COMP755-Lect05

September 3, 2018

1 COMP 755

Plan for today

1. Review closed form linear regression, ridge regression
2. Introduce sigmoid function
3. Logistic regression
4. Geometric view of logistic regression -- separating hyperplanes
5. Regularization in logistic regression
6. HW1 overview
7. Bayesian interpretation of penalties

2 Last time -- closed form solution for linear regression

$$\begin{bmatrix} \beta_0^{\text{MLE}} \\ \beta^{\text{MLE}} \end{bmatrix} = (X_1^T X_1)^{-1} X_1^T \mathbf{y}$$

$$(\sigma^{\text{MLE}})^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{y} - \beta_0^{\text{MLE}} - X\beta^{\text{MLE}} \right)^2$$

where

$$X_1 = \begin{bmatrix} \mathbf{1}_p \; X \end{bmatrix}$$

and $\mathbf{1}_r$ denotes r long column-vector of 1s.

3 Last time -- regularization

Ill-posed problems have many solutions.

One way to break the ties between different solutions is to add regularization.

Ridge regression is adds regularization to the log-likelihood:

$$\log \mathcal{L}(\beta, \beta_0, \sigma^2 | \mathbf{y}, X) - \lambda \left\| \beta \right\|^2 = -\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \left\| \mathbf{y} - \beta_0 - X\beta \right\|^2 - \lambda \left\| \beta \right\|^2 .$$

Using this regularization spreads the weights between correlated predictors.

Setting $\lambda = 0$ recovers the linear regression log-likelihood.

4 Last time -- gradients for ridge penalized optimization

Gradients of linear regression log-likelihood:

$$\frac{\partial}{\partial \beta_0} = \frac{1}{\sigma^2}(\mathbf{y} - \beta_0 \mathbf{1} - X\beta)^T \mathbf{1}$$
$$\nabla_\beta \log \mathcal{L}(\beta | \mathbf{y}, \mathbf{x}) = \frac{1}{\sigma^2}(\mathbf{y} - \beta_0 \mathbf{1} - X\beta)^T X$$

Gradients of ridge regression log-likelihood:

$$\frac{\partial}{\partial \beta_0} = \frac{1}{\sigma^2}(\mathbf{y} - \beta_0 \mathbf{1} - X\beta)^T \mathbf{1}$$
$$\nabla_\beta \log \mathcal{L}(\beta | \mathbf{y}, \mathbf{x}) = \frac{1}{\sigma^2}(\mathbf{y} - \beta_0 \mathbf{1} - X\beta)^T X - \lambda\beta$$

5 Last time -- closed form solution for ridge regression

$$\begin{bmatrix} \beta_0^{\text{MLE}} \\ \beta^{\text{MLE}} \end{bmatrix} = \left(X_1^T X_1 + \text{diag}\left(\begin{bmatrix} 0 \\ \lambda \mathbf{1}_p \end{bmatrix} \right) \right)^{-1} X_1^T \mathbf{y}$$
$$(\sigma^{\text{MLE}})^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{y} - \beta_0^{\text{MLE}} - X\beta^{\text{MLE}} \right)^2$$

where

$$X_1 = \left[\mathbf{1}_p \; X \right],$$

$\mathbf{1}_r$ denotes r long column-vector of 1s, and diag $(\mathbf{v})$ is a diagonal matrix whose diagonal is populated by entries of vector $\mathbf{v}$.

The only difference between the closed form of linear regression and ridge regression is the addition of the term shown in red.

6 Feature preprocessing -- Centering

Center features

$$\mu_i = \frac{1}{N} \sum_{k=1}^{N} x_{i,k}$$
$$x_{i,j} = x_{i,j} - \mu_i$$

This makes each feature's mean equal to 0. Compute the mean first, then subtract it!

7 Feature preprocessing -- Standardizing

Standardize centered features

$$\sigma_i = \sqrt{\frac{1}{N-1} \sum_j x_{i,j}^2}$$
$$x_{i,j} = \frac{x_{i,j}}{\sigma_i}$$

This makes each feature's scale the same. Compute the standard deviation, then divide by it.

Note that standardized features are first centered and then divided by their standard deviation.

8 Feature preprocessing -- Normalizing

Alternatively, **normalize** centered features

$$r_i = \sqrt{\sum_j x_{i,j}^2}$$
$$x_{i,j} = \frac{x_{i,j}}{r_i}$$

This makes each feature's scale the same regardless of the data set size.

Note that normalized features are first centered and then divided by their norm.

