Machine Learning 1 - Homework 4

Pascal Mattia Esser

October 11th, 2017

1 Lagrange Multipliers: Warm-up

1.1

1.2

1.3

$$max: x_1 + 2x_2 - 2x_3 \tag{1}$$
$$const: x_1^2 + x_2^2 + x_3^2 = 1 \tag{2}$$

$$L(x_1, x_2, x_3, \lambda) = x_1 + 2x_2 - 2x_3 + \lambda(x_1^2 + x_2^2 + x_3^2 - 1) \tag{3}$$

Constraints:

$$\nabla_{x_1,x_2,x_3,\lambda} L(x_1, x_2, x_3, \lambda) = (2\lambda x_1 + 1, 2\lambda x_2 + 2, 2\lambda x_3 - 2, x_1^2 + x_2^2 + x_3^2 - 1) = 0 \tag{4}$$

we get:$\lambda = -1/(2x_1), x_2 = 2x_1, x_3 = -2x_1, 9x_1^2 = 1$ gives:

$$x_1 = 1/3 \tag{5}$$
$$x_2 = 2/3 \tag{6}$$
$$x_3 = -2/3 \tag{7}$$

which gives us $f(x_1, x_2, x_3) = 3$ or

$$\lambda = 1.5 \tag{8}$$
$$x_1 = -1/3 \tag{9}$$
$$x_2 = -2/3 \tag{10}$$
$$x_3 = 2/3 \tag{11}$$

which gives us $f(x_1, x_2, x_3) = -3$. therefore the fist alternative maximizes the function.

1.4

$$max: 1 - x_1^2 + x_2^2 \tag{12}$$
$$const: -x_1 - x_2 - 1 >= 0 \tag{13}$$

$$L(x_1, x_2, \lambda) = 1 - x_1^2 + x_2^2 + \lambda(-x_1 - x_2 - 1) \tag{14}$$

$$\nabla_{x_1,x_2,\lambda} L(x_1, x_2, \lambda) = (-\lambda - 2x_1, -\lambda - 2x_2, x_1 + x_2 - 1) = 0 \tag{15}$$

we ge $x_1 = x_2$ if $\lambda \neq 0$. we would get that $\lambda = -1$, which could contradict the constrain $\lambda \geq 0$, which gives us:

$$\lambda = 0 \tag{16}$$
$$x_1 = 0 \tag{17}$$
$$x_2 = 0 \tag{18}$$

as the maximized valued. Alternatively:

1.5

$$max: 6x^{2/3}y^{1/2} \tag{19}$$
$$const: 4x + 3y <= 7000 \tag{20}$$

$$L(x, y, \lambda) = 6x^{2/3}y^{1/2} + \lambda(4x + 3y - 7000) \tag{21}$$

$$\nabla_{x,y,\lambda} L(x, y, \lambda) = \left(\frac{4\sqrt{y}}{\sqrt[3]{x}} + 4\lambda, \frac{3x^{2/3}}{\sqrt{y}} + 3\lambda, 4x + 3y - 7000 \right) = 0 \tag{22}$$

we get $\lambda = \frac{z^{1/2}}{x^{1/3}}$. this gives us $x = y$

$$\lambda = \frac{1}{\sqrt{10}} \tag{23}$$
$$x = 1000 \tag{24}$$
$$y = 1000 \tag{25}$$

as the maximized valued.

2 Kernel Outlier Detection

2.1

define the primal program for the circle as: From

$$\forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 \leq R^2 + \xi_i \leq 0, \xi \geq 0 \tag{26}$$

rewrite as:

$$\forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 + \xi_i \leq 0, -\xi \leq 0 \tag{27}$$

primal Lagrangian:

$$L(\boldsymbol{a}, R, \boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = R^2 + C\sum_{i=1}^{N}\xi_i + \sum^{N}\alpha_i\left(||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i\right) - \sum^{N}\mu_i\xi_i \tag{28}$$

2.2

let in general $N_{\setminus z}$ stand for terms that are not depending on z. Use this to shorten the notation for derivatives.

