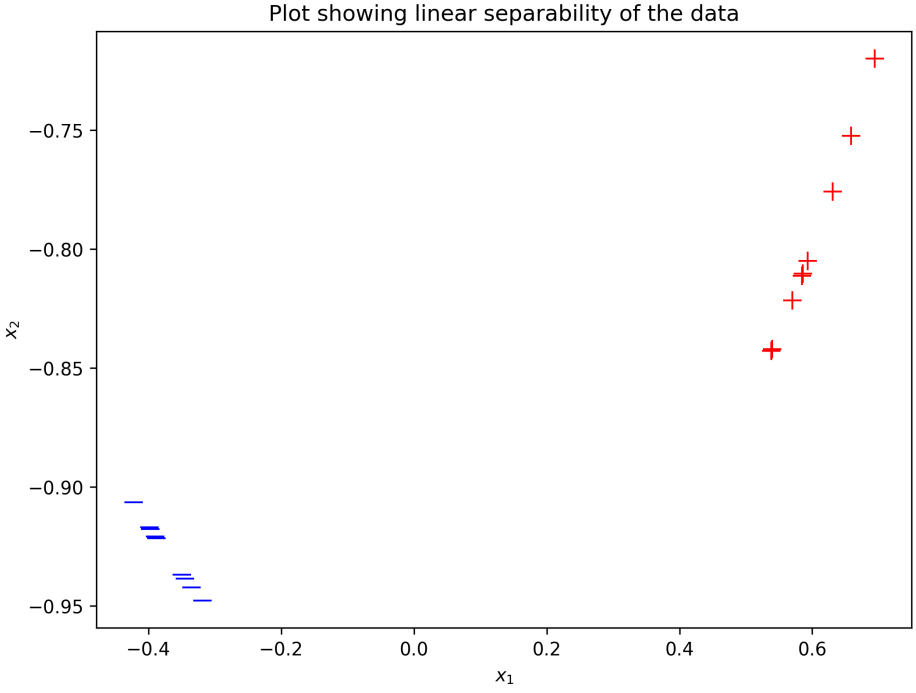
MA4270: Computational Exercise Solutions
AY2017/18 Sem II

Total marks: 100

[45pts] Problem 1: Perceptron

1. [5pts] Plot the dataset (using different colors/markers for the two different classes) to show it is linearly separable. You may find the information on these websites[1] useful for plotting.

   Solution:



2. [5pts] Find θ^* (the optimal θ; please normalise it to be of length one) and corresponding

$$\gamma^* := \min_{t\in\{1,\ldots,n\}} y_t\langle\theta^*, x_t\rangle$$

   using a linearly separable (primal) SVM without offset and without slack variables. To do this, you can use the `quadprog` function in MATLAB or `quadprog` module in Python. Again, please be reminded not to use any SVM package.

[1] MATLAB: https://www.mathworks.com/help/stats/kmeans.html
Python: https://matplotlib.org/devdocs/api/_as_gen/matplotlib.pyplot.plot.html

Solution:

$$\theta^* = \begin{bmatrix} 0.9926 \\ 0.1216 \end{bmatrix}$$

$$\gamma^* = 0.4315$$

3. Write your code for the standard perceptron algorithm, and

   (a) [5pts] Using the zero vector as the starting point (i.e., $\theta^{(0)} = 0$), report the number of updates required to successfully classify the dataset, the converged solution θ (also normalise it to be of length one), and the corresponding

   $$\gamma := \min_{t \in \{1,\ldots,n\}} y_t \langle \theta, x_t \rangle.$$

   Use the natural order, i.e., update your parameter vector by cycling through the points in the order $\{1, 2, 3, \ldots\}$.

   Solution: # updates required, $k = 2$

   $$\theta = \begin{bmatrix} 0.9837 \\ 0.1801 \end{bmatrix}$$

   $$\gamma = 0.3775$$

   (b) [5pts] Run your code 10 times with different starting points and report your results. What do you observe about the results? In particular, are the parameter vectors obtained the same?

