DAM lecture 11:

Logistic Regression

14.03.2016

Aasa Feragen
aasa@diku.dk

After today's lecture you should

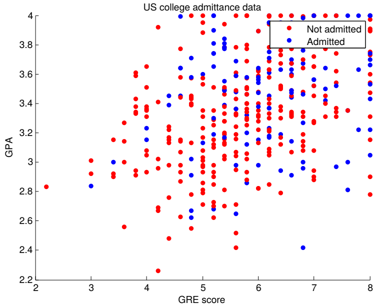
- be able to explain the model defined in logistic regression
- understand how gradient descent solves logistic regression
- be able to implement a gradient descent solver for logistic regression

Change in Assignment 3

- You will be allowed to use built-in solvers for logistic regression in Assignment 3
- In the exam assignment, you will be given a choice of using the built-in solver or implementing your own – using the built-in solver will be worth fewer points.

Case: Predicting college admittance

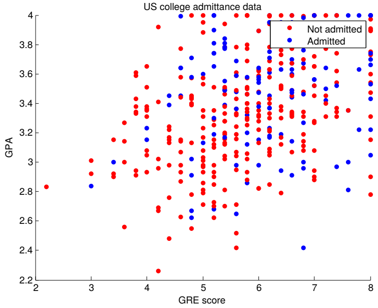
- 400 American collage applicants
- Features: High school GPA and results of the pre-college GRE test (Graduate Record Examinations)
- Labels: Admittance (1) or not (0)
- **Task:** Predict a probability of being admitted based on GPA and GRE score[1]



[1] Data from: http://www.ats.ucla.edu/stat/r/dae/logit.htm

Case: Predicting college admittance

- Let's take a look at the data



The meaning of $\boldsymbol{w}^T\boldsymbol{x}$

Assume that your data lives in $\mathbb{R}^d$, that $\boldsymbol{x} \in \mathbb{R}^d$ is a variable, and that $\boldsymbol{w} \in \mathbb{R}^d$ is some vector.
What does the equation

$$\boldsymbol{w}^T\boldsymbol{x}$$

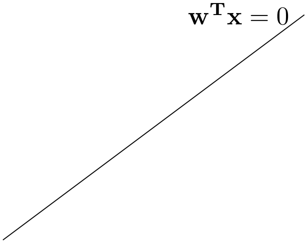
describe?

The meaning of $\boldsymbol{w}^T\boldsymbol{x}$

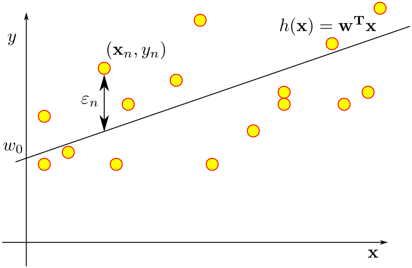
Assume that your data lives in $\mathbb{R}^d$, that $\boldsymbol{x} \in \mathbb{R}^d$ is a variable, and that $\boldsymbol{w} \in \mathbb{R}^d$ is some vector.
What does the equation

$$\boldsymbol{w}^T\boldsymbol{x}$$

describe?

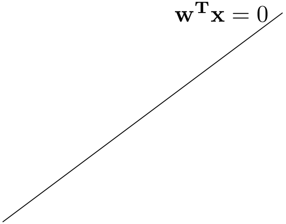


Reminder: Linear regression



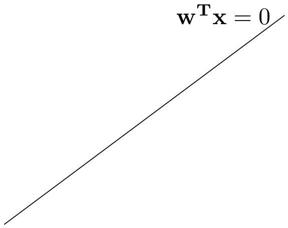
Logistic regression

- What happens when your point $\boldsymbol{x}$ moves away from the boundary defined by $\boldsymbol{w}^T\boldsymbol{x}$?



Logistic regression

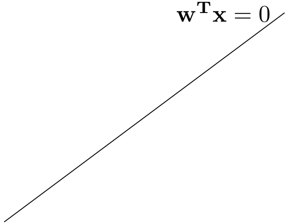
- What happens when your point $\boldsymbol{x}$ moves away from the boundary defined by $\boldsymbol{w}^T\boldsymbol{x}$?



