Machine Learning homework 6 solution

Constrained Optimization and SVM

Wiktor Jurasz - M.Nr. 03709419

December 2, 2018

1 Problem 1

1.1 Strong Duality

$$f_0(\theta) = \theta_1 - \sqrt{3}\theta_2 \tag{1}$$

$$f_1(\theta) = \theta_1^2 + \theta_2^2 - 4 \tag{2}$$

1. f_0 and f_1 are convex (sum of convex functions)
2. There exists θ (e.g $[0,0]$) for which $f_1 < 0$

From 1 and 2 $\Rightarrow$ Slater's constraint qualification is fulfilled $\Rightarrow$ Strong duality holds

1.2 f_0 Minimum

$$L(\theta_1, \theta_2, \alpha) = \theta_1 - \sqrt{3}\theta_2 + \alpha(\theta_1^2 + \theta_2^2 - 4) \tag{3}$$

$$\nabla_\theta L(\theta_1, \theta_2, \alpha) = [1 + 2\alpha\theta_1, -\sqrt{3} + 2\alpha\theta_2] \tag{4}$$

$$\nabla_\theta L(\theta_1, \theta_2, \alpha) = 0 \Leftrightarrow \theta_1 = \frac{-1}{2\alpha} \wedge \theta_2 = \frac{\sqrt{3}}{2\alpha} \tag{5}$$

$$\theta^*(\alpha) = \operatorname*{argmin}_\theta L(\theta_1, \theta_2, \alpha) = [\frac{-1}{2\alpha}, \frac{\sqrt{3}}{2\alpha}] \tag{6}$$

$$g(\alpha) = L(\theta^*(\alpha), \alpha) = -\frac{1}{2\alpha} - \frac{3}{2\alpha} + \frac{1}{4\alpha} + \frac{3}{4\alpha} - 4\alpha = -\frac{1}{\alpha} - 4\alpha \tag{7}$$

$$\begin{gathered} g'(\alpha) = \alpha^{-2} - 4 = 0 \Leftrightarrow \alpha = \frac{1}{2} \\ g''(\alpha) = -2\alpha^{-3} \\ g''(\frac{1}{2}) = -16 < 0 \Rightarrow max(g(\alpha)) = -4 = min_\theta(L(\theta_1, \theta_2, \alpha)) \end{gathered} \tag{8}$$

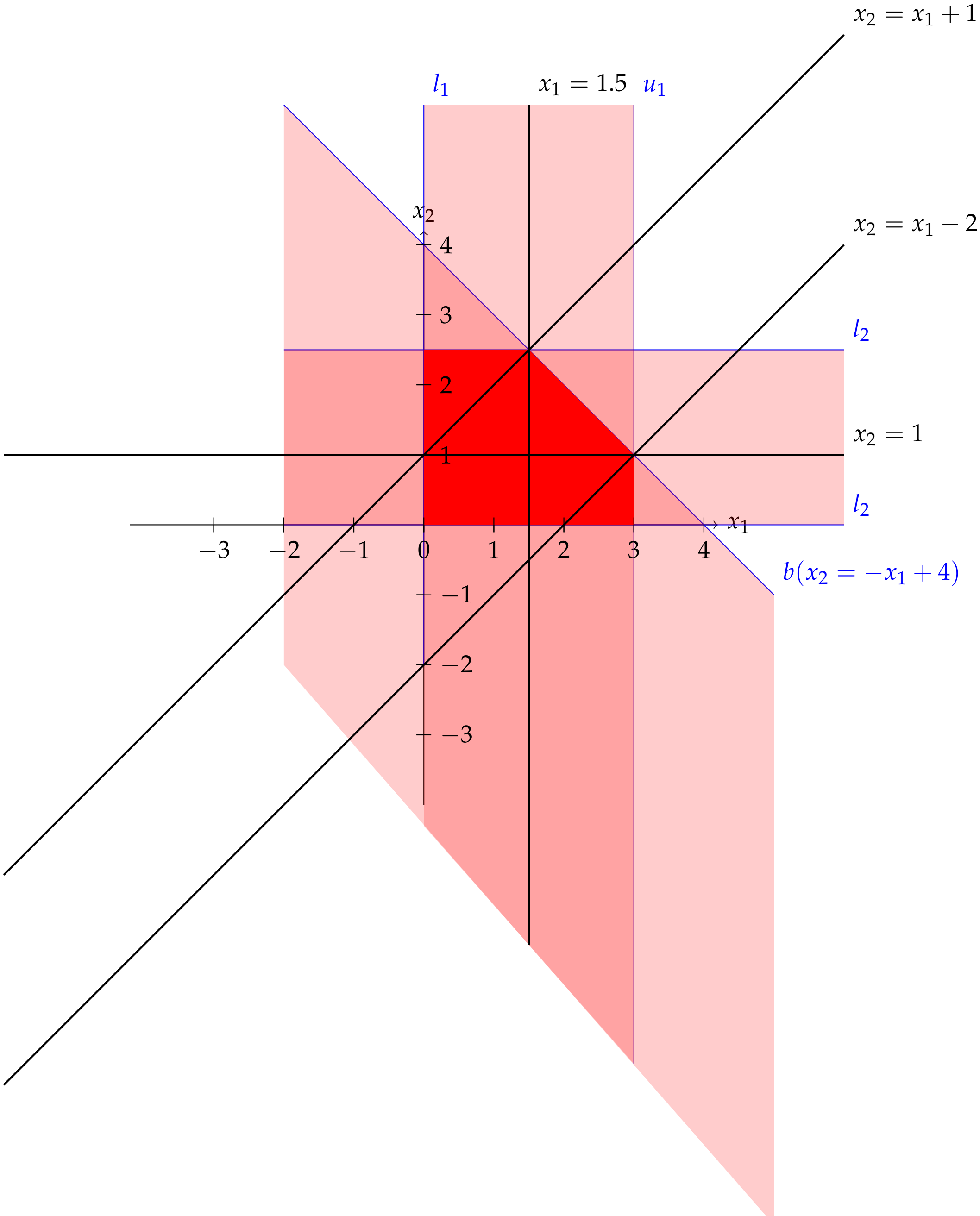
Now we can go back to the gradient of L to determine θ for which f_0 has minimum.