   Solution: Most solutions are acceptable. Ideally with a range of $\theta^{(0)}$, with γ no larger than γ^*. Following is a sample:

	$(\theta^{(0)})^T$	k	$(\theta^{(k)})^T$	γ
1	[0, 1]	1	[0.9361, 0.3518]	0.2071
2	[1, 0]	0	[1, 0]	0.3185
3	[1, 1]	1	[0.9890, 0.1476]	0.4077
4	[-1, 0]	4	[0.9656, 0.2600]	0.3003
5	[0, -1]	1	[0.9841, -0.1775]	0.1452
6	[-1, -1]	3	[0.9980, 0.0636]	0.3782
7	[99, 0]	0	[1, 0]	0.3185
8	[0, 99]	84	[0.8449, 0.5349]	0.0037
9	[0, 999]	846	[0.8435, 0.5372]	0.00097
10	[-99, 0]	204	[0.8869, 0.4620]	0.0877

   (c) [10pts] Suppose the starting point $\theta^{(0)}$ is drawn from the boundary of the unit circle. Prove that the number of iterations required to successfully classify the dataset in terms of the starting point $\theta^{(0)}$ is upper bounded as

   $$k \leq \max \left\{ -\frac{a}{\gamma^*}, \frac{1 - 2a\gamma^* + \sqrt{(2a\gamma^* - 1)^2 - 4(\gamma^*)^2(a^2 - 1)}}{2(\gamma^*)^2} \right\}.$$

   where $a = \langle \theta^*, \theta^{(0)} \rangle$ and γ^*, θ^* are defined as in part 2.

   Hint: The proof is very similar to that in lec2.pdf of the MIT (Jaakkola) lecture notes. Note that the starting point is not zero vector and you need to modify equations (3) and (8) in that proof.

Solution: Consider

$$\begin{aligned}\langle \theta^*, \theta^{(k)} \rangle &= \langle \theta^*, \theta^{(k-1)} + y_t x_t \rangle \\ &= \langle \theta^*, \theta^{(k-1)} \rangle + y_t \langle \theta^*, x_t \rangle \\ &\geq \langle \theta^*, \theta^{(k-1)} \rangle + \gamma^* \\ &\geq \langle \theta^*, \theta^{(0)} \rangle + k\gamma^* \\ &= a + k\gamma^* \end{aligned} \tag{1}$$

Next

$$\begin{aligned}\|\theta^{(k)}\|^2 &= \|\theta^{(k-1)} + y_t x_t\|^2 \\ &= \|\theta^{(k-1)}\|^2 + 2y_t \langle \theta^{(k-1)}, x_t \rangle + y_t^2 \|x_t\|^2 \\ &\leq \|\theta^{(k-1)}\|^2 + 1 \\ &\leq \|\theta^{(0)}\|^2 + k \end{aligned} \tag{2}$$

where the second last inequality follows since $y_t \langle \theta^{(k-1)}, x_t \rangle < 0$ whenever an update is made and $\|x_t\| = 1 \ \forall t$.
By Cauchy-Schwarz and from (2),

$$\begin{aligned}\langle \theta^*, \theta^{(k)} \rangle^2 &\leq \|\theta^*\|^2 \|\theta^{(k)}\|^2 \\ &\leq \|\theta^*\|^2 (\|\theta^{(0)}\|^2 + k)\end{aligned}$$

Combining with (1), for the case $a + k\gamma^* > 0$ such that squaring both sides in (1) preserves the inequality,

$$\begin{aligned}(a + k\gamma^*)^2 &\leq \|\theta^*\|^2 (\|\theta^{(0)}\|^2 + k) \\ a^2 + 2ak\gamma^* + (\gamma^*)^2 k^2 &\leq \|\theta^*\|^2 \|\theta^{(0)}\|^2 + k\|\theta^*\|^2\end{aligned}$$

Assuming $\|\theta^*\|^2 = 1$ and $\|\theta^{(0)}\|^2 \leq 1$, above becomes

$$\begin{aligned}a^2 + 2ak\gamma^* + (\gamma^*)^2 k^2 &\leq 1 + k \\ (\gamma^*)^2 k^2 + (2ak\gamma^* - 1)k + (a^2 - 1) &\leq 0\end{aligned}$$

Solving the quadratic inequality yields

$$k \leq \frac{1 - 2a\gamma^* + \sqrt{(2a\gamma^* - 1)^2 - 4(\gamma^*)^2 (a^2 - 1)}}{2(\gamma^*)^2} \tag{3}$$

For the case $a + k\gamma^* \leq 0$ we have that

$$k \leq -\frac{a}{\gamma^*} \tag{4}$$

Combine (3) and (4) gives

$$k \leq \max \left\{ -\frac{a}{\gamma^*}, \frac{1 - 2a\gamma^* + \sqrt{(2a\gamma^* - 1)^2 - 4(\gamma^*)^2 (a^2 - 1)}}{2(\gamma^*)^2} \right\}$$