2.2.1

$$\nabla_{R^2}L = \nabla_{R^2}\left(R^2 + \sum^{N}\alpha_i\left(-R^2\right) + N_{\setminus R^2}\right) \tag{29}$$

$$= 1 - \sum^{N}\alpha_i = 0 \tag{30}$$

$$\Leftrightarrow \sum^{N}\alpha_i = 1 \tag{31}$$

2.2.2

$$\nabla_{\boldsymbol{a}}L = \nabla_{\boldsymbol{a}}\left(\sum^{N}\alpha_i\left(||\boldsymbol{x}_i - \boldsymbol{a}||^2\right) + N_{\setminus a}\right) \tag{32}$$

$$= \nabla_{\boldsymbol{a}}\left(\sum^{N}\alpha_i\left((\boldsymbol{x}_i - \boldsymbol{a})^T(\boldsymbol{x}_i - \boldsymbol{a})\right) + N_{\setminus a}\right) \tag{33}$$

$$= \nabla_{\boldsymbol{a}}\left(\sum^{N}\alpha_i\left(\boldsymbol{x}_i^T\boldsymbol{x}_i - 2\boldsymbol{x}_i^T\boldsymbol{a} + \boldsymbol{a}^T\boldsymbol{a}\right) + N_{\setminus a}\right) \tag{34}$$

$$= \sum^{n}\alpha_i(-2\boldsymbol{x}_i + 2\boldsymbol{a}) = 0 \tag{35}$$

$$\Leftrightarrow \sum^{N}\alpha_i\boldsymbol{x}_i = \boldsymbol{a}\sum^{N}\alpha_i \tag{36}$$

Using Equation 31 (37)

$$\sum^{N}\alpha_i\boldsymbol{x}_i = \boldsymbol{a} * 1 = \boldsymbol{a} \tag{38}$$

2.2.3

$$\forall i : \nabla_{\boldsymbol{\xi}}L = \nabla_{\boldsymbol{\xi}}\left(C\sum_{i=1}^{N}\xi_i + \sum^{N}\alpha_i\left(-\xi_i\right) - \sum^{N}\mu_i\xi_i\right) \tag{39}$$

$$C - \alpha_i - \mu_i = 0 \tag{40}$$

$$\Leftrightarrow \forall i : \mu_i = \alpha_i + C \tag{41}$$

2.2.4

KKT for slack term

$$\forall i : \alpha_i \left(||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i \right) = 0 \tag{42}$$

$$\mu_i \xi_i = 0 \tag{43}$$

$$\forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i \leq 0 \tag{44}$$

$$\alpha_i \geq 0 \tag{45}$$

$$\mu_i \geq 0 \tag{46}$$

$$\xi_i \geq 0 \tag{47}$$

2.3

Derive which data-cases xi will have $\alpha_i > 0$ and which ones will have $\mu_i > 0$. NOTE: I solved this question by staring from the given equations Equation 41 to Equation 45 and derive from them the relation of the point to the radius:

For $\alpha_i > 0$: From Equation 42 and Equation 45 follows, that $\alpha_i > 0 \Rightarrow ||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i = 0 \Leftrightarrow ||\boldsymbol{x}_i - \boldsymbol{a}||^2 = R^2 + \xi_i$. from $\alpha_i > 0$ and Equation 41 follows $\mu_i \neq 0 \because \mu_i = \boldsymbol{a}_i + C$ with $\xi_i = 0 \because \mu_i \xi_i = 0$ (Equation 43).

What follows is: $||\boldsymbol{x}_i - \boldsymbol{a}||^2 = R^2$, assuming $C \neq a_i$. meaning that $\boldsymbol{x}_i$ is a support vector and therefore defines the radius.

If we now consider $C = a_i$, ξ can also be $\neq 0$ because $\mu_i \xi_i = 0$ is no longer required to have $\xi_i = 0$ to be fulfilled. This gives us $||\boldsymbol{x}_i - \boldsymbol{a}||^2 \geq R^2$, meaning, the point can be outside of the radius.

For $\mu_i > 0$: $\xi_i = 0 \because \mu_i \xi_i = 0$ (Equation 43). From Equation 44 follows: $||\boldsymbol{x}_i - \boldsymbol{a}||^2 \leq R^2$ meaning that $\boldsymbol{x}_i$ is in or on the radius.