- The value of $\boldsymbol{w}^T\boldsymbol{x}$ gets either larger or smaller

Logistic regression

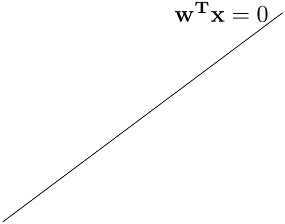
- What happens when your point $\boldsymbol{x}$ moves away from the boundary defined by $\boldsymbol{w}^T\boldsymbol{x}$?



- The value of $\boldsymbol{w}^T\boldsymbol{x}$ gets either larger or smaller
- *Logistic regression* returns a probability $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$ of belonging to either the red or blue class, such that either $|h(\boldsymbol{x})|$ approaches either 0 or 1 as $|\boldsymbol{w}^T\boldsymbol{x}|$ becomes large.

Logistic regression

- What happens when your point $\boldsymbol{x}$ moves away from the boundary defined by $\boldsymbol{w}^T\boldsymbol{x}$?

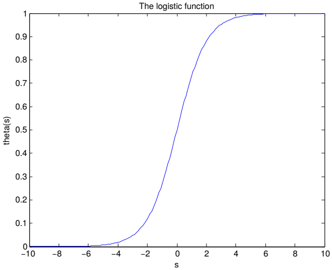


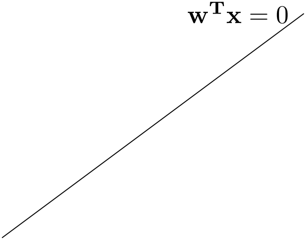
- The value of $\boldsymbol{w}^T\boldsymbol{x}$ gets either larger or smaller
- *Logistic regression* returns a probability $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$ of belonging to either the red or blue class, such that either $|h(\boldsymbol{x})|$ approaches either 0 or 1 as $|\boldsymbol{w}^T\boldsymbol{x}|$ becomes large.
- How could you obtain that?

How do we obtain that?

Logistic regression makes use of the *logistic function*

$$\theta(s) = \frac{e^s}{1 + e^s}$$





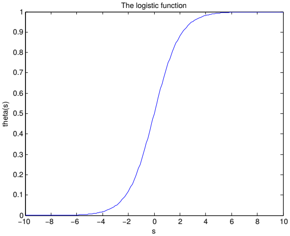
WARNING

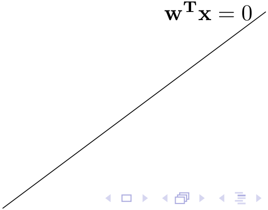
In the following, we shall use classification labels ± 1. The textbook uses classification labels $0/1$. This makes the formulas incompatible; do not mix them.

Logistic regression

- Our goal is to learn a target probability of belonging to each of the two classes,

$$P(y|\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{for } y = +1, \\ 1 - h(\boldsymbol{x}) & \text{for } y = -1. \end{cases}$$



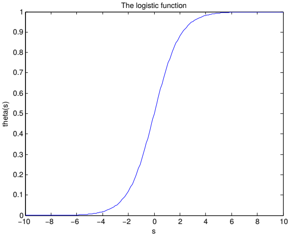


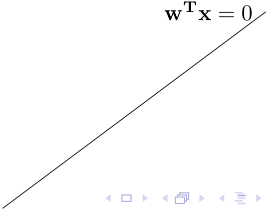
Logistic regression

- Our goal is to learn a target probability of belonging to each of the two classes,

$$P(y|\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{for } y = +1, \\ 1 - h(\boldsymbol{x}) & \text{for } y = -1. \end{cases}$$

- We define $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T \boldsymbol{x})$. **Determined completely by $\boldsymbol{w}$.**





Logistic regression

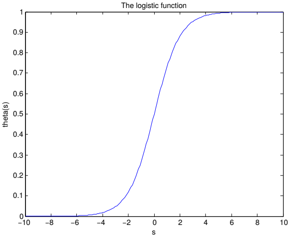
- Our goal is to learn a target probability of belonging to each of the two classes,

