do not know. Remember  $p(\theta)$ ,  $p(\theta|D)$ ,  $P(X|D)$ .
- In this case, we put a prior on  $w$ . That is, we have a prior distribution  $p(w)$ .
- $L_2$  regularization corresponds to
  - $p(w)$  is a Gaussian distribution with zero mean and variance related to  $1/\lambda$ , and
  - Taking the maximum of the posterior



# $L_1$ REGULARIZATION

- Instead of a quadratic penalty, absolute value is used
- Assuming  $n$  features
  - $W \leftarrow \underset{W}{\operatorname{argmax}} (\sum \ln P(Y[d] | X[d]) - \beta \sum_{i=1}^n |w_i|)$
- In the Bayesian case,  $p(w)$  is assumed to be not a Gaussian distribution but instead a Laplace distribution



## $L_2$ VS $L_1$

- $L_2$  forces the large weights to get closer to zero and places an emphasis on the large weights
  - Even though the weights get closer to zero, they are often not zero
- $L_1$  also penalizes large weights but the emphasis is not necessarily on the large weights
  - Some of the weights become zero
  - Leads to sparser representation



# CATEGORICAL FEATURES

- Logistic regression's parameters are feature weights
  - Hence, features need to have values that can be multiplied by a weight
- What if you have a binary feature?
  - Two choices: 0/1, or -1/+1.
- What if you have a categorical features that has more than two possible values, such as R, G, B?
  - Incorrect way: R=1, G=2, B=3. Why?
  - How should we handle these features?



# REFERENCES

- <http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf>
- [https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

