Assignment 3 - Support Vector Machines and duality

Machine Learning KU, SS2019

Team Members		
Last Name	First Name	Matriculation Number
Heyer	Yasmine	01431145
Hofer	Florian	01430259

Graz, June 19, 2019

1 Support Vector Regression: the primal problem

1. Figure 1 shows the dataset, which was generated as a swiss-roll dataset.

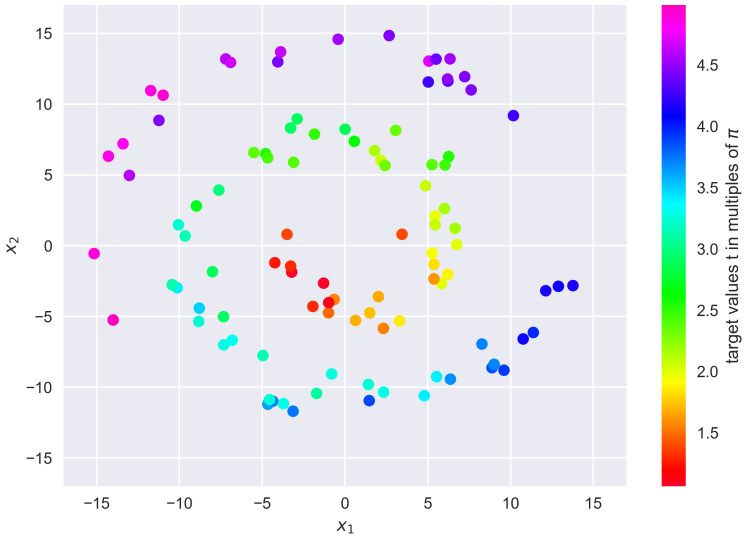


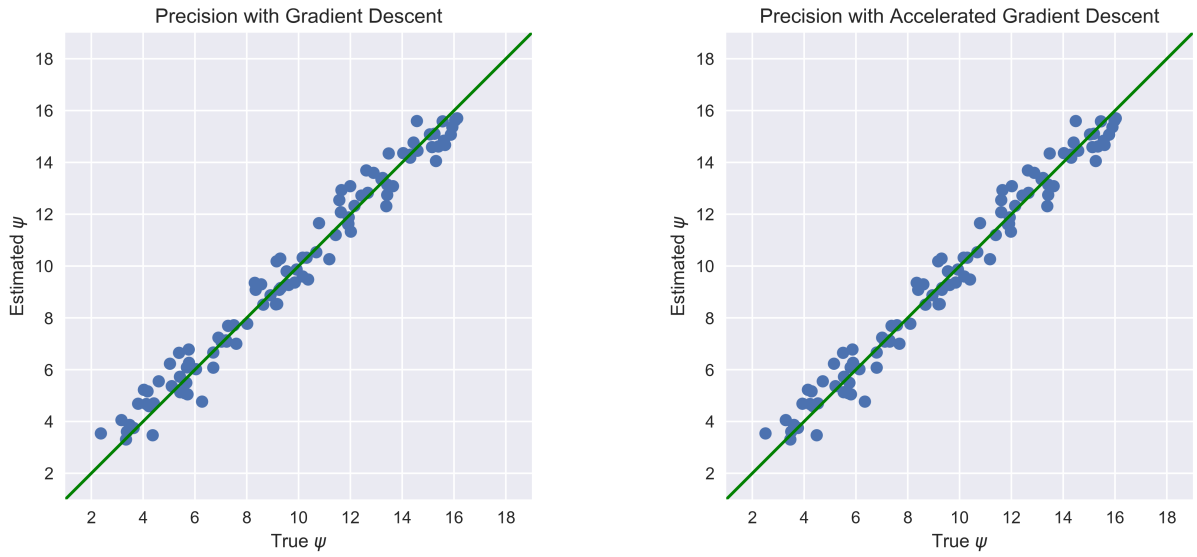
Figure 1: Dataset from the swiss-roll distribution.