9 Feature preprocessing

Benefits: 1. Centering 1. β_0 is equal to the mean of the target variable 2. feature weights β now tell us how much does feature's departure from mean affect the target variable 2. Standardization 1. all the features are on the same scale and their effects comparable 2. interpretation is easier: βs tell us how much departure by single standard deviation affects the target variable 3. Normalization 1. scale of features is the same, regardles of the size of the dataset 2. hence weights learend on different sized datasets can be compared 3. however, their combination might be problematic -- certainly we don't trust weights learned on few samples

10 Classification

We used linear regression to fit a predictive model of continuous variables:

$$y|\mathbf{x} \sim \mathcal{N}\left(\beta_0 + \mathbf{x}^T\beta, \sigma^2\right)$$

Gaussian distribution is not well suited for modelling discrete variables

Q: Why?

11 Classification -- Bernoulli view

We can model a target variable $y \in \{0, 1\}$ using Bernouli

$$p(y = 1|\theta) = \theta$$

We note that θ has to be in range $[0, 1]$.

We cannot directly take weighted combination of features to obtain θ.

We need a way to map $\mathbf{x}^T\beta \in \mathbb{R}$ to range $[0, 1]$. # Sigmoid function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

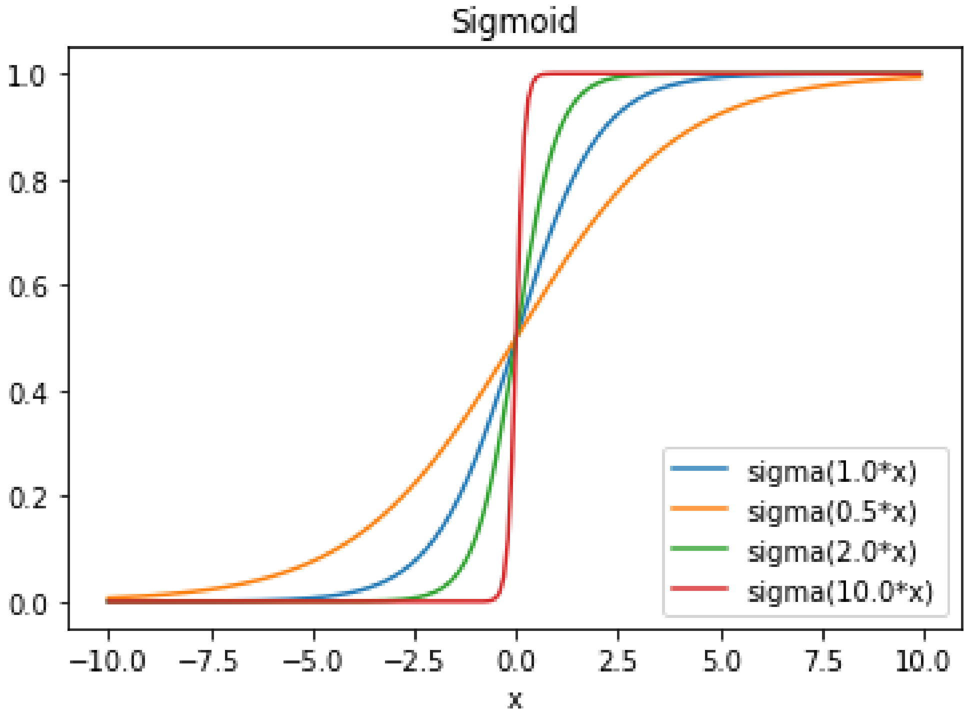
Q: Argue that this function maps real line to range $[0, 1]$?

Q: When does it achieve 0.5?

Q: Where does it achieve 0? Where does it achieve 1?

```
In [1]: import numpy
        import matplotlib.pyplot as plt
        %matplotlib inline
        x = numpy.arange(-10,10,0.1)
        scales = [1.0,0.5,2.0,10.0]
        labels = []
        for s in scales:
            plt.plot(x,1.0/(1.0 + numpy.exp(-s*x)))
            labels.append('sigma(' + str(s) +'*x)')
        plt.xlabel('x')
        plt.title('Sigmoid')
        plt.legend(labels,loc=4)
```

```
Out[1]: <matplotlib.legend.Legend at 0x10685e080>
```



