Andrew Elysee (aelys2)

CS446

Homework 5

1.

a. The definition of the optimization problem is

$$\min_{v,b} \frac{1}{2}\|v\|^2$$

You can easily rationalize that $\|v\|$ is a convex function because as you move towards the 0 vector from any direction $\|v\|$ approaches 0 which means the second derivative or hessian is positive semidefinite. Also, the second derivative of $f(x) = x^2$ is 2 which is greater than 0, so x^2 must also be convex. Making the composite function of these two convex functions, we can see that $\frac{1}{2}\|v\|^2$ would be convex. Additionally, we can see that $y_i(v^T x_i + b) \geq 1$ is a convex set. Thus, the second formulation is a convex program.

b. If you set v equal to the 0 vector in $\mathbb{R}^d$ (since this is the smallest possible magnitude squared of vector v), then you can choose any $b \geq 1$ and all would be valid solutions to the optimization problem since $y_i(v^T x_i + b) \geq 1$ would be true. For example for $b = 1$ and $b = 2$:
If $b = 1$ then:

$$\begin{aligned} &y_i(v^T x_i + b) \geq 1 \\ &y_i(0x_i + 1) \geq 1 \\ &1(0 + 1) \geq 1 \\ &1 \geq 1 \end{aligned}$$

if $b = 2$ then:

$$\begin{aligned} &y_i(v^T x_i + b) \geq 1 \\ &y_i(0x_i + 2) \geq 1 \\ &1(0 + 2) \geq 1 \\ &2 \geq 1 \end{aligned}$$

As we can see both resolve as true showing that there is no unique solution for the second formulation in this situation.

c. The explicit constraints to the first convex program is:
$\begin{bmatrix} e_1 \\ 1 \end{bmatrix}^T u \geq 1$ and $-\begin{bmatrix} -ae_1 \\ 1 \end{bmatrix}^T u \geq 1$

d. For u_a:

$$\lim_{a\to\infty} \frac{1}{2}\|\bar{u}_a\|^2 = \lim_{a\to\infty} \frac{1}{2}\left\|\frac{1}{2}e_1 + \frac{1}{2}e_{d+1}\right\|^2 = \frac{1}{2}(1)^2 = \frac{1}{2}$$

For v_a:

$$\lim_{a\to\infty} \frac{1}{2}\|\bar{v}_a\|^2 = \lim_{a\to\infty} \frac{1}{2}\left\|\frac{2}{a+1}e_1\right\|^2 = \frac{1}{2}\left\|\frac{2}{\infty+1}e_1\right\|^2 = \frac{1}{2}\|0e_1\|^2 = \frac{1}{2}(0)^2 = 0$$

e. I prefer the first version a little bit more because it feels more consistent and it is able to provide a unique solution more often than the second formulation. For me the first version is slightly more intuitive for me.

2.

a. For the hard-margin case,

$$\left(\Pi_{[0,\infty)^n}[\alpha]\right)_i = \left(\arg\min_{a' \in \mathcal{C}} \|\alpha' - \alpha\|_2\right)_i = \arg\min_{a' \in \mathcal{C}} |{\alpha'}_i - \alpha_i|$$

In order to minimize $|{\alpha'}_i - \alpha_i|$, ${\alpha'}_i$ must be as close to α_i and since α' can be any vector in $\mathcal{C}$, ${\alpha'}_i = \alpha_i$ unless α_i is negative. If it is negative, then we have to cap it to zero. Thus, in this case $|{\alpha'}_i - \alpha_i| \geq |\max\{\alpha_i, 0\} - \alpha_i|$, because at its best case we can produce α_i meaning the 2 would be equal but in the worst case we produce 0 and that is the best that we can do. Therefore,

$$\left(\Pi_{[0,\infty)^n}[\alpha]\right)_i = \max\{\alpha_i, 0\}$$

For the soft-margin case,

$$\left(\Pi_{[0,C)^n}[\alpha]\right)_i = \left(\arg\min_{a' \in \mathcal{C}} \|\alpha' - \alpha\|^2\right)_i = \arg\min_{a' \in \mathcal{C}} |{\alpha'}_i - \alpha_i|$$

