Homework Assignment # 4

Due: Wednesday, December 9, 2015, 11:59 p.m.

Total marks: 100

Question 1. [10 MARKS]

Use the method of Lagrange multipliers to find extrema of function $f(x, y)$ under the given constraints.

(a) [3 MARKS] $f(x, y) = x^2 - y^2$ subject to $x^2 + y^2 = 4$

(b) [3 MARKS] $f(x, y) = x^2 y - \log x$ subject to $x + 2y = 0$

(c) [4 MARKS] $f(x, y) = x^2 + 2xy + y^2 - 2x$ subject to $x^2 - y^2 = -1$

Question 2. [40 MARKS]

In this question, you will implement kernel linear regression. Kernel linear regression can be derived using the kernel trick, where the optimal solution $\mathbf{w}$ is always a function of the training data $\mathbf{w} = \mathbf{X}^\top \boldsymbol{\alpha}$ for $\mathbf{X}^\top \in \mathbb{R}^{n \times d}$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$. Therefore, we could instead learn $\boldsymbol{\alpha}$, and whenever we predict on a new value $\mathbf{x}$, the prediction is $\mathbf{x}^\top \mathbf{w} = \mathbf{x}^\top \mathbf{X}^\top \boldsymbol{\alpha} = \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}_i)\alpha_i$ with $k(\mathbf{x}, \mathbf{x}_i) = \langle \mathbf{x}, \mathbf{x}_i \rangle$ in this case. In general, we can extend to other feature representations on $\mathbf{x}$, giving $\phi(\mathbf{x})$ and so a different kernel $k(\mathbf{x}, \mathbf{x}_i) = \langle \mathbf{x}, \mathbf{x}_i \rangle$.

The kernel trick is useful conceptually, and for algorithm derivation. In practice, when implementing kernel regression, we do not need to consider the kernel trick. Rather, the procedure is simple, involving replacing your current features with the kernel features and performing standard regression. For learning, we replace the training data with the new kernel representation:

$$\mathbf{K}_{\text{train}} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{c}_1) & k(\mathbf{x}_1, \mathbf{c}_2) & \dots & k(\mathbf{x}_1, \mathbf{c}_k) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathbf{x}_n, \mathbf{c}_1) & k(\mathbf{x}_n, \mathbf{c}_2) & \dots & k(\mathbf{x}_n, \mathbf{c}_k) \end{bmatrix} \in \mathbb{R}^{n \times k}$$

for some chosen centers (above those chosen centers were the training data samples $\mathbf{x}_i$). For example, for the linear kernel above with $k(\mathbf{x}, \mathbf{x}_i) = \langle \mathbf{x}, \mathbf{x}_i \rangle$, the center is $\mathbf{c} = \mathbf{x}_i$. Notice that the number of features is now k, the number of selected centers, as opposed to the original dimension d. Once your've transformed your data to this new representation, then you learn $\mathbf{w}$ with linear regression as usual, such that $\mathbf{K}_{\text{train}}\mathbf{w}$ approximates $\mathbf{y}_{\text{train}}$. As before, you can consider adding regularization. The prediction is similarly simple, where each new point is transformed into a kernel representation using the selected centers.

(a) [20 MARKS] Implement this algorithm. Run it on the dataset from Assignment 2.

(b) [5 MARKS] Explain your implementation choices.

(c) [15 MARKS] Try three different kernels. Explain any differences in behavior on this dataset.

Question 3. [50 MARKS]

Take two methods from the last assignment, that you implemented, and run them on a single dataset, the SUSY dataset. Proper comparisons should be made using 10-fold cross-validation experiments. Use your knowledge about model comparison to formally conclude which of the two algorithms is better. This includes proper training-test splits, statistical significance tests, and proper meta-parameter selection techniques (e.g., cross-validation). You can use statistical significance tests built-in to python (or other languages). Provide a precise conclusion of your experiment.

Homework policies:

Your assignment will be submitted as a single pdf document and a zip file with code, on canvas. The questions must be typed; for example, in Latex, Microsoft Word, Lyx, etc. or must be written legibly and scanned. Images may be scanned and inserted into the document if it is too complicated to draw them properly. All code (if applicable) should be turned in when you submit your assignment. Use Matlab, Python, R, Java or C.

Policy for late submission assignments: Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rule:

on time: your score 1

1 day late: your score 0.9

2 days late: your score 0.7

3 days late: your score 0.5

4 days late: your score 0.3

5 days late: your score 0.1

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All assignments are individual, except when collaboration is explicitly allowed. All the sources used for problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

Good luck!