Machine Learning	Prof. Dr. Bernt Schiele
	Dr. Paul Swoboda & Dr. Gerard Pons-Moll
Exercise Sheet 4 - 12.11.2018	due: 19.11.2018, 14:15

Exercise 8 - Projected Gradient Descent for the Lasso

Let $y \in \mathbb{R}^n$ be the n outputs and $\Phi \in \mathbb{R}^{n \times D}$ the design matrix of a regression problem (D basis functions $\phi_1, \ldots, \phi_D$, then $\Phi_{ij} = \phi_j(x_i)$, $i = 1, \ldots, n$, $j = 1, \ldots, D$).

The Lasso problem

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \|Y - \Phi w\|_2^2 + \lambda \|w\|_1 \tag{1}$$

can be rewritten into the following smooth, constrained optimization problem

$$\begin{aligned} \min_{w^+, w^- \in \mathbb{R}^D} \ & \frac{1}{n} \left\|Y - \Phi w^+ + \Phi w^-\right\|_2^2 + \lambda \sum_{i=1}^{D} w_i^+ + \lambda \sum_{i=1}^{D} w_i^- \\ \text{subject to: } \ & w_i^+ \geq 0, \ i = 1, \ldots, D, \\ & w_i^- \geq 0, \ i = 1, \ldots, D. \end{aligned} \tag{2}$$

a. **(3 points)** The projection $P_C : \mathbb{R}^d \to \mathbb{R}$ onto a convex set C is defined for $x \in \mathbb{R}^d$ as

$$P_C(x) := \arg\min_{y \in C} \frac{1}{2} \|x - y\|_2^2 .$$

1. Show that the projection onto a convex set is uniquely defined (Under which condition on the objective is the global minimum unique ?)
2. Derive an analytical expression for the projection onto the convex set

$$C = \{x \in \mathbb{R}^d \,|\, x_i \geq 0\}, \quad \text{(positive orthant in } \mathbb{R}^d\text{)}.$$

b. **(5 points)** Instead of the interior point method introduced in the lecture we use projected gradient descent which is a more simple method for constrained convex optimization. Let C be a convex, closed set and ϕ the differentiable, convex objective function, then the convex optimization problem $\min_{x \in C} \phi(x)$ can be solved via projected gradient descent which is defined as

$$x_{t+1} = P_C(x_t - \alpha_t \nabla \phi(x_t)),$$

where $\alpha_t > 0$ is the stepsize.

1. Complete the Matlab Function `Lasso` which has as arguments Y, Φ, λ and returns the weight vector w of the Lasso problem in Equation (1). Use projected gradient descent for the optimization problem (2).
2. Run your Lasso implementation with linear design (no offset) for the training data from the last exercise sheet (prediction of crime rate of U.S. cities) with $\lambda = 10^{-3}$. You should get 31 non-zero coefficients in the optimal weight vector.

   i. What is the influence of a feature with positive resp. negative component of the weight vector ?

   ii. Check the corresponding information on the features (see FeatureInformation file) to see if the features chosen by Lasso (corresponding to non-zero components in the weight vector) make actually sense for this problem

Both parts should be answered on paper. The last question should be answered briefly (just check the 10 largest (in absolute value) components of the optimal weight vector).

Hints:

- A sum $f + g$ of convex functions f, g is strictly convex if f or g is strictly convex.
- For the computation of the projection, note that each component can be minimized independently of the other ones. Why ?

Solution:

a.
- The global minimum of a convex function f is unique if the function f is strictly convex, that is if $x \neq y$,

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y), \quad \forall \lambda \in]0,1[.$$

What remains to show is that $f(x) = \|y - x\|_2^2 = \|y\|_2^2 + \|x\|_2^2 + 2\langle x, y\rangle$ is strictly convex. A sum of convex functions is strictly convex if already one of them is strictly convex. Thus we check $g(x) = \|x\|_2^2$. Since $\lambda \geq 0$ and $(1-\lambda) \geq 0$, we get

$$\begin{aligned}
\|\lambda x + (1-\lambda)y\|_2^2 &= \lambda^2 \|x\|_2^2 + 2\lambda(1-\lambda)\langle x, y\rangle + (1-\lambda)^2 \|y\|_2^2 \\
&\leq \lambda^2 \|x\|_2^2 + 2\lambda(1-\lambda)\|x\|_2 \|y\|_2 + (1-\lambda)^2 \|y\|_2^2 \\
&\leq \lambda^2 \|x\|_2^2 + \lambda(1-\lambda)(\|x\|_2^2 + \|y\|_2^2) + (1-\lambda)^2 \|y\|_2^2 \\
&= \lambda \|x\|_2^2 + (1-\lambda)\|y\|_2^2,
\end{aligned}$$

where in the last step we used $\lambda + (1-\lambda) = 1$. So far we just derived that g is a convex function. We used two inequalities

$$\langle x, y\rangle \leq \|x\|_2 \|y\|_2, \qquad \text{and} \qquad 2\|x\|_2 \|y\|_2 \leq \|x\|_2^2 + \|y\|_2^2.$$