2.4

Derive the dual Lagrangian and specify the dual optimization problem. Kernelize the problem, i.e. write the dual program only in terms of kernel entries and Lagrange multipliers.

$$D_{\boldsymbol{x}}(\alpha,\mu) = \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(R^2 + C\sum_{i=1}^{N}\xi_i + \sum^{N}\alpha_i\left(||\boldsymbol{x}_i-\boldsymbol{a}||^2 - R^2 - \xi_i\right) - \sum^{N}\mu_i\xi_i \right) \tag{48}$$

$$= \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(R^2 + \sum^{N}\alpha_i\left(||\boldsymbol{x}_i-\boldsymbol{a}||^2\right) - \sum^{N}\alpha_i R^2 \right) \tag{49}$$

$$= \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(\sum^{N}\alpha_i\left(\boldsymbol{x}_i^T\boldsymbol{x}_i - \boldsymbol{x}_i^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{x}_i + \boldsymbol{a}^T\boldsymbol{a}\right) \right) \tag{50}$$

$$= \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(\sum^{N}\alpha_i\boldsymbol{x}_i^T\boldsymbol{x}_i - \sum^{N}\alpha_i\boldsymbol{x}_i^T\boldsymbol{a} - \boldsymbol{a}^T\sum^{N}\alpha_i(\boldsymbol{x}_i-\boldsymbol{a}) \right) \tag{51}$$

$$= \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(\sum^{N}\alpha_i k(\boldsymbol{x}_i^T\boldsymbol{x}_i) - \sum_{i}^{N}\alpha_i\boldsymbol{x}_i^T\sum_{j}^{N}\alpha_j\boldsymbol{x}_j \right) \tag{52}$$

$$= \min_{\boldsymbol{a},R,\boldsymbol{\xi}} \left(\sum^{N}\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}_i) - \sum_{ij}^{N}(\alpha_i\alpha_j)(\boldsymbol{x}_i^T\boldsymbol{x}_j) \right) \tag{53}$$

rewriting with a kernel (54)

$$= \sum^{N}\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}_i) - \sum_{ij}^{N}(\alpha_i\alpha_j)k(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{55}$$

2.5

Optimize Values for $\{\alpha_i\}, 0 < \alpha < c$. For this, we have ot maximize over α :

$$\min_{\alpha} D_{\boldsymbol{x}}(\boldsymbol{\alpha},\boldsymbol{\mu}) = \min_{\alpha}\left(\sum^{N}\alpha_i k(\boldsymbol{x}_i\boldsymbol{x}_i) - \sum_{ij}^{N}(\alpha_i\alpha_j)k(\boldsymbol{x}_i\boldsymbol{x}_j) \right) \tag{56}$$

in terms on α_i compute μ_i, use $\alpha_i >= 0$ (Equation 45), $\sum\alpha_i = 1$ (Equation 31) and $\mu_i = C - \alpha_i$ (Equation 41).

Solving for $\boldsymbol{a}^*$ (Equation 38)

$$\sum^{N}\alpha_i\boldsymbol{x}_i = \boldsymbol{a}^* \tag{57}$$

Solve for R: for $\alpha_i > 0 \rightarrow ||\boldsymbol{x}_i - a^*||^2 = R^2$ for this $x_i = 0$ holds. write the radius as:

$$R = ||\boldsymbol{x}_r - \boldsymbol{a}^*|| \tag{58}$$

solve for ξ_i

$$||\boldsymbol{x}_r - \boldsymbol{a}^*||^2 < R \Rightarrow \xi_i = 0 \tag{59}$$

$$||\boldsymbol{x}_r - \boldsymbol{a}^*||^2 > R \Rightarrow \xi_i = ||\boldsymbol{x}_r - \boldsymbol{a}^*||^2 - R \tag{60}$$

2.6

let $\boldsymbol{x}_o$ being an out layer. we need a way to write R as described in the last section for $\boldsymbol{x}_r$, being a support vector.

