Lasso and Optimality

Alain Rakotomamonjy
alain.rakotomamonjy@univ-rouen.fr, sites.google.com/site/alainrakotomamonjy/home

Practical session description

This practical session aims at writing functions solving the sparse linear regression problem that occurs in variable selection or in compressed sensing problems.

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

To make it work, you are supposed to have CVX installed (you can download it from `http://cvxr.com/cvx/`)

Ex. 1 — Sparsity in linear regression

1. Building the signal to approximate
   a) Build the covariate matrix $\mathbf{X}$: Generate a set of $n = 200$ data points in dimension $2n$, randomly distributed on a Gaussian hypersphere

```
n = 200;
d = 2*n;
X=randn(n,d);
X=X./(ones(n,1)*sqrt(sum(X.^2)));
```

   b) From this covariate matrix $\mathbf{X}$, we build a toy signal $\mathbf{y} = \mathbf{X}\mathbf{w}^\star$ with T non-zero elements in $\mathbf{w}^\star$

```
%create signal
T=5;
rsnr=30;
ind=randperm(size(X,2));
indice=ind(1:T);
weights=randn(T,1);
weights=weights + 0.1*sign(weights);
y=X(:,indice)*weights;
stdnoise=std(y)/rsnr;
y=y+randn(size(y)).*(ones(n,1)*stdnoise);
wopt=zeros(d,1);
wopt(indice)=weights;
```

2. Solving the optimization problem
   a) Implement in CVX the optimization problem above

```
lambda=0.1;
% Create and solve problem in CVX
lambda=0.1 ;
cvx_quiet(true)
cvx_begin
    variable wcvx(d)
    minimize( 0.5*(X *wcvx - y)'*(X*wcvx-y) + lambda*sum(abs(wcvx)))
cvx_end
```

   b) Check if the problem has been properly solved by verifying the optimality conditions

$$\begin{aligned} sign(w_j^\star) \neq 0 &\implies -\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\mathbf{w}^\star) + \lambda sign(w_j) = 0 \\ sign(w_j^\star) = 0 &\implies |\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\mathbf{w}^\star)| \leq \lambda \end{aligned}$$

```
epsi=1e-6;
indzero=find(abs(wcvx)<epsi);
indnonzero=find(abs(wcvx)>=epsi);
grad=-X'*(y-X*wcvx);
exactOnZeros= max(abs(grad(indzero))-lambda)
exactOnNonZeros= max(abs(grad(indnonzero) + lambda*sign(wcvx(indnonzero))))
```

For optimality, exactOnZeros should be negative or smaller than a threshold and exactOnNonZeros should be smaller than a threshold.

c) compare the amplitudes of retrieved weights for the true non-zeros elements

```
plot(wcvx - wopt)
```

d) play with λ and compare $\mathbf{w}$ with $\mathbf{w}^\star$ in terms of support and difference of amplitudes. The comparison should hold for the same matrix $\mathbf{X}$ and signal $\mathbf{y}$.

```
% count the number of non-zero vectors
nbnonzeros=length(indnonzero);
% compute the difference
 maxi=max(abs(wcvx(indice)-wopt(indice)));
fprintf('lambda:%2.2f T:%d nbnonzeros:%d diff:%2.3f \n',lambda,T,nbnonzeros,maxi);
```

3. Implementing a coordinatewise approach

a) for the same problem as above, implement a coordinatewise approach that goes through the full data 1000 times

```
tic
lambda=0.1;
w=zeros(d,1);
for i=1:1000
    for k=1:d
        xk=X(:,k);
        wtemp=w;
        wtemp(k)=0;
        s=y-X*wtemp;
        w(k)= sign(xk'*s)*max(0,abs(xk'*s) - lambda)/(xk'*xk);
    end;
end;
cdtiming=toc
```

b) Check if the solution is correct through the optimality conditions

```
 epsi=1e-6;
 indzero=find(abs(w)<epsi);
 indnonzero=find(abs(w)>=epsi);
 grad=-X'*(y-X*w);
 exactOnZeros= max(abs(grad(indzero))-lambda)
 exactOnNonZeros= max(abs(grad(indnonzero) + lambda*sign(wcvx(indnonzero))))
```

c) Write a function `CDsparseRegression` that implements a coordinatewise descent algorithm that stops when a stopping criterion based on the optimality conditions is reached. Note that the number of iterations can be allowed to be larger than 1000 since we have another criterion to stop.

```
function w=CDsparseRegression(X,y,lambda,epsi)
```

hint: compute the optimality conditions for the zeros and non-zeros w_i for all i and exit the outer loop when the condition is lower that a threshold `epsi`.

d) compare the computational running time of coordinatewise descent, CVX and Cplex (or Gurobi) when $n = 500$, $d = 1000$, and for different values of λ and T. Discuss the results