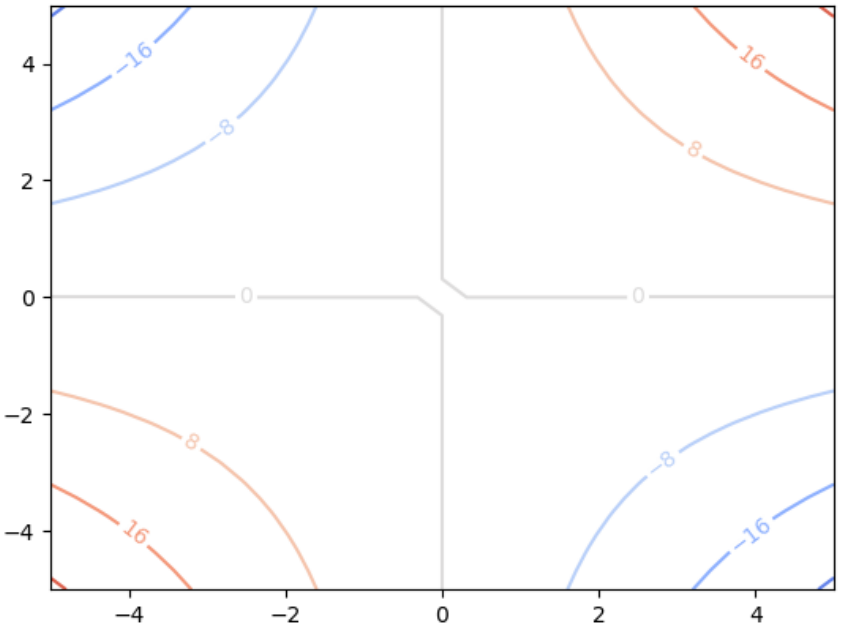
We can use similar reasoning in the soft margin case the hard margin case, except now we are unable to produce both negative numbers as well as numbers greater than C, in other words $0 \leq {\alpha'}_i \leq C$. So since ${\alpha'}_i = \alpha_i$, so long as $0 \leq {\alpha'}_i \leq C$, then $|{\alpha'}_i - \alpha_i| \geq |\min\{\max\{\alpha_i, 0\}, C\} - \alpha_i|$ because at its best case we can produce α_i meaning our minimum argument would be equal to α_i but in the worst case we produce 0 or C and that is the best that we can do. Therefore,

