Lab #4

Install MOSEK on your local machine

1. First go to this website

   `https://www.mosek.com/cgi-bin/student.py`

   and request a free academic license for MOSEK using your .edu email.

2. Download MOSEK trial version from MOSEK website, unzip it

3. Replace the pre-bundle license file with the one in the email you got from MOSEK (in mosek/6/licenses)

4. Follow install instructions from

   `http://mosek.com/fileadmin/products/6_0/tools/doc/html/toolsinstall/node003.html`

5. To use MOSEK with MATLAB, follow

   `http://mosek.com/fileadmin/products/6_0/tools/doc/html/toolbox/node006.html`

Alternatively, you can login to *c199.eecs.berkeley.edu* to use MATLAB/MOSEK

```
add path /share/b/mosek/toolbox/solvers/
```

Support Vector Machine (SVM) with L1-regularization

Given a set of training examples $\{(x_i, y_i)\}_{i=1}^n$ where the input $x_i \in R^d$ is a d-dimensional vector and the output $y_i \in \{-1, 1\}$ is the binary variable, the two-class classification problem aims to find a classification rule from training data so that given new input $x \in R^d$, we can "optimally" assign a class label $y \in \{-1, 1\}$ to x. The general SVM problems seeks for the optimal solution of the following optimization problem

$$\min_{w \in R^d, b \in R} \sum_{i=1}^n l(y_i(w^T x_i + b)) + \Omega(w)$$

where $l(.)$ is the loss function and $\Omega(.)$ is the regularizer. In this lab, we will work with hinge loss $l(t) = \max(0, 1-t) = [1-t]_+$ and L1 regularization $\Omega(w) = \lambda\|w\|_1$

$$\min_{w \in R^d, b \in R} \sum_{i=1}^n \left[1 - y_i(w^T x_i + b)\right]_+ + \lambda\|w\|_1$$

1. Show how to formulate L1-SVM as a linear program.

   WHICH DATASET TO USE?

2. Use CVX to solve the problem with the given data.

3. Use MOSEK to solve the problem with the given data.

   msklpopt function is designed to solve linear programm of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & l^c \leq Ax \leq u^c \\ & l^x \leq x \leq u^x \end{array}$$

   where l^c, u^c are constraint bounds where as l^x and u^x are variable bounds.

```
[res] = msklpopt(c,A,lc,uc,lx,ux);
res.sol % the optimal solution
res.sol.bas.xx % x solution in basic solution
```

4. Compare CVX and MOSEK in solving the problem. Which one is more efficient?