$$\begin{gathered} \theta_1 = \frac{-1}{2 * \frac{1}{2}} = -1 \\ \theta_2 = \frac{\sqrt{3}}{2 * \frac{1}{2}} = \sqrt{3} \end{gathered} \tag{9}$$

Because Strong duality holds we can say that f_0 subject to $f_1 < 0$ has minimum at $[-1, \sqrt{3}]$ with value -4

2 Problem 2

2.1 Set Visualization



2.2 Closed Form

Looking at chart above, it's easy so see that all points with $x_2 < 1$ or $x_1 < 1.5$ or $x_2 x_1 \leq 4$ we can use box-constraints. For points between $x_2 = x_1 + 1$ and $x_2 = x_1 - 2$ and above $x_2 = -x_1 + 4$ we can use line projection. In every other case the projection function should return point $(1.5, 2.5)$ or $(3, 1)$ (Simple observation that those points are the nearest to the projection on the line, which makes them the best choice).

$$\pi_x(p) = \begin{cases} a + \frac{(p-a)^T(b-a)}{||b-a||_2^2}(b-a) & (x_2+x_1>4) \wedge (x_2-x_1<1) \wedge (x_2-x_1>-2) \\ (1.5, 2.5) & x_1 > 1.5 \wedge (x_2 - x_1 \geq 1) \\ (3,1) & x_2 > 1 \wedge (x_2 - x_1 \leq -2) \\ min(max(l_i, p_i), u_i) & else \end{cases}$$

where a and b are lay on the $x_2 = -x_1 + 4$,

2.3 Gradient Descent

$$\nabla f(x_1, x_2) = [2x_1 - 4, 4x_2 - 14] \tag{10}$$

$$\begin{gathered} \nabla f(x_0) = (1, -10) \\ p_1 = (2.5, 1) - 0.05(1, -10) = (2.45, 0.5) = x_1 \\ p_2 = (2.45, 0.5) - 0.05(0.9, -12) = (2.405, -0.1) \\ x_2 = \pi_x(p_2) = min(max(l_i, p_i), u_i) = (2.405, 0) \end{gathered} \tag{11}$$

3 Problem 3

3.1 Similarities

- Linear classifiers (can also be used for non-linear classification)
- Binary classifiers

3.2 Differences

- SVM looks for the best possible separation. Where best is understood as the one that represents the largest separation between two classes. While perceptron looks for any separation.
- SVM separation is based only on few points that are are the closest to the separation line. Adding new points the set does not change the classification line (unless new points are "closer"). Perceptron takes all points into the account.

4 Problem 4

SVN has a optimization form of minimizing:

$$f_0(w, b) = \frac{1}{2} w^T w \tag{12}$$

Subject to:

$$f_i(w, b) = y_i(w^T x_i + b) - 1 \geq 0 \tag{13}$$

To show that duality gap is 0 we can use weak Slater's condition which says that strong duality holds when f_0 is convex and f_i are affine.
First is satisfy as quadratic function is convex. Second is satisfy as f_i are affine functions w.r.t w and b.

5 Problem 5

5.1 Matrix Q

Elementwise $g(\alpha) = \sum_i^N \alpha_i - \frac{1}{2}\sum_i^N \sum_j^N \alpha_i \alpha_j y_i y_j x_i^T x_j$ Comparing this with vector from it's clear that Q must contains y and x terms.

Now we define matrix X as a matrix of data points with NxK shape, where K is a number of data dimensions and $i-th$ row contains $i-th$ data point.
We also define matrix Y with KxN shape where each row is the same y vector of data point class, so that i, j element of Y contains class of $j-th$ data point.

Now following is true:

$$(Y^T \odot X)(Y^T \odot X)^T = \sum_i^N \sum_j^N y_i y_j x_i^T x_j \tag{14}$$

Using above and also incorporating $-$ sign from elementwise equation we can deduce that matrix Q can by computed by following operation:

$$Q = -(Y^T \odot X)(Y^T \odot X)^T \tag{15}$$

5.2 Negative semi-definiteness proof

Matrix is negative semi-definite iff $z^T Q z \leq 0$. Using matrix Q form from above we cen rewrite this inequality as follows:

$$-z^T (Y^T \odot X)(Y^T \odot X)^T z \leq 0 \tag{16}$$

Because we multiply quadratic terms in vector notation with $-$ sign at the beginning above is always true, thus Q is negative semi-definite.

5.3 Negative semi-definiteness consequences

In SVN dual optimization we try to maximize $g(\alpha)$. Maximization is much more efficient when function is concave (i.e. Have only one (global) maximum). Thanks to negative semi-definiteness we know that $g(\alpha)$ is concave and maximization problem can be solved in polynomial time (in other case the problem is NP-hard).