Pattern Classification And Machine Learning - EPFL - Fall 2015
Emtiyaz Khan, Timur Bagautdinov, Carlos Becker, Ilija Bogunovic & Ksenia Konyushkova

5. Logistic regression

5.1 Goals

The goal of this exercise is to

- Do classification using linear regression.
- Implement and debug logistic regression using gradient descent.
- Compare it to linear regression.
- Implement Newton's method for logistic regression.

5.2 Classification using linear regression

We will try to use linear regression to do classification. Although this is not a good idea in general (as discussed in the class), it will work for simple data. We will use the height-weight data from the first exercise. For better visualization, we will randomly sub-select the data. The following code randomly sub-selects 200 data points.

```
% Load data
load('height_weight_gender.mat');
height = height * 0.025;
weight = weight * 0.454;
y = gender;
X = [height(:)  weight(:)];
% randomly permute data
N = length(y);
idx = randperm(N);
y = y(idx);
X = X(idx,:);
% subsample
y = y(1:200);
X = X(1:200,:);
```

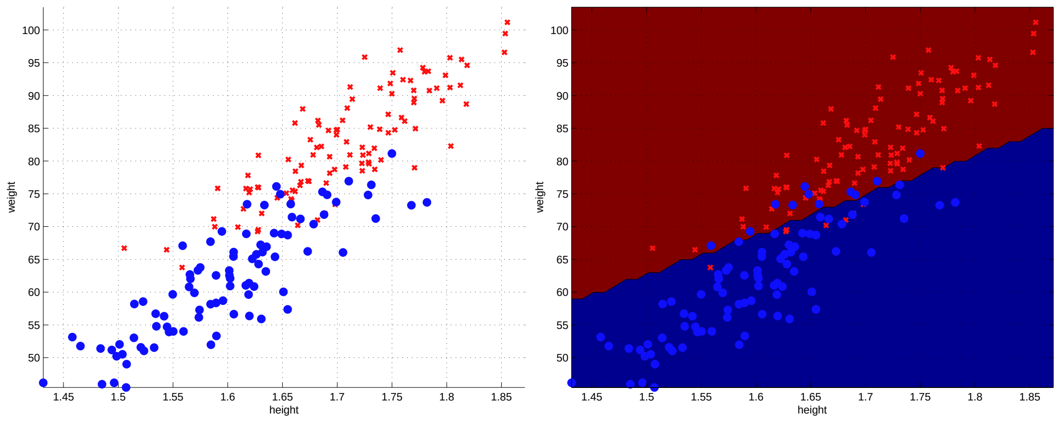
Exercise 5.1 Classification using linear regression.

- Plot the data as shown in Fig. 5.1 (a).
- Use least squares to compute a β.
- Plot predictions to get Fig. 5.1 (b). See the code below for help.

■

Given a matrix X of size $N \times 2$, you can plot decision boundary using the following code.

```
% create a 2-D meshgrid of values of heights and weights
h = [min(X(:,1)):.01:max(X(:,1))];
w = [min(X(:,2)):1:max(X(:,2))];
[hx, wx] = meshgrid(h,w);
% predict for each pair, i.e. create tX for each [hx,wx]
% and then predict the value. After that you should
% reshape `pred` so that you can use `contourf`.
% For this you need to understand how `meshgrid` works.
```



(a) Height-weight data for males (x in red) and females (o in blue)

(b) Decision boundary using least squares

Figure 5.1: Classification using linear regression.

```
pred = ....

% plot the decision surface
contourf(hx, wx, pred, 1);
% plot indiviual data points
hold on
myBlue = [0.06 0.06 1];
myRed = [1 0.06 0.06];
plot(X(males,1), X(males,2),'xr','color',myRed,'linewidth', ...
   2, 'markerfacecolor', myRed);
hold on
plot(X(females,1), X(females,2),'or','color', ...
   myBlue,'linewidth', 2, 'markerfacecolor', myBlue);
xlabel('height');
ylabel('weight');
xlim([min(h) max(h)]);
ylim([min(w) max(w)]);
grid on;
```

5.3 Logistic regression

Exercise 5.2 Implement logistic regression using gradient descent.

- Write two functions `computeCost()` and `computeGradient`. The first function should return negative of the value of log-likelihood, while the second function should return the corresponding gradient.
- Write the gradient descent code for logistic regression (following the code for linear regression from Exercise 2).
- Plot predictions to get a visualization like 5.1 (b). Check if you get similar or different results.
- Do your results make sense?

■

Once you have gradient descent, it is also straightforward to implement Newton's method.

Exercise 5.3 Newton's method for logistic regression.

- Write a new function that combines `computeCost()` and `computeGradient()` in one file. Also, add the Hessian computation in this function. The function should return the cost function, gradient, and Hessian altogether.

  ```
  [L, g, H] = logisticRegLoss(beta, y, tX);
  ```

- Your gradient descent code can now be turned into a Newton's method algorithm where you compute the descent direction by solving $\mathbf{Hd} = \mathbf{g}$. You can use Matlab's backslash operator for this.
- Check if this gives the same answer as gradient descent. To debug, print the function value and the norm of the gradient in every iteration. All of these values should decrease in every iteration.

■

Exercise 5.4 Penalized logistic regression.
Modify `logisticRegLoss.m` to write a new function `penLogisticRegLoss.m` to add the regularization term $\lambda \sum_{d=1}^{D} \beta_d^2$. Set λ to a low value and check if it gives the same result. Now increase the value of λ and check whether $\boldsymbol{\beta}$ is shrinking or not.
■

Exercise 5.5 BONUS: Implement IRLS. Follow the pseudo-code given in the leture notes. ■