Machine Learning
Assignment 2.1

Nikolaj Dybdahl Rathcke (rfq695)

December 3, 2015

1 Logistic regression

1.1 Cross-entropy error measure

(a)
The maximum likelihood selects a hypthesis h which maximizes a probability. We can rewrite this so it minimizes it instead:

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

We know y only takes values in $\{-1, 1\}$ and we know that $P(y_n|x_n) = h(x_n)$ when $y_n = 1$ and $1 - h(x_n)$ in the other case. Since we want to minimize this quantity, we can treat it as an error measure, so we can write $E_{in}(w)$ as:

$$\begin{aligned}
E_{in}(w) &= \frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{P(y_n|x_n)}\right) \\
&= \frac{1}{N}\sum_{n=1}^{N}[\![y_n = 1]\!]\ln\left(\frac{1}{h(x_n)}\right) + [\![y_n = -1]\!]\ln\left(\frac{1}{1-h(x_n)}\right) \\
&= \sum_{n=1}^{N}[\![y_n = 1]\!]\ln\left(\frac{1}{h(x_n)}\right) + [\![y_n = -1]\!]\ln\left(\frac{1}{1-h(x_n)}\right)
\end{aligned}$$

We can do the last step since we are minimizing and dividing the value by N will not change the w we find, and this is what we wanted to show.

(b)
Remember $\theta(s) = \frac{e^s}{1+e^s}$, so we have the following limits:

$$\begin{aligned}
\lim_{s\to\infty} &= 1 \\
\lim_{s\to-\infty} &= 0 \\
\lim_{s\to 0} &= 1/2 && \text{(From both sides)}
\end{aligned}$$

In equation (3.9), the term $e^{-y_n w^T x_n}$ is deciding for how small the in-sample error is. Thus we want $y_n w^T x_n$ to be as large as possible so the entire term is as small as possible. This means we want $w^T x_n$ to be the same sign as y_n.
For the equation we found in (a), if $y_n = 1$, we want $h(x) = \theta(w^T x_n)$ to be large and positive as it contributes less to the sum than if it is small (see limits). When $y_n = -1$, we want $1 - h(x)$ to be large, which means we want $h(x)$ to be as small as possible (large and negative).
So for both equations we want them to be to be the same sign and they grow in same fashion, the task of minimizing are equivalent.

1.2 Logistic regression loss gradient

Equation (3.9) in the textbook says:

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N} \ln(1 + e^{-y_n w^T x_n})$$

We can take the derivative, of this, by first applying the chain rule and get:

$$\begin{aligned}
\nabla E_{in}(w) &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{1 + e^{-y_n w^T x_n}}(-y_n x_n e^{-y_n w^T x_n}) \\
&= -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n x_n e^{-y_n w^T x_n}}{1 + e^{-y_n w^T x_n}} \\
&= -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n x_n}{1 + e^{y_n w^T x_n}} \\
&= \frac{1}{N}\sum_{n=1}^{N} -y_n x_n \theta(-y_n w^T x_n)
\end{aligned}$$

And we have showed the derivative.
When y_n and $w^T x_n$ have different signs (they are misclassified)), the product $s = -y_n w^T x_n$ becomes positive. Our limits from (1.1) tells us that the value $\theta(s)$ must lie in $(0.5, 1)$. When y_n and $w^T x_n$ have the same sign (classified correctly), then $s = -y_n w^T x_n$ is negative and our limits tell us they must lie in $(0, 0.5)$. Thus, misclassified examples contribute more to the gradient.

1.3 Logistic regression implementation

The logistic regression algorithm (Example 3.3 in the textbook) has been implemented in `src/logistic_regr.py`. Running the file will report the parameters we found as well as the error on the training and test set:

```
Parameters for the model:
[-0.00813814 -0.02809434]

Error on training set with found parameters:
0.451612903226

Error on test set with found parameters:
0.384615384615
```

Where a successfull prediction is when the probability if more than 0.5 for the right classication and if not, it is a misprediction.

2 Linear least squares

2.1

To implement this, since we expect the canonball to fly in a parabola like fashion, we use $d = 2$. The method `leastSquares` in `least_squares.py` implements this by constructing the matrix:

$$A = \begin{bmatrix} 0 & 0 & 1 \\ x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_8^2 & x_8 & 1 \end{bmatrix}$$

The first line is added because we also know the position of the cannon (and the height is 0 in that point). We can find the parameters by calculating

$$w = (A^T A)^{-1} A^T y$$

where y is our measured height and x is the horizontal distance from the cannon. This yields the parameters:

```
Parameters from least squares with d=2:
[-0.97880952  9.98280952 -0.466     ]
```

which is equal to the formula $f(x) = -0.9788x^2 + 9.9828x - 0.466$.

2.2

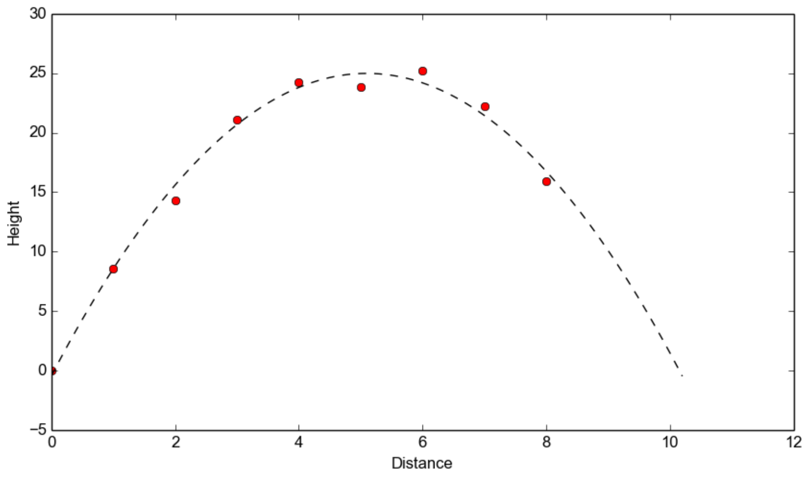
To estimate how far from the cannon Baron Münchhausen will fall, we simply solve for x when the $f(x)$ is equal to zero. This gives the following results:

$$x \approx 0.0469 \text{ or } x = 10.152$$

since the first solution is the cannon's position, the second solution is where Baron Münchhausen will fall, so at distance 10.152.

2.3

The following figure shows the comparison between the measured points and the result from using the least squares method:



Where the red points are the measured points and the dotted line is our estimation of a function. The estimation has very little noise.