Johannes Gutenberg-Universität Mainz
Institut für Informatik
Data Mining
Prof. Dr. Stefan Kramer

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Machine Learning - Sheet 6

25.06.2020
Deadline: 02.07.2020 - 18:00

Task 1: Multivariate Linear Regression *(3 Points)*

Given is the annual expenses of various livestock markets in `livestock.csv`. How do these depend on the number of animals sold? Implement a simple `multivariateLinearRegression` method (with constant term). You should use only basic linear algebra to build the model (matrix multiplication, transposition, solving linear equations etc., but *not* a complete linear least square fitting procedure).

Task 2: Logistic Regression with Nonlinear Features *(5 Points)*

The goal of this exercise is to see how logistic regression can be used to classify points in the plane, even in cases when the points are not linearly separable. Run the attached script `linearRegression.py`. You will see three point clouds, which belong to two classes. These two classes are obviously not linearly separable. Our goal is to modify two functions in the script to separate the points.

(1) (*1 point*) Implement the `gaussianRbf` function which takes coordinates of a central point $c = (c_x, c_y)$ (`centerX` and `centerY`), a squared sigma σ^2 (`sigmaSquared`), and coordinates of a point $p = (p_x, p_y)$. It is supposed to return:

$$\phi_{c,\sigma^2}(p) := \exp\left(-\frac{\|p-c\|^2}{2\sigma^2}\right).$$

(2) (*3 points*) Currently, the `extractFeatures` function takes coordinates of a point (x, y), and simply returns the coordinates (`[x, y]`), without modifying them. You should replace `[x, y]` by an array of non-linear features, computed from x and y. Try to come up with at least three different ways to separate the points. Samples are:

- Various multivariate polynomials, e.g. `[x * x, x * y]`
- `gaussianRbf` as a nonlinear feature. Experiment with various settings for center and sigma.

(3) (*1 point*) Briefly discuss your results.

Task 3: Perceptron *(2 Points)*

Design a two-layer network of perceptrons that implements the exclusive-OR (XOR) function $(A, B) \mapsto A \oplus B$. Explain how you came up with the weights, sketch the decision boundaries for all intermediate steps.