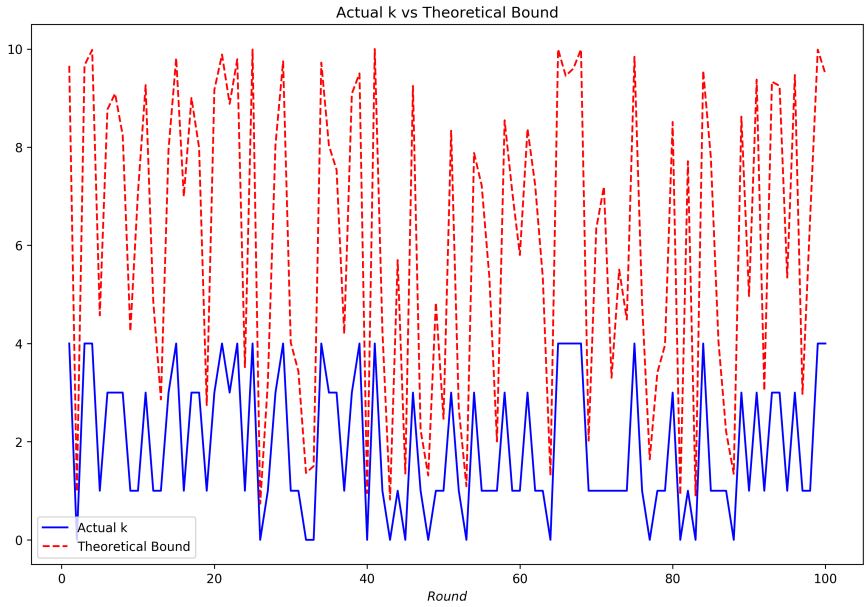
□

(d) [10pts] Run your code 10000 times with starting points randomly drawn from the unit circle. You can refer to this website[2] to find out how to sample points uniformly at random from the unit sphere.

Plot the results of the first 100 runs by comparing the number of updates required to successfully classify the dataset (i.e. achieve zero training error) and the corresponding theoretical bound.

Report the average of the number of updates over the 10000 required to successfully classify the dataset and the average of the γ's. Compare this average with the γ^* derived from part 2.

Solution:



Average # of updates ≈ 1.821
Average $\gamma \approx 0.2461$

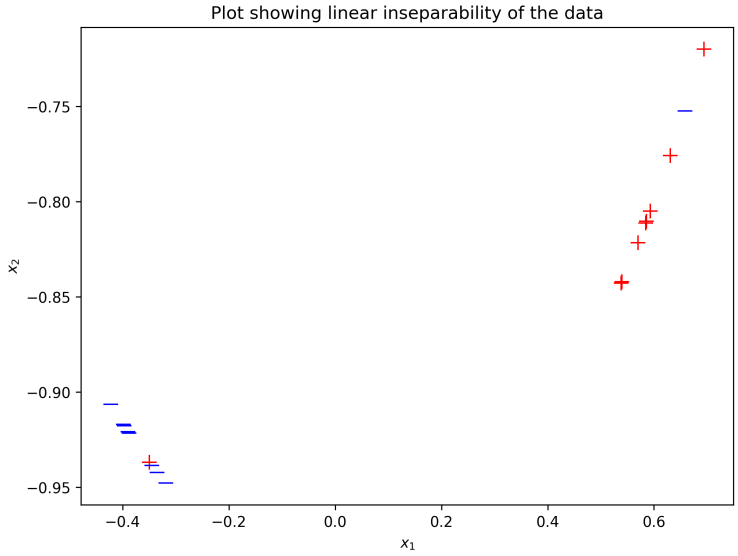
The average γ is no larger than the optimal γ^*, as expected.

4. [5pts] Change the label of the first sample to -1 and change the label of the third sample to $+1$.

   - Plot the dataset to show it is not linearly separable.
   - Run part 2 again. Can we find a feasible solution from the SVM without offset and without slack variables?
   - Run the standard perceptron algorithm with any initial vector. Does the standard perceptron algorithm converge?