$$\left(\Pi_{[0,\infty)^n}[\alpha]\right)_i = \min\{\max\{\alpha_i, 0\}, C\}$$

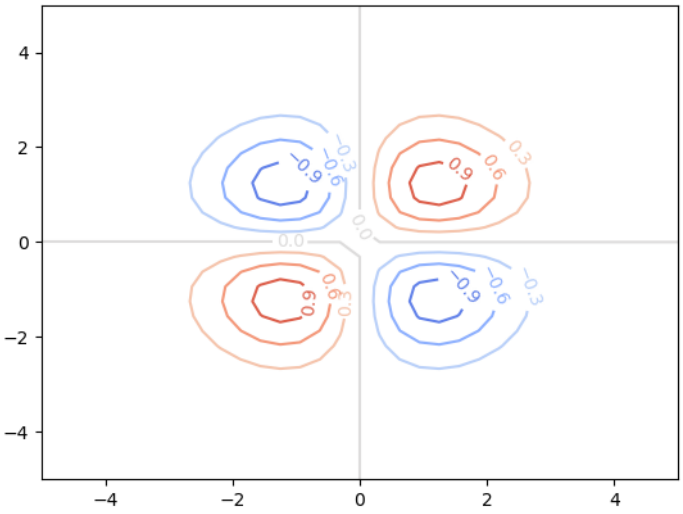
b. In hw5.py

c. In hw5.py

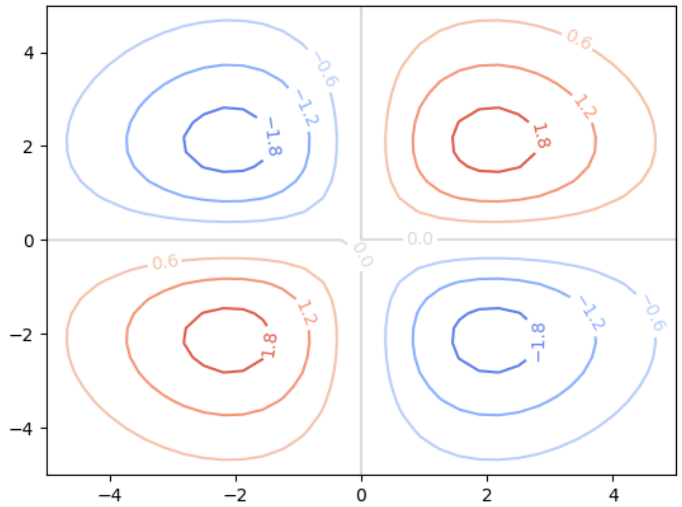
d. Polynomial with degree 2



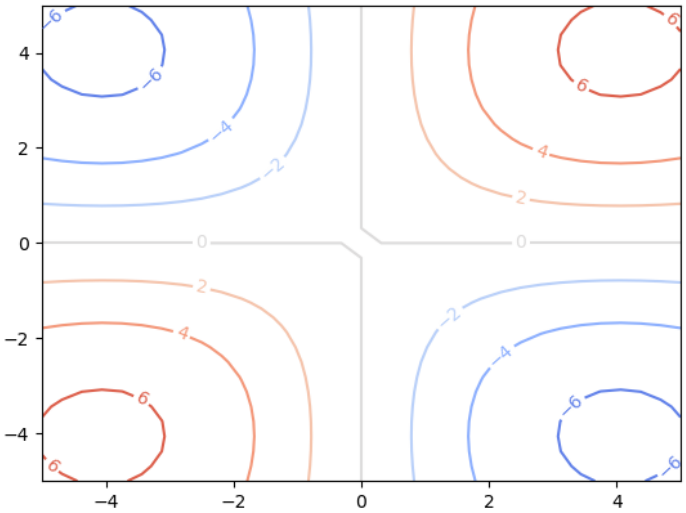
RBF with $\sigma = 1$



RBF with $\sigma = 2$



RBF with $\sigma = 4$



3.

a.

$$\hat{\mathcal{R}}_{log}(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + \exp(-y_i w^T x_i))$$

$$\nabla\hat{\mathcal{R}}_{log}(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathrm{d}}{dw}(\ln(1 + \exp(-y_i w^T x_i)))$$

$$\nabla\hat{\mathcal{R}}_{log}(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{-y_i x_i \exp(-y_i w^T x_i)}{1 + \exp(-y_i w^T x_i)}$$

$$\nabla\hat{\mathcal{R}}_{log}(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{-y_i x_i \exp(-y_i w^T x_i)}{1 + \exp(-y_i w^T x_i)} \times \frac{\exp(y_i w^T x_i)}{\exp(y_i w^T x_i)}$$

Since $e^{-y_i w^T x_i} \times e^{y_i w^T x_i} = e^{y_i w^T x_i - y_i w^T x_i} = e^0 = 1$, we get that

$$\nabla\hat{\mathcal{R}}_{log}(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{-y_i x_i}{1 + \exp(y_i w^T x_i)}$$

Now that we have the gradient of our empirical risk, we can plug it into the gradient update,