$$R = ||\boldsymbol{x}_r - \boldsymbol{a}|| \tag{61}$$

Now write an outlayer, $\boldsymbol{x}_o$, as:

$$||\boldsymbol{x}_o - \boldsymbol{a}|| > R \tag{62}$$

$$||\boldsymbol{x}_o - \boldsymbol{a}|| > ||\boldsymbol{x}_r - \boldsymbol{a}|| \tag{63}$$

$$\left(\boldsymbol{x}_o^T\boldsymbol{x}_o - 2\boldsymbol{x}_o^T\boldsymbol{a} + \boldsymbol{a}^T\boldsymbol{a}\right) > \left(\boldsymbol{x}_r^T\boldsymbol{x}_r - 2\boldsymbol{x}_r^T\boldsymbol{a} + \boldsymbol{a}^T\boldsymbol{a}\right) \tag{64}$$

rewrite with kernels and Lagrangian parameters:

$$k(\boldsymbol{x}_o, \boldsymbol{x}_o) - 2\sum_i^N \alpha_i k(\boldsymbol{x}_o^T\boldsymbol{x}_i) + \sum_{ij}^N (\alpha_i\alpha_j) k(\boldsymbol{x}_i^T\boldsymbol{x}_j) \tag{65}$$

$$> k(\boldsymbol{x}_r, \boldsymbol{x}_r) - 2\sum_i^N \alpha_i k(\boldsymbol{x}_r^T\boldsymbol{x}_i) + \sum_{ij}^N (\alpha_i\alpha_j) k(\boldsymbol{x}_i^T\boldsymbol{x}_j) \tag{66}$$

2.7

now let $C = 0$. this means, no penalty for out layers. we can

$$\min_{\alpha,\mu,R} L(\boldsymbol{a}, R, \boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{R}, \boldsymbol{\xi}) \tag{67}$$

if we set $R^2 = 0$

$$\forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i \leq 0, -\xi_i \leq 0 \tag{68}$$

$$\forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 \leq \xi_i \leq 0, -\xi_i \leq 0 \tag{69}$$

Therefore all points are out of the circle. the circle can become a point.

$C = \infty$ means, that all points are in the circle, because

$$\lim_{C\to\infty} \mu_i \Rightarrow \xi_i \to 0 \Rightarrow \forall i : ||\boldsymbol{x}_i - \boldsymbol{a}||^2 \leq R^2 \tag{70}$$

2.8

Gaussian Kernel

$$k(x_i, x_j) = exp\left(-\frac{1}{2\sigma^2}\right) ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \tag{71}$$

smaller bandwidth, so for $\sigma^2 \to 0$ two points the higher the kernel value. Data with less divination is weighted higher. As kernels express a function for similarity, this makes sense, that data with less divination, that are near by have higher 'weights'. Very small σ can lead to over fitting, because the Gaussian kernel is because of its construction likely to create areas around single points, which if they are outlays are over fitted.

2.9

$$\min_{\alpha,\mu,R} R^2 + C\sum_{i}^{N} \xi_i \tag{72}$$

$$\forall i : y_i(||\boldsymbol{x}_i - \boldsymbol{a}||^2 - R^2 - \xi_i) \leq 0, \xi \geq 0 \tag{73}$$

3 Neural Network

3.1

Do the forward pass to calculate the activations of all nodes and then deter- mine the total error of the neural network. Let in the following $\sigma(\cdot)$ be the sigmoid function.

$$\begin{aligned}
z1_{out} &= \sigma(x1 * w1 + x2 * w2) && (74)\\
&= 0.5664790559676278 && (75)\\
z2_{out} &= \sigma(x1 * w3 + x2 * w4) && (76)\\
&= 0.513746534902355 && (77)\\
y_{out} &= \sigma(w_5 * z1_{out} + w_6 * z1_{out}) && (78)\\
&= 0.6295403049912803 && (79)\\
error = E &= -t * \ln(y_{out}) + (1-t)\ln(1-(y_{out})) && (80)\\
&= 0.4627654005556673 && (81)
\end{aligned}$$