For Cauchy-Schwarz inequality, equality is achieved if and only if $x = \alpha y$ for some $\alpha \geq 0$. For the second inequality, equality is achieved if and only if $\|x\| = \|y\|$ (follows from $(\|x\|_2 - \|y\|_2)^2 = 0$). Putting this together we have

$$\|x\| = \|\alpha y\| = |\alpha| \, \|y\|,$$

and thus with the last equality we have $|\alpha| = \alpha = 1$. Thus we have equality for any value $\lambda \in [0,1]$ if and only if $x = y$. If $x \neq y$ we have strict inequality by the same argument if $\lambda(1-\lambda) > 0$ which is the case if $\lambda \in]0,1[$.

- We have to solve

$$\underset{y_i \geq 0,\, i=1,\ldots,d}{\arg\min} \quad \frac{1}{2}\sum_{i=1}^{d}(x_i - y_i)^2.$$

Note that in the objective there is no coupling of the variables (the objective is **separable**). Moreover, also the constraints do not couple the variables, thus we can solve for each component individually,

$$\underset{y_i \geq 0}{\arg\min} \quad \frac{1}{2}(x_i - y_i)^2.$$

If $x_i \geq 0$ we minimize the objective by setting $y_i = x_i$. In the other case where $x_i < 0$, the objective is minimized by setting $y_i = 0$. Thus the solution is

$$P_C(x)_i = y_i = \max\{0, x_i\}, \quad i = 1, \ldots, d.$$

b. 1. The code for projected gradient descent:

```
function wstar = LassoSolution(Y,Phi,lambda)
% input:
% Y    - outputs (column vector of size n)
% Phi  - design matrix (size n times D)
% lambda - regularization parameter (>=0)
%
% output:
% w               - optimal weight vector (size D, column vector) of the Lasso

[num,dim]=size(Phi);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THIS IS NOT THE MOST EFFICIENT WAY TO IMPLEMENT PROJECTED GRADIENT
% DESCENT !
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% in the transformed optimization problem you have two variables wplus and
% wneg - both are restricted to be positive

wplus=rand(dim,1); % the old iterate x_t
wneg =rand(dim,1);

wplusnew=zeros(dim,1); % this is the new iterate (x_{t+1})
wnegnew=zeros(dim,1);
counter=1;
% stopping criterion is here the norm of difference of two iterates
while( sqrt(norm(wplus-wplusnew)^2 + norm(wneg-wnegnew)^2)>1E-7)

  wplus = wplusnew;
  wneg  = wnegnew;

  % compute the gradient
  [gradplus,gradneg]=GradLassoObjective(Y,Phi,lambda,wplus,wneg);

  % get stepsize
  stepsize = getStepSize(Y,Phi,lambda,wplus,wneg,gradplus,gradneg);

  % projected gradient steps
  wplusnew = ProjectionPositiveOrthant(wplus - stepsize*gradplus);
  wnegnew = ProjectionPositiveOrthant(wneg - stepsize*gradneg);

  if(rem(counter,10)==0)
    Obj = LassoObjective(Y,Phi,lambda,wplusnew,wnegnew);
    disp(['Iteration: ',num2str(counter),' - Current Objective: ',num2str(Obj,'%1.12f
        '),' - stepsize: ',num2str(stepsize)]);
  end
  counter=counter+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% return the weight vector for the original Lasso problem
wstar = wplus-wneg;









%%%%%% the functions described in the exercise %%%%%%%%%%%%%%%%%%

function Projw = ProjectionPositiveOrthant(w) % returns the projection of w onto the
    positive orthant
  Projw = max(w,0);

function [gradplus,gradneg] = GradLassoObjective(Y,Phi,lambda,wplus,wneg) % returns
    the gradient of the objective at (wplus,wneg)
  [num,dim]=size(Phi);

  firstterm = 2/num*Phi'*(Y-Phi*(wplus-wneg));

  gradplus = -firstterm + lambda*ones(dim,1);
  gradneg  =  firstterm + lambda*ones(dim,1);

function fval = LassoObjective(Y,Phi,lambda,wplus,wneg) % returns the objective of
    the optimization problem given wplus and wneg
  [num,dim]=size(Phi);
  fval = 1/num*norm(Y-Phi*(wplus-wneg))^2 + lambda*sum(wplus) + lambda*sum(wneg);

function stepsize=getStepSize(Y,Phi,lambda,wplus,wneg,gradplus,gradneg)  % given the
```

```
    current points and their gradients returns the stepsize
78  stepsize=0.5; beta = 0.5;
79  objective = LassoObjective(Y,Phi,lambda,wplus,wneg);
80  wplusnew = ProjectionPositiveOrthant(wplus - stepsize*gradplus);
81  wnegnew  = ProjectionPositiveOrthant(wneg - stepsize*gradneg);
82  newobjective = LassoObjective(Y,Phi,lambda,wplusnew,wnegnew);
83
84  % stepsize selection via backtracking line search
85  % (specific for projected gradient descent)
86  while( newobjective > objective + gradplus'*(wplusnew-wplus)+gradneg'*(wnegnew-wneg
        )+1/(2*stepsize)*(norm(wplusnew-wplus)^2+norm(wnegnew-wneg)^2))
87   stepsize=beta*stepsize;
88   wplusnew = ProjectionPositiveOrthant(wplus - stepsize*gradplus);
89   wnegnew  = ProjectionPositiveOrthant(wneg - stepsize*gradneg);
90   newobjective = LassoObjective(Y,Phi,lambda,wplusnew,wnegnew);
91  end
```

