

Exercises to the lecture

Machine learning and Pervasive Computing

Winter term 2014

Stephan Sigg (ssigg@gwdg.de)

Exercise: 25.11.2013, 14:15 – 15:45

Assignment 2: Feature subset selection and gradient descent

In this assignment, you will find new and optimized features for the detection of your classes and implement the gradient descent function utilised for linear regression.

- a) For the data collected in Assignment 1, find and calculate at least three additional features not considered in Assignment 1.
- b) Plot the evolution of your feature values over time and add labels for the respective classes. Can you identify features from these plots that appear to be better suited to distinguish between the classes?
- c) Using the Pearson Correlation Coefficient, reduce your set of features to the three with highest expected impact.
- d) Calculate the confusion matrix for this new set of features and compare the precision and recall achieved to those previously computed. Could you improve the classification performance?
- e) Implement the gradient descent function to compute optimum regression parameters. You may extend the Octave/Matlab function below for a start. Implementations in other languages are fine too.

```
function [W, E_history] = gradientDescent(X, y, W, delta, num_iter)
    %gradientDescent learns W via gradient descent
    % W = gradientDescent(X, y, W, delta, num_iter) updates W in
    % num_iter steps with learning rate delta

    % Initialisation
    m = length(y); % # of training examples
    E_history = zeros(num_iter, 1);

    for iter = 1:num_iter

        % ===== YOUR CODE HERE =====
        % Perform gradient step on the parameter vector W.
        %

        % =====

        % Save the cost E in every iteration
        E_history(iter) = computeCost(X, y, W);

    end
end
```