12 Some useful equalities involving sigmoid

Definition:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Recognize the alternative way to write it:

$$\sigma(z) = \frac{\exp z}{1 + \exp z}$$

Complement is just flip of the sign in the argument

$$\sigma(-z) = 1 - \sigma(z)$$

Log ratio of probability (log odds)

$$\log \frac{\sigma(z)}{\sigma(-z)} = z$$

13 Using sigmoid to parameterize Bernoulli

$$p(y = 1|\theta) = \theta$$

Sigmoid "squashes" the whole real line into range $[0, 1]$.
Hence we can map weighted features into a parameter θ

$$\theta = \sigma(\beta_0 + \mathbf{x}^T \beta)$$

and use that θ in our Bernoulli

$$p(y = 1 \big| \theta = \sigma(\beta_0 + \mathbf{x}^T \beta)) = \sigma(\beta_0 + \mathbf{x}^T \beta)$$

14 Logistic regression

In logistic regression we model a binary variable $y \in \{-1, +1\}$

$$p(y = +1 \big| \mathbf{x}, \beta_0, \beta) = \sigma\left(+(\beta_0 + \mathbf{x}^T \beta)\right)$$
$$p(y = -1 \big| \mathbf{x}, \beta_0, \beta) = 1 - \sigma\left(-(\beta_0 + \mathbf{x}^T \beta)\right) = \sigma\left(-(\beta_0 + \mathbf{x}^T \beta)\right)$$

This is equivalent to

$$p(y \big| \mathbf{x}, \beta_0, \beta) = \sigma\left(y(\beta_0 + \mathbf{x}^T \beta)\right) = \frac{1}{1 + \exp\{-y(\beta_0 + \mathbf{x}^T \beta)\}}$$

Q: Does above formula work for $y \in \{0, 1\}$?

15 Logistic regression -- log-likelihood

Probability of a single sample is:

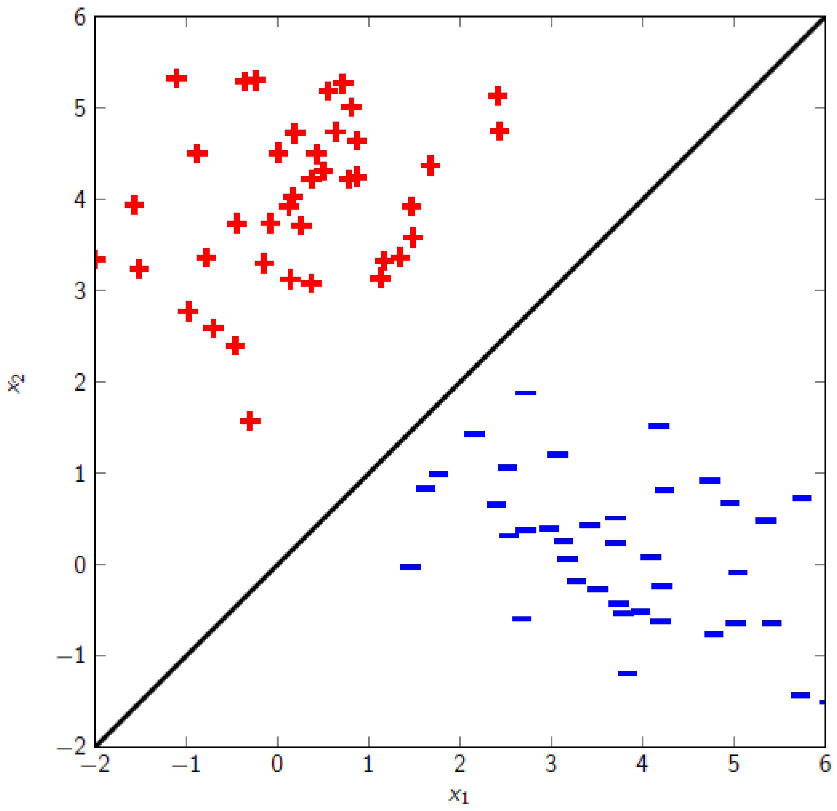
$$p(y \big| \mathbf{x}, \beta_0, \beta) = \frac{1}{1 + \exp\{-y(\beta_0 + \mathbf{x}^T \beta)\}}$$

Likelihood function is:

$$\mathcal{L}(\beta_0, \beta \big| \mathbf{y}, \mathbf{x}) = \prod_i \frac{1}{1 + \exp\{-y_i(\beta_0 + \mathbf{x}_i^T \beta)\}}$$

Log-likelihood function is:

$$\log \mathcal{L}(\beta_0, \beta \big| \mathbf{y}, \mathbf{x}) = -\sum_i \log\left\{1 + \exp\left\{-y_i(\beta_0 + \mathbf{x}_i^T \beta)\right\}\right\}$$



Separating Hyperplane

16 Interpreting weights in logistic regression

Q: When is $\frac{1}{1+\exp\{-(\beta_0+\mathbf{x}^T\beta)\}} = 0.5$?
Q: What does this set of $\mathbf{x}$ look like?