The GD algorithm was implemented using the derivation of the Cost Function, which was calculated as follows:

$$\frac{\delta L}{\delta w} = \frac{\delta}{\delta w}\left(\max(0, |w^T\Phi - t| - \epsilon) + \frac{1}{2}||w||^2\right) = \max(0, \Phi \cdot sign(w^T\Phi - t) - \epsilon) + w$$

The termination criterion of the algorithm was set to a tolerance of 10^{-6} between the last two Cost values.
Figure 2 (a) shows the precision of the estimated targets over the true targets. The scatter plot shows that y = x (green line) fits the sample points pretty well and Gradient Descent (GD) therefore results in a good solution for the task.

2. In comparison the Accelerated Gradient Descent (AGD) algorithm was implemented and the precision achieved is shown in Figure 2 (b). The results are comparable to the ones achieved with GD.
Again the termination criterion of the algorithm was set to a tolerance of 10^{-6} between the last two Cost values, enabling direct comparison of the two algorithms.
The iterations needed in order to achieve the following results on the other hand differed for the two algorithms. In Figure 3 the Cost over the iterations for AGD and GD are presented. Both Cost functions converge to similar values (approx. 0.5). The main difference of the algorithms lies in the iterations needed until convergence. AGD's Cost falls steeper and faster and after some oscillations converges rapidly. GD's shows no oscillations but a less steep decline in the graph. For both algorithms the learning rate was set to $\tau = 10^{-5}$.



(a) Testing precision using GD.

(b) Testing precision using AGD.

Figure 2: Testing precision of GD and AGD.

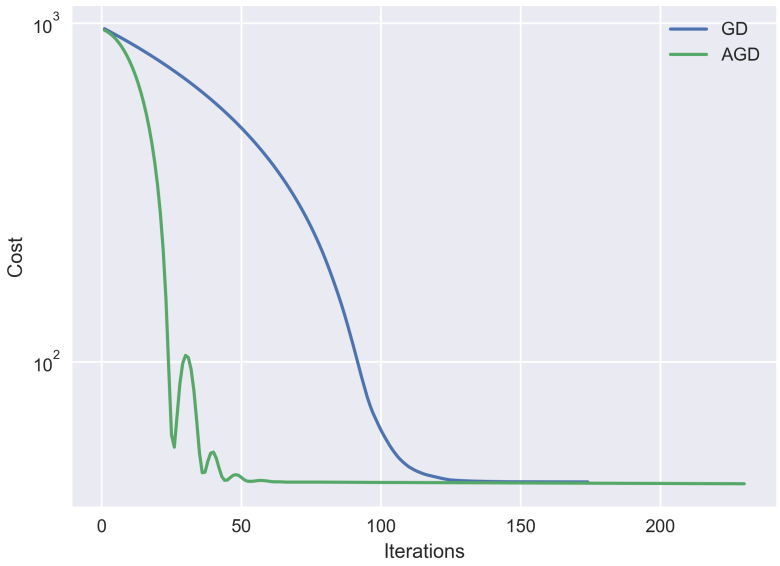


Figure 3: Relative Cost over iterations for AGD and GD.

2 Support Vector Regression: the dual problem

1. The same swiss-role dataset was used for training and testing of the dual problem solved with the projection gradient descent algorithm (PG).
The PG algorithm was implemented using the derivation of the Cost Function, which was calculated as follows:

First we reformulated the cost function to matrix notation:

$$\underset{\boldsymbol{a},\boldsymbol{b}}{\text{maximize}} \; -\frac{1}{2}(\boldsymbol{a}-\boldsymbol{b})^T \cdot \boldsymbol{K} \cdot (\boldsymbol{a}-\boldsymbol{b}) - \epsilon \cdot (\boldsymbol{a}+\boldsymbol{b}) + \boldsymbol{t}^T \cdot (\boldsymbol{a}-\boldsymbol{b})$$

$$\underset{\boldsymbol{a},\boldsymbol{b}}{\text{maximize}} \; -\frac{1}{2}\left(\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{K}\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{K}\boldsymbol{a} + \boldsymbol{b}^T\boldsymbol{K}\boldsymbol{b}\right) - \epsilon\boldsymbol{a} - \epsilon\boldsymbol{b} + \boldsymbol{t}^T\boldsymbol{a} - \boldsymbol{t}^T\boldsymbol{b}$$