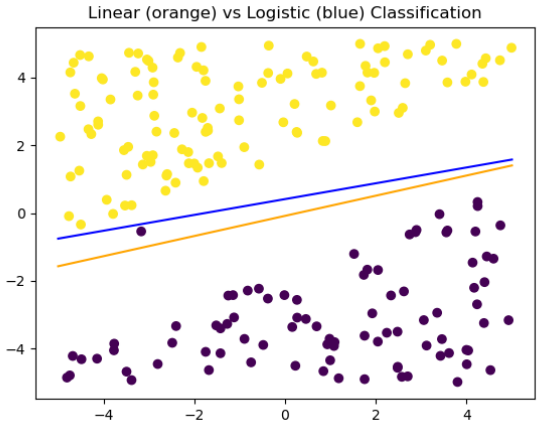
$$w' \leftarrow w - \eta\nabla\hat{\mathcal{R}}_{log}(w) = w - \frac{\eta}{n}\sum_{i=1}^{n} \frac{-y_i x_i}{1 + \exp(y_i w^T x_i)}$$

So our final Gradient Descent update rule looks is:

$$w' \leftarrow w - \frac{\eta}{n}\sum_{i=1}^{n} \frac{-y_i x_i}{1 + \exp(y_i w^T x_i)}$$

b. In hw5.py

c. I think that looking at the plot below the logistic regression does a better job of classifying these data points because it clearly divides them into 2 separate groups while the linear classifier leaves one point on the wrong side of its boundary. With that being said I think it is important to acknowledge that if this is a sample of points, then the true population may not be linearly separable meaning that the linear classifier boundary could possibly work better in that case since it is not over correcting for the one outlier.



4.

a. Using algebraic manipulation, we get,

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] \geq \frac{3n}{14}\right] = \Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{10n}{14} - \frac{7n}{14}\right] =$$

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{5n}{7}\left(1 - \frac{4}{5}\right) + \frac{10n}{14} \times \frac{4}{5} - \frac{7n}{14}\right] = \Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{5n}{7}\mathbb{E}[Z_i] + \frac{n}{14}\right]$$

Asserting Hoeffding's inequality,

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{5n}{7}\mathbb{E}[Z_i] + \frac{n}{14}\right] = \Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{5n}{7}\mathbb{E}[Z_i] + \frac{5n}{7} \times \frac{n}{10}\right]$$

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Z_i \geq \frac{5n}{7}\mathbb{E}[Z_i] + \frac{5n}{7} \times \frac{n}{10}\right] \leq \exp\left(-2 \times \frac{5n}{7}\left(\frac{1}{10}\right)^2\right) = \exp\left(-\frac{-n}{70}\right)$$

Thus we have shown that,

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] \geq \frac{3n}{14}\right] \leq \exp\left(-2 \times \frac{5n}{7}\left(\frac{1}{10}\right)^2\right) = \exp\left(-\frac{-n}{70}\right)$$

b. It is important to consider that the worst-case scenario for the probability of the remaining $\frac{2n}{7}$ classifiers is that they are all wrong 100% of the time meaning that $p = 0$. In the real world of course this would probably never happen. Since this is the worst case scenario, it will result in the highest possible probability for $\Pr[\text{MAJ}(x; f_1, \ldots, f_n) \neq y]$ and thus we can use it to prove that all other possible behaviors for the arbitrary classifiers will result in a probability at less than or equal to .

$$\Pr[\text{MAJ}(x; f_1, \ldots, f_n) \neq y] = \Pr\left[\sum_{i=1}^{n} Ind[f_i(x) \neq y] \geq \frac{n}{2}\right] \leq$$

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] + \frac{2n}{7} \geq \frac{n}{2}\right]$$

Since the last $\frac{2n}{7}$ classifiers will always be wrong in the worst case we can split our summation into 2 pieces and resolve one to be $\frac{2n}{7}$.

$$\Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] + \frac{2n}{7} \geq \frac{n}{2}\right] = \Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] \geq \frac{n}{2} - \frac{2n}{7}\right]$$

$$= \Pr\left[\sum_{i=1}^{\frac{5n}{7}} Ind[f_i(x) \neq y] \geq \frac{3n}{14}\right] \leq \exp\left(-\frac{n}{70}\right)$$

Using 4a, we can reach our desired result. Thus, we have shown that,
$\Pr[\text{MAJ}(x; f_1, \ldots, f_n) \neq y] \leq \exp(-\frac{n}{70})$.

c. The probability of correctly classifying x for large n is good and actually gets better the more n that we have because as n approaches infinity the exponent will approach 0.