17 Separating hyperplanes

18 Convergence of optimization of logistic regression

If there exists a hyperplane that splits the data perfectly, then log-likelihood can be indefinitely impoved driving weights to infinity.

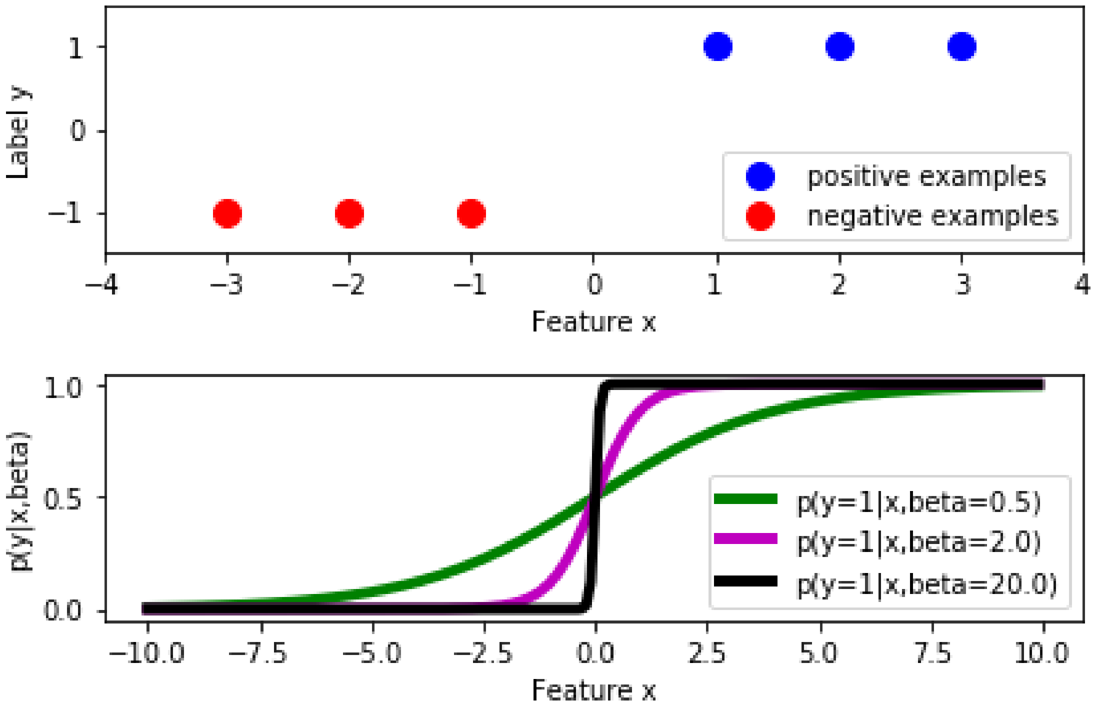
```
In [2]: import numpy
        import matplotlib.pyplot as plt

        def plot_data(xs,ys):
            pos = [i for i in range(len(ys)) if ys[i]>0]
            neg = [i for i in range(len(ys)) if ys[i]<0]
            plt.plot(xs[pos],ys[pos],'b.',
                     markersize=20.,
                     label='positive examples')
```

```
    plt.plot(xs[neg],ys[neg],'r.',
             markersize=20.,
             label='negative examples')

def plot_sigmoid(s,color):
    x = numpy.arange(-10.0,10.0,0.1)
    plt.plot(x,1.0/(1.0 + numpy.exp(-s*x)),color,
             linewidth = 4.,
             label='p(y=1|x,beta='+str(s)+')')
ys = numpy.asarray([-1.0,-1.0,-1.0,1.0,1.0,1.0])
xs = numpy.asarray([-3.0,-2.0,-1.0,1.0,2.0,3.0])
plt.subplot(2,1,1)
plot_data(xs,ys)
plt.ylim([-1.5,1.5])
plt.xlim([-4.0,4.0])
plt.xlabel('Feature x')
plt.ylabel('Label y')
plt.legend(loc=4)

plt.subplot(2,1,2)
plot_sigmoid(0.5,'g')
plot_sigmoid(2.0,'m')
plot_sigmoid(20.0,'k')
plt.legend(loc=4)
plt.xlabel('Feature x')
plt.ylabel('p(y|x,beta)')
plt.tight_layout()
```