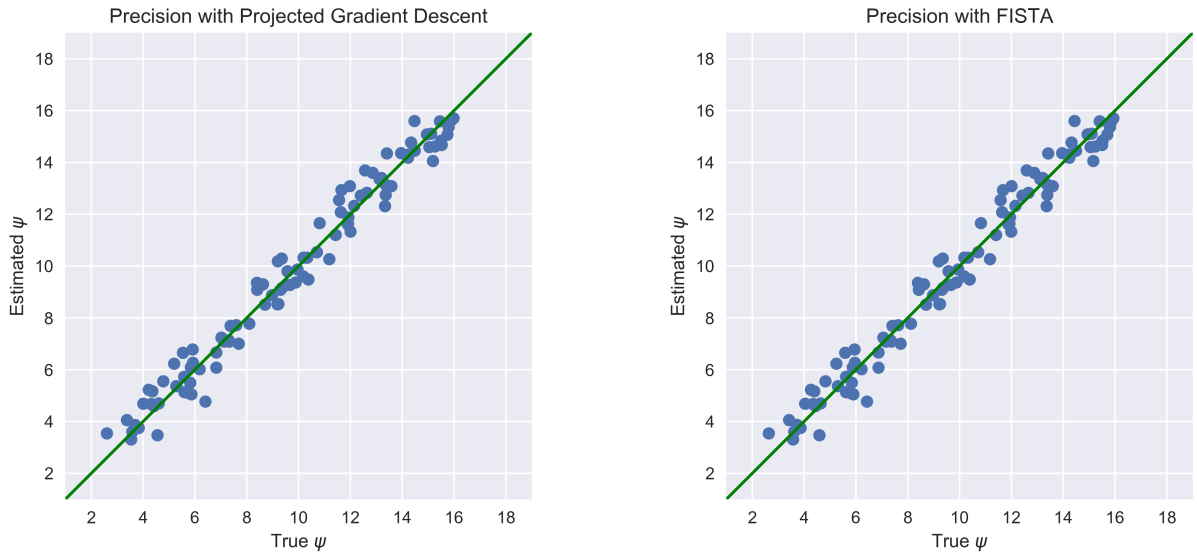
The derivations with respect to $\boldsymbol{a}$ and $\boldsymbol{b}$ then yield (with $\boldsymbol{K} = \boldsymbol{K}^T$):

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{a}} &= -\frac{1}{2}\left((\boldsymbol{K}+\boldsymbol{K}^T)\boldsymbol{a} - \boldsymbol{K}\boldsymbol{b} - \boldsymbol{K}^T\boldsymbol{b}\right) - \epsilon + \boldsymbol{t} \\
&= \boldsymbol{K}\boldsymbol{b} - \boldsymbol{K}\boldsymbol{a} - \epsilon + \boldsymbol{t} \\
\frac{\partial L}{\partial \boldsymbol{b}} &= -\frac{1}{2}\left(-\boldsymbol{K}^T\boldsymbol{a} - \boldsymbol{K}\boldsymbol{a} + (\boldsymbol{K}+\boldsymbol{K}^T)\boldsymbol{b}\right) - \epsilon - \boldsymbol{t} \\
&= \boldsymbol{K}\boldsymbol{a} - \boldsymbol{K}\boldsymbol{b} - \epsilon - \boldsymbol{t}
\end{aligned}$$

As we are dealing with a maximization problem and we wanted to use gradient descent to solve the the dual problem, the gradients had to be multiplied by -1 for the parameter updates.
The termination condition was set to the difference between the last two Cost calculations, which should be smaller than a tolerance of 10^{-10} or the maximum of iterations of 10,000.
Figure 4 (a) shows the precision of the estimated targets of a freshly generated test set over the true targets. PG shows similar results to the previous results obtained with GD and AGD in the precision plot.