3.2

Apply back-propagation to obtain the partial derivatives of the error function with respect to the weights between the hidden layer and the output layer (w5 and w6). Then, use these derivatives to update the weights between the hidden layer and the output layer of the network using stochastic gradient descent (SGD) with a learning rate $\eta = 0.05$

$$\begin{aligned}
w5` &= w5 - lr * \frac{\partial E}{\partial w5} && (82)\\
&= w5 - lr * \frac{\partial E}{\partial y_{out}} \frac{\partial y_{out}}{\partial y} \frac{\partial y}{\partial w5} && (83)\\
&= w5 - lr * (-t/(y_{out}) * y_{out} * (1-y_{out}) * z1_{out}) && (84)\\
&= 0.8095161192315857 && (85)\\
w6` &= w6 - lr * (-t/(y_{out}) * y_{out} * (1-y_{out}) * z2_{out}) && (86)\\
&= 0.16049288291512973 && (87)
\end{aligned}$$

3.3

Further apply back-propagation to obtain the partial derivatives of the error function with respect to the weights between the input layer and the hidden layer (w1, w2, w3 and w4). Again, use SGD to find the updates for these weights. Finally, calculate the total error for the neural network with the updated weights.

$$w1` = w1 - lr * \frac{\partial E}{\partial w1} \tag{88}$$
$$= w1 - lr * \frac{\partial E}{\partial z1_{out}} \frac{\partial z1_{out}}{\partial z1} \frac{\partial z1}{\partial w1} \tag{89}$$
$$\frac{\partial E}{\partial z1_{out}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial z1_{out}} \tag{90}$$
$$\frac{\partial E}{\partial y} = \frac{\partial E}{\partial y_{out}} \frac{\partial y_{out}}{\partial y} \tag{91}$$
$$= -t/(y_{out}) * y_{out} * (1 - y_{out}) \tag{92}$$
$$\frac{\partial y}{\partial z1_{out}} = \frac{\partial (w5 * z1_{out} + w6 * z2_{out})}{\partial z1_{out}} \tag{93}$$
$$= w5 \tag{94}$$
$$\frac{\partial z1_{out}}{\partial z1} = z1_{out}(1 - z1_{out}) \tag{95}$$
$$\frac{\partial z1}{\partial w1} = \frac{\partial (w1 * x1 + w2 * x2)}{\partial w1} \tag{96}$$
$$= x1 \tag{97}$$
$$w1` = w1 - lr * -t/(y_{out}) * y_{out} * (1 - y_{out}) * w5 * z1_{out}(1 - z1_{out}) * x1 \tag{98}$$
$$= 0.4003639107605591 \tag{99}$$
$$\Rightarrow \text{doing this for all other weights gives us:} \tag{100}$$
$$w2` = w2 - lr * -t/(y_{out}) * y_{out} * (1 - y_{out}) * w5 * z2_{out}(1 - z2_{out}) * x1 \tag{101}$$
$$= 0.6503701796760513 \tag{102}$$
$$w3` = w3 - lr * -t/(y_{out}) * y_{out} * (1 - y_{out}) * w6 * z1_{out}(1 - z1_{out}) * x2 \tag{103}$$
$$= 0.2002388164366169 \tag{104}$$
$$w4` = w4 - lr * -t/(y_{out}) * y_{out} * (1 - y_{out}) * w6 * z2_{out}(1 - z2_{out}) * x2 \tag{105}$$
$$= 0.10024293041240866 \tag{106}$$

Apply feed forward again with updated values:

$$z1`_{out} = \sigma(x1 * w1` + x2 * w2`) \tag{107}$$
$$= 0.566519810580869 \tag{108}$$
$$z2`_{out} = \sigma(x1 * w3` + x2 * w4`) \tag{109}$$
$$= 0.5137737411036912 \tag{110}$$
$$y`_{out} = \sigma(w_5` * z1`_{out} + w_6` * z1`_{out}) \tag{111}$$
$$= 0.632059884290427 \tag{112}$$
$$error` = -t * \ln \sigma(y`_{out}) + (1 - t) \ln(1 - \sigma(y`_{out})) \tag{113}$$
$$= 0.4587711357000273 \tag{114}$$

This gives an improve of error of: 0.003994264855639973