2. i. Features with positive weights have positive influence on the crime rate, whereas features with negative weight have negative influence on the crime rate

   ii. The 10 features with the largest components (in absolute value) and the associated feature description:

Weight	Feature Description	
51	0.269	percentage of kids born to never married
3	0.189	percentage of population that is african american
39	0.185	percentage of males who are divorced
69	0.131	percent of persons in dense housing (more than 1 person per room)
72	0.126	number of vacant households
91	0.104	number of homeless people counted in the street
45	-0.090	percentage of kids in family housing with two parents
73	-0.053	percent of housing occupied
89	-0.050	median owners cost as a percentage of household income for owners without a mortgage
87	0.046	median gross rent as a percentage of household income

   Apart from the last two the chosen features make roughly sense. It is also a bit unclear, why the percentage of kids born to never married should have such a huge influence.
   This dataset turned out to be only of partial use. The problem is that through the normalization done and in particular by clipping extreme values the interpretation of the results becomes difficult.

Exercise 9 - Derivation of a dual problem

Let $(x_i, y_i)_{i=1}^n$ be a training sample for a binary classification task, that is $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. The so-called hard-margin Support Vector Machine (SVM) without offset corresponds to the optimization problem

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & \frac{1}{2} \|w\|_2^2 \\ \text{subject to :} \quad & y_i \langle w, x_i \rangle \geq 1, \qquad i = 1, \ldots, n \end{aligned}$$

a. **(3 Points)** Derive the dual problem.

b. **(1 Point)** Which problem, dual or primal, would you solve depending on n (number of training samples) versus d (number of features) ?

Hints:

- Note that inequality constraints have the form $g(x) \leq 0$

Solution:

a. The Lagrange function is

$$L(w, \lambda) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i \langle w, x_i \rangle).$$

The Lagrange function is convex in w and thus a necessary and sufficient condition for the global minimum is

$$\nabla_w L(w, \lambda) = 0 \quad \Longrightarrow \quad w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0,$$

and thus $w^* = \sum_{i=1}^{n} \lambda_i y_i x_i$. We plug this form of the weights into the Lagrange function in order to derive the dual function $q(\lambda)$,

$$\begin{aligned} q(\lambda) &= \frac{1}{2} \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 + \sum_{i=1}^{n} \lambda_i - \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 \\ &= -\frac{1}{2} \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 + \sum_{i=1}^{n} \lambda_i \end{aligned}$$

The dual problem is now

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^n} \quad & -\frac{1}{2} \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 + \sum_{i=1}^{n} \lambda_i \\ \text{subject to :} \quad & \lambda_i \geq 0, \qquad i = 1, \ldots, n \end{aligned}$$

b. The primal problem has d variables and n constraints while the dual problem has n variables and n constraints. If $d \ll n$, then it makes sense to solve the primal problem. If $d \gg n$, solving the dual problem is better, especially the constraints have much simpler form in the dual.

Note that if one solves the dual problem, then the prediction of a new test input x is computed using the dual solution λ as

$$\langle w, x \rangle = \sum_{i=1}^{n} \lambda_i y_i \langle x_i, x \rangle$$

That is one has to compute n inner products which seems to be expensive. As we will see in the lecture many of λ_i's would be zero and hence one needs to compute only fewer inner products. Moreover, the dual has additional advantage that the problem as well as the formula for the prediction is expressed in terms of inner products between the training inputs. This allows one to introduce new similarity measure (kernel) between training points (especially when the inputs are not vectorial data).