2. For the FISTA implemenatation the tolerance of the termination criterion was set to 10^{-15} (lower than for PG) for this algorithm, as the fast convergence and stopping of the algorithm complicated the comparison of the the two algorithms (PG and FISTA). The precision of the FISTA algorithm is shown in Figure 4 (b). Here again the predicted values by FISTA are very similar to the values predicted values obtained by the PG algorithm. For all problems the exact same training and test sets were used in order to enable exact comparison of the different algorithms. This leads to both dual problems resulting in very similar results as the primal problems, when comparing the precision plots in Figure 2 and Figure 4.

The learning rates were again set to $\tau = 10^{-5}$. Looking at the cost in Figure 5, FISTA converges much faster (2000 iterations) than PG (which does not converge in 10,000 iterations). Nonetheless, both provide similar results in the precision plot. To compare the convergence behaviour better, we performed an optimization over 1,000,000 iterations for both PG and FISTA. As the resulting Figure 6 shows, PG needs around 500,000 iterations, but then converges to the same maximum as FISTA.



(a) Testing precision using PG.

(b) Testing precision using FISTA.

Figure 4: Testing precision of PG and FISTA.

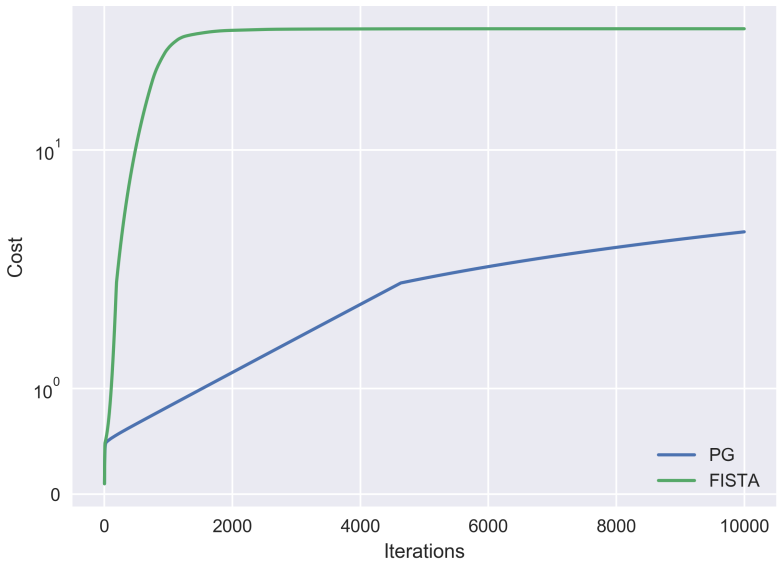


Figure 5: Relative Cost over iterations for PG and FISTA

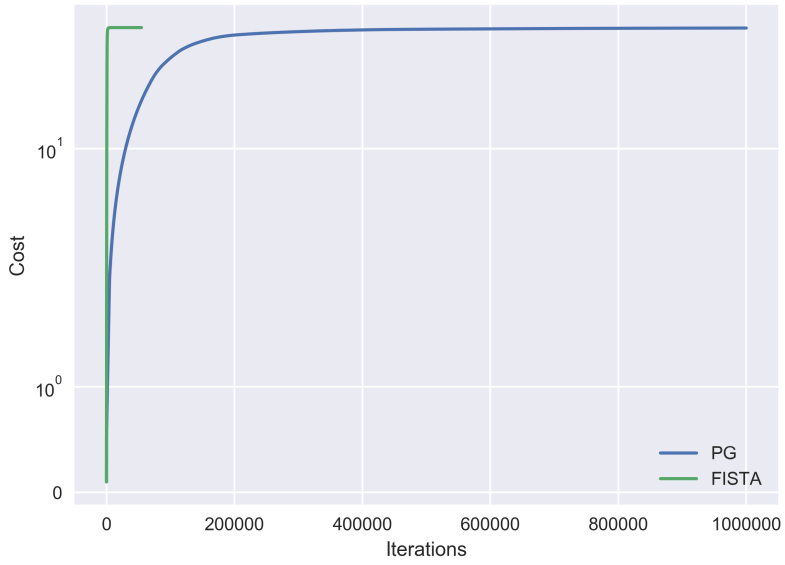


Figure 6: Relative Cost over iterations for PG and FISTA

3 Investigations

Not implemented.