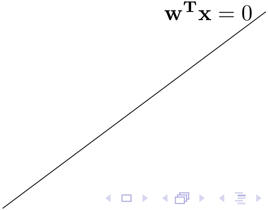
$$P(y|\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{for } y = +1, \\ 1 - h(\boldsymbol{x}) & \text{for } y = -1. \end{cases}$$

- We define $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T \boldsymbol{x})$. **Determined completely by $\boldsymbol{w}$.**
- Substituting $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T \boldsymbol{x})$, since $1 - \theta(s) = \theta(-s)$, we get

$$P(y|\boldsymbol{x}) = \theta(y\boldsymbol{w}^T \boldsymbol{x}).$$

- $P(y|\boldsymbol{x})$ is called the *likelihood* of observing the output class y given the input $\boldsymbol{x}$.





Logistic regression

- Assume independent data points $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$.
- Probability of observing all these y_n, given the inputs $\boldsymbol{x}_n$:

$$\prod_{n=1}^{N} P(y_n | \boldsymbol{x}_n).$$

Logistic regression

- Assume independent data points $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$.
- Probability of observing all these y_n, given the inputs $\boldsymbol{x}_n$:

$$\prod_{n=1}^{N} P(y_n | \boldsymbol{x}_n).$$

- A good choice of model parameters $\boldsymbol{w}$ maximizes the probability of observing the data, that is

$$\mathrm{argmax}_{\boldsymbol{w}} \prod_{n=1}^{N} P(y_n | \boldsymbol{x}_n)$$

- A *maximum likelihood* model!

Logistic regression

- Maximizing $\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)$ is equivalent to minimizing

$$-\frac{1}{N}\ln\left(\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)\right) = \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{P(y_n|\mathbf{x}_n)}\right).$$

- Since

$$P(y_n|\mathbf{x}_n) = \theta(y_n\mathbf{w}^T\mathbf{x}_n), \text{ and } \theta(s) = \frac{e^s}{1+e^s}$$

we end up minimizing

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{\theta(y_n\mathbf{w}^T\mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N}\ln\left(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}\right).$$

- Pointwise log likelihood: $\ln(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n})$

Logistic regression

- Maximizing $\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)$ is equivalent to minimizing

$$-\frac{1}{N}\ln\left(\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)\right) = \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{P(y_n|\mathbf{x}_n)}\right).$$

- Since

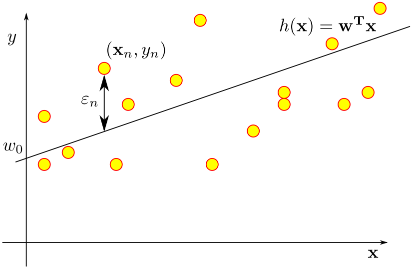
$$P(y_n|\mathbf{x}_n) = \theta(y_n\mathbf{w}^T\mathbf{x}_n), \text{ and } \theta(s) = \frac{e^s}{1+e^s}$$

we end up minimizing

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{\theta(y_n\mathbf{w}^T\mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N}\ln\left(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}\right).$$

- Pointwise log likelihood: $\ln(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n})$
- How can I minimize E_{in}?

Reminder: Solving linear regression



- Minimize in-sample error:

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(\boldsymbol{x}_n) - y_n)^2$$

- Compute gradient $\nabla_{\boldsymbol{w}} E_{in}(\boldsymbol{w})$
- Solve $\nabla_{\boldsymbol{w}} E_{in}(\boldsymbol{w}) = 0$
- Obtain an analytic solution:

$$\boldsymbol{w} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Logistic regression

- Need to minimize

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right).$$

Logistic regression

- Need to minimize

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right).$$

- Can compute:

$$\nabla_{\mathbf{w}} E_{in} = \frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^T \mathbf{x}_n)$$

Logistic regression

- Need to minimize

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right).$$

- Can compute:

$$\nabla_{\mathbf{w}} E_{in} = \frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^T \mathbf{x}_n)$$

- Unfortunately, setting $\nabla_{\mathbf{w}} E_{in} = 0$ does not make us much wiser!!

