# Homework. 3

Exercise. 1 (100 points)

Implement Kernel Logistic Regression with $L_2$ regularizer using empirical kernel map, i.e., optimize,

$$J(\omega) = -\sum_{i=1}^{N} \log(\sigma(y_i \omega^\top k_i)) + \lambda \omega^\top \omega,$$

to get $\omega$. Here, $k_i$ is a column vector such that $k_i = [k(x_i, x_1). \ldots, k(x_i, x_j), \ldots, k(x_i, x_N)]^\top$, $y_i$ is a label of data point $x_i$, and $\sigma(v) = 1/(1 + e^{-v})$. Use RBF (Gaussian) kernel with $\sigma^2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \|x_i - x_j\|^2$.

After $\omega$ is obtained, for any test data $x$, compute $p(y = 1|x) = \sigma(\omega^\top k_x)$, where $k_x = [k(x, x_1), k(x, x_2), \ldots, k(x, x_N)]^\top$. If $p(y = 1|x) > 0.5)$ the predicted label is 1, else it is $-1$. Report the accuracy.

Use the following methods to optimize $J(\omega)$:

a) [15 points] GD

b) [25 points] SGD (for each iteration use $p$ points to estimate the gradients and explore two settings of $p$: $p = 1$ and $p = 100$)

c) [30 points] BFGS(randomly sample 4000 training points, i.e. 2000 from each class, and use them to describe the empirical kernel map and construct the approximation of inverse Hessian using BFGS method)

d) [30 points] repeat the same experiment as for BFGS, but instead for LBFGS, where you use a small number of vectors (experiment with a couple of choices) to approximate inverse Hessian

You will use data set “data1.mat”. Experiment with various step sizes and pick what works the best for you. Compare how the value of the cost function decreases with time for different methods. Stop the iterations, if the gradient becomes smaller than epsilon (say, $1e - 5$). Compare the methods.