   Solution: Not linearly separable:

[2] http://mathoverflow.net/questions/24688/efficiently-sampling-points-uniformly-from-the-surface-of-an-n-sphere



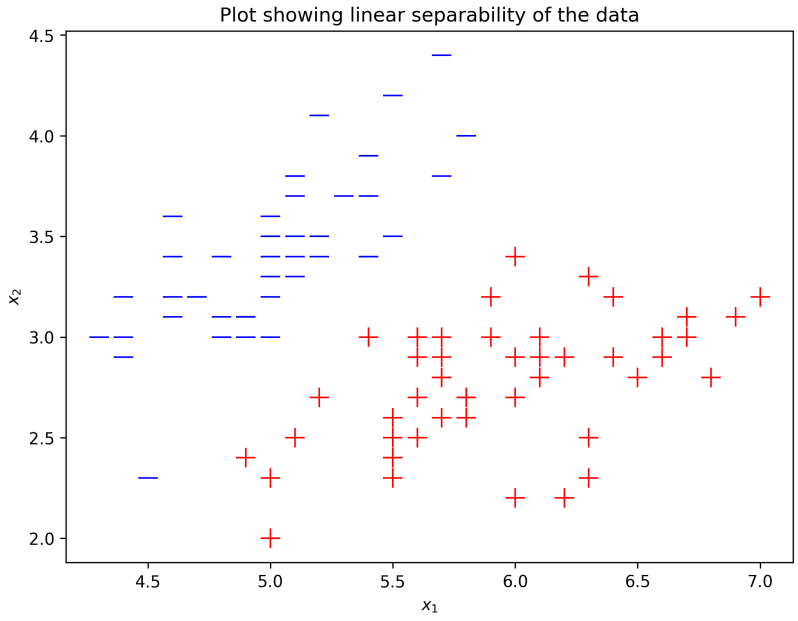
Solving primal SVM without slack/offset in MATLAB/Python gives that no feasible solution can be found/constraints cannot be satisfied etc.

The standard perceptron algorithm runs indefinitely in MATLAB/Python and does not converge.

[30pts] Problem 2: Hard-SVM

1. [5pts] Plot the dataset to show it is linearly separable;

   Solution:



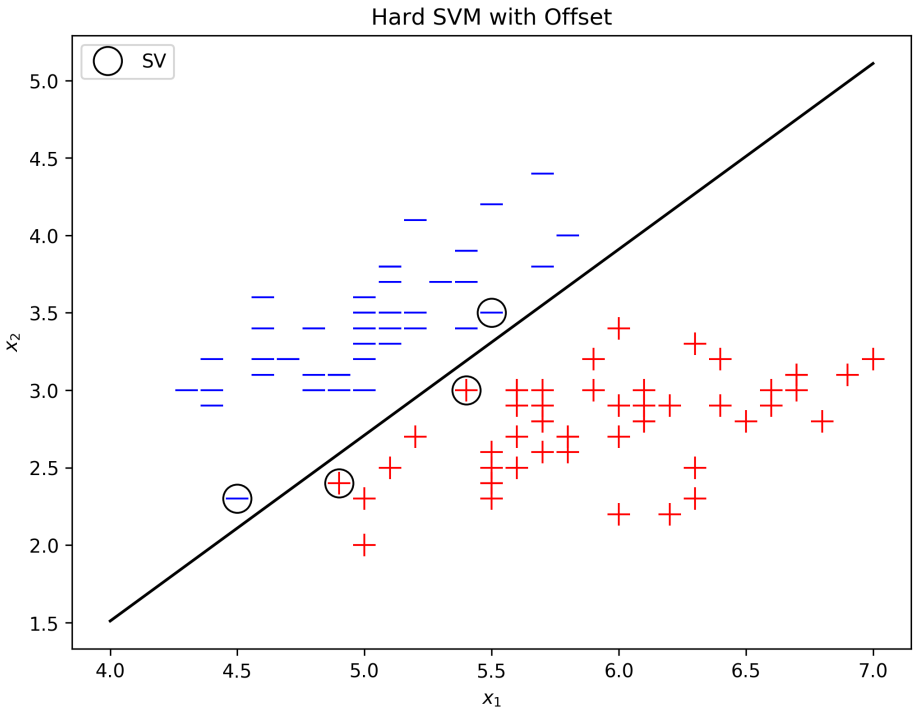
2. [5pts] Solve the *primal* problem of the SVM for this linearly separable dataset, and report the optimal θ, optimal θ_0 and the optimal objective function value $\frac{1}{2}\|\theta\|_2^2$;

   Solution:

$$\theta = \begin{bmatrix} 6.3158 \\ -5.2632 \end{bmatrix}$$
$$\theta_0 = -17.3158$$
$$\frac{1}{2}\|\theta\|^2 = 33.7950$$

3. [2pts]Plot the line $\theta^T x + \theta_0 = 0$ and indicate all the support vectors.