Logistic regression

- Need to minimize

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \boldsymbol{w}^T \boldsymbol{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \boldsymbol{w}^T \boldsymbol{x}_n}\right).$$

- Can compute:

$$\nabla_{\boldsymbol{w}} E_{in} = \frac{1}{N}\sum_{n=1}^{N} -y_n \boldsymbol{x}_n \theta(-y_n \boldsymbol{w}^T \boldsymbol{x}_n)$$

- Unfortunately, setting $\nabla_{\boldsymbol{w}} E_{in} = 0$ does not make us much wiser!!
- Numerical optimization (gradient descent)

Let's start implementing the function and its gradient!

Fill in the functions

- `E = logistic_insample(X,y,w)`
- `g = logistic_gradient(X, y, w)`

where

- `X` is an N by (d+1) data matrix with a column of ones appended
- `w` is a (d+1)-dimensional weight vector
- `y` is a N-dimensional output vector

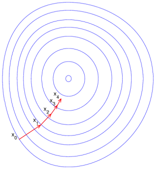
Remember:

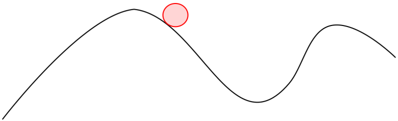
$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)}\right) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)$$

$$\nabla_{\mathbf{w}} E_{in} = \frac{1}{N} = \sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^T \mathbf{x}_n)$$

Idea behind gradient descent

Walk downhill until you hit bottom



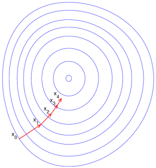


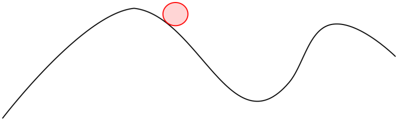
Issues that need your attention:

- Where do I start?
- How long steps?
- When have I hit bottom?

Idea behind gradient descent

Walk downhill until you hit bottom



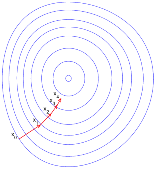


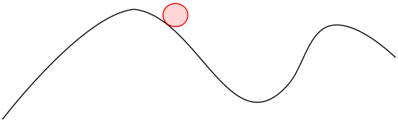
Issues that need your attention:

- Where do I start? Common: Initialize with random sample from normal distribution
- How long steps?
- When have I hit bottom?

Idea behind gradient descent

Walk downhill until you hit bottom



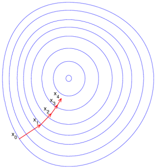


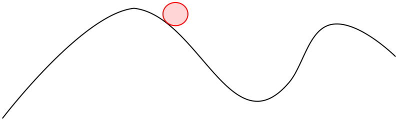
Issues that need your attention:

- Where do I start? Common: Initialize with random sample from normal distribution
- How long steps? Common: $\eta_t = \eta \|\nabla E_{in}\|$; η called *learning rate*
- When have I hit bottom?

Idea behind gradient descent

Walk downhill until you hit bottom





Issues that need your attention:

- Where do I start? Common: Initialize with random sample from normal distribution
- How long steps? Common: $\eta_t = \eta \|\nabla E_{in}\|$; η called *learning rate*
- When have I hit bottom? Threshold on $\sharp$ steps or step size

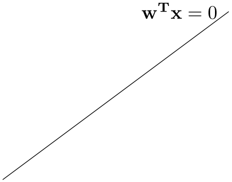
Let's implement gradient descent for logistic regression!

Goal: Weights $\boldsymbol{w}$ that define optimal classifier minimizing $E_{in}(\boldsymbol{w})$.

1. Initialize weights
2. For $t = 0, 1, 2, \ldots$
    3. Compute the gradient

$$\boldsymbol{g}_t = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n \boldsymbol{x}_n}{1 + e^{y_n \boldsymbol{w}^T \boldsymbol{x}_n}},$$

    4. Set direction to step: $\boldsymbol{v}_t = -\boldsymbol{g}_t$
    5. Update the weights: $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \eta \boldsymbol{v}_t$
    6. Iterate until the next step until stopping
7. Return the final weights $\boldsymbol{w}$.



Once you have the optimal model

Let's visualize the output!

- Define function `log_pred` which, given $\boldsymbol{w}$ and $\boldsymbol{x}$, returns $P(y|\boldsymbol{x})$.
- Threshold into classes

Next time:

- Lecture with Christian on decision trees