19 Ridge penalty and logistic regression

Adding ridge penalty to the logistic regression achieves 1. Shrinkage of weights -- weights no longer explode in separable case 2. Even splitting between correlated weights

20 Bayesian view of penalties

We have seen two examples of supervised models 1. Linear regression, $p(y|\mathbf{x}, \beta)$ where $y \in \mathbb{R}$ 2. Logistic regression, $p(y|\mathbf{x}, \beta)$ where $y \in \{-1, +1\}$

We then ut log-likelihoods

$$\log \mathcal{L}(\beta|\mathbf{y}, X) = \sum_i \log p(y_i|\mathbf{x}_i, \beta)$$

and observed that we can add penalties to log-likelihoods

$$\log \mathcal{L}(\beta|\mathbf{y}, X) + \lambda f(\beta)$$

in order to deal with ill-posedness of the problems.

21 Bayesian view of penalties

Given a likelihood

$$p(\text{Data}|\theta)$$

Bayesian view of models treats each parameter θ as just another random variable.

This random variable has a distribution called **prior** distribution

$$p(\theta)$$

Using Bayes rule we can also compute

$$\overbrace{p(\theta|\text{Data})}^{\text{posterior}} = \frac{\overbrace{p(\text{Data}|\theta)}^{\text{likelihood}}\overbrace{p(\theta)}^{\text{prior}}}{p(\text{Data})}$$

called **posterior** distribution.

Prior encodes our beliefs **before** seing the data.

Posterior reflects our updated beliefs **after** seeing the data.

22 Bayesian view of penalties -- Gaussian prior and linear regression

For example we can assume a Gaussian **prior** on β_i

$$\beta_i \sim \mathcal{N}\left(0, \frac{1}{\lambda}\right), \qquad i > 0$$

$$y \sim \mathcal{N}\left(\beta_0 + \mathbf{x}^T\beta, \sigma^2\right)$$

Then posterior probability of the parameter β_i:

$$p(\beta|\mathbf{y},\mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x},\beta)p(\beta)}{p(\mathbf{y}|\mathbf{x})}$$

23 Bayesian view of penalties

We can now try to find Maximum-A-Posteriori (MAP) estimate of θ

$$\arg\max_\beta p(\beta|\mathbf{y},\mathbf{x}) = \arg\max_\beta \log p(\mathbf{y}|\mathbf{x},\beta) + \log p(\beta)$$

and this is equivalent to

$$\arg\max_\beta p(\beta|\mathbf{y},\mathbf{x}) = \arg\max_\beta -\sum_{i=1}^{N}\frac{1}{2\sigma^2}(y_i - \beta_0 - \mathbf{x}_i^T\beta) - \sum_{j=1}^{p}\frac{\lambda}{2}\beta_j^2 + \text{const}$$

Solving ridge regression is equivalent to finding Maximum-A-Posteriori estimate in Bayesian linear regression with Gaussian prior on weights.

24 Bayesian view of penalties

Penalties are log-probabilities of the parameters.

Maximiziation of penalized log-likelihood is equivalent to maximizing posterior probability of the parameters.

Further, uncertainty about parameters can be quantified once we have a distribution

$$p(\theta|\text{Data})$$

instead of just maximum likelihood estimate.

25 HW1 overview -- submission

Submit your .ipynb file to **jsilva@cs.unc.edu** Give us your name, PID. Working in pairs is fine. List the name of your co-worker.

```
To: jsilva@cs.unc.edu
From: Super Student
Subject: HW1 submission

First Name: Super
Last Name: Student
PID: 11111111

Colaborated with:
First Name: Another
Last Name: Student
```

26 HW1 overview -- submission

The HW1 will be up at midgnight tonight.

You will have until 11:59PM 9/16 to submit.

TAs office hours will be on Wednesday

You can submit as many times as you want.

The last one will be graded.

We will look at the submitted files periodically and send comments when appropriate.

If you are stuck, send an e-mail to both jsilva@cs and poirson@cs

We will **not** debug your code. We might suggest things to think about. **Hints might help or hinder.**

27 HW1 a brief look

28 Today

1. Review regularized/penalized linear regression
2. Introduce sigmoid and logistic regression
3. Geometric interpretation of logistic regression
4. Bayesian view of regularized methods