   Solution: Support vector indices are [37, 42, 58, 85]. (Will accept [36, 41, 57, 84] as well since most programming languages count from $0, 1, 2, \ldots$. However in class our notation is $t = 1, 2, \ldots, n$ so it is better to start counting from 1 in this case)



4. [10pts] Solve the *dual* problem of the SVM (without slack variables) for this linearly separable dataset. Report the non-zero (larger than 10^{-6}) entries of the optimal $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ vector and their indices and report the optimal objective function value for the dual problem, i.e.,

$$\sum_{t=1}^{n} \alpha_t - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

   What do you observe? In particular, is the dual optimal objective value the same as the primal? Which problem (dual or primal) takes a shorter time to solve? Why?

Solution: Optimal objective is 33.795. We have 4 support vectors corresponding to:

$$\alpha_{37} \approx 17.6317$$
$$\alpha_{42} \approx 16.1633$$
$$\alpha_{58} \approx 12.9361$$
$$\alpha_{85} \approx 20.8589$$

We observe that the support vectors identified in the dual (where $\alpha > 0$) correspond to the tight constraints in the primal (where $y_t(\theta^T x_t + \theta_0) = 1$).

The optimal primal and dual objectives are the same.

The primal is faster to solve. The primal has $p + 1$ variables to optimize for and $n = 100$ constraints. The dual has n variables to be optimized for and $\approx n$ inequality and equality constraints. Since $n \gg p = 2$, the primal is faster.

5. [8pts] Calculate

$$\sum_{i=1}^{n} \alpha_i y_i x_i$$

By arbitrarily picking a $j \in \{1, 2, \ldots, n\}$ which satisfies $\alpha_j > 0$, calculate as well

$$y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j.$$

What do you observe? In particular, what are the two terms in the displayed equations above equal to?

Solution: We have

$$\sum_{i=1}^{n} \alpha_i y_i x_i = \begin{bmatrix} 6.3157 \\ -5.2630 \end{bmatrix}$$

We observe that the above is equal to θ, and this is to be expected based on KKT stationarity conditions.

For arbitrary j such that $\alpha_j > 0$ (say $j = 37$), we have:

$$y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j = -17.3156$$

The above is equal to θ_0. This is because when $\alpha_j > 0$, by complementary slackness the primal constraint $y_j(\theta^T x_j + \theta_0) \geq 1$ is tight and we can solve for θ_0.

[25pts] Problem 3: Soft-SVM and Cross-validation

1. [15pts] Solve the relaxed (primal) SVM problem by setting $C = 100$. Report the optimal θ, optimal θ_0 and the optimal objective function value

$$\frac{1}{2}\|\theta\|_2^2 + C\sum_{t=1}^{n} \xi_t.$$

Report the number of misclassified samples.

Solution: Optimal objective 654.1942 with parameters:

$$\theta^* = \begin{bmatrix} -1.8478 \\ -3.2609 \\ 4.6739 \\ 10.8696 \end{bmatrix}$$

$$\theta_0^* = -20.4130$$

With **3** misclassified samples, i.e. those with $\text{sign}(\theta^T x_j + \theta_0) \neq y_j$.

Note: Common mistake is to use $\xi_t > 0$ to check if the t^{th} sample is misclassified. It is possible that the sample lies *within* the margins but on the correct side. For instance in this problem with $\xi_t > 10^{-6}$:

```
xi[20] = 1.250, y[20] = -1, pred X[20,] = [ 1.]
xi[27] = 0.272, y[27] = -1, pred X[27,] = [-1.]
xi[33] = 1.924, y[33] = -1, pred X[33,] = [ 1.]
xi[83] = 2.043, y[83] = 1, pred X[83,] = [-1.]
xi[88] = 0.283, y[88] = 1, pred X[88,] = [ 1.]
```

2. [10pts] Now we are going to explore using cross-validation to find a good C. By 10-*fold cross-validation*, we mean that in the first run, we use the first 10 samples as validation set, and other 90 samples as training dataset; then we use the next 10 samples as validation set, and other 90 samples as training dataset and so on. Finally, we use the last 10 samples as validation set, and other 90 samples as training dataset. Then we average the resultant 10 test errors.

   There are three candidates for $C \in \{1, 100, 10000\}$. For each fixed C, use 10-fold cross validation and report the misclassified rate, i.e., the averages of the 10 test errors. Thus, find the best choice of C among these three values.

   Solution:

   For $C = 1$, total 4 misclassified samples.
   For $C = 100$, total 6 misclassified samples.
   For $C = 10000$, total 7 misclassified samples.

   Hence we may pick $C = 1$ as the best choice.