Solving the solution path of OSCAR with "genlasso"

Solution path of OSCAR described in Example 4

The matrix $X \in \mathbb{R}^{2\times 3}$ and the vector $y \in \mathbb{R}^2$ given in Example 4 are reported hereafter:

```
X=matrix(nrow=2,ncol=3,c(2,1,1,2,0,1))
y=c(15,5)
```

We construct the matrix D for the generalized LASSO for which $\|Db\|_1 = 6|b|_{\downarrow 1} + 4|b|_{\downarrow 2} + 2|b|_{\downarrow 3}$.

```
D=matrix(nrow =9, ncol=3)
D[1,]=c(2,0,0)
D[2,]=c(0,2,0)
D[3,]=c(0,0,2)
D[4,]=c(1,1,0)
D[5,]=c(1,-1,0)
D[6,]=c(1,0,1)
D[7,]=c(1,0,-1)
D[8,]=c(0,1,1)
D[9,]=c(0,1,-1)
```

We use the package genlasso to solve the solution path of OSCAR

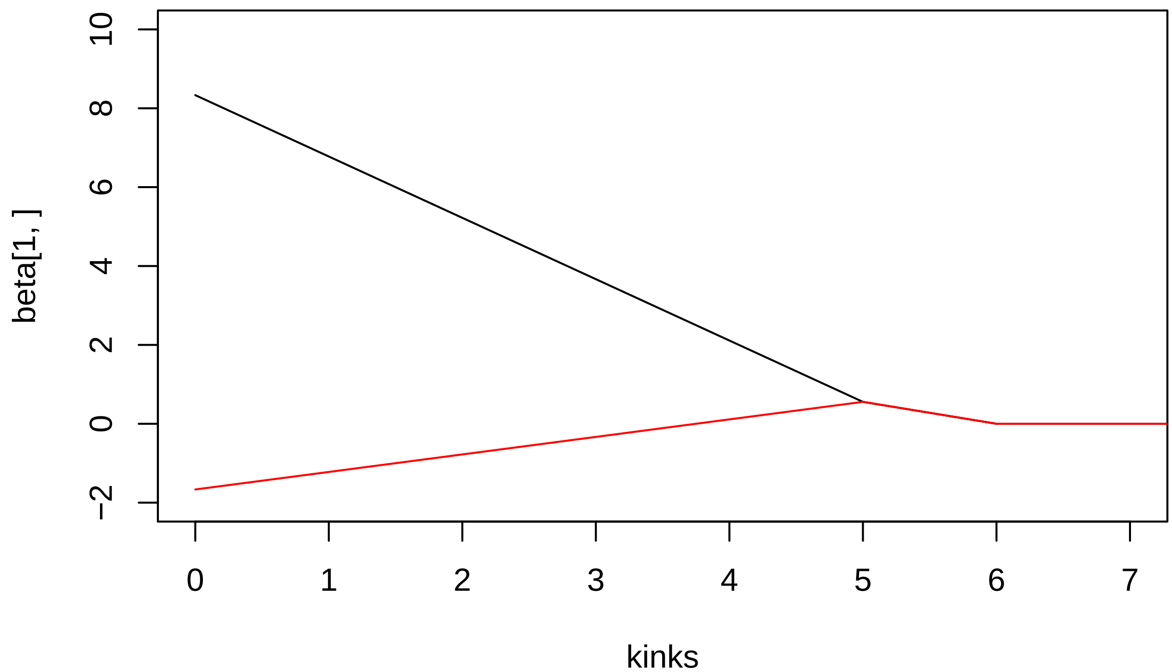
```
library(genlasso)
```

```
## Loading required package: Matrix

## Loading required package: igraph

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
Sol = genlasso(y,X,D,approx=FALSE)
```

```
## Warning in genlasso(y, X, D, approx = FALSE): Adding a small ridge penalty
## (multiplier 0.0001), because X has more columns than rows.
```

```r
kinks = Sol$lambda
beta = Sol$beta
plot(kinks,beta[1,],type="l",xlim=c(0,7),ylim=c(-2,10))
lines(kinks,beta[2,],col=2)
```



Some comments on the solution path provided by this package:

Actually 6 and 5 are true kinks of the OSCAR solution path but other kinks $8.75, 7.92, 6.87, 6.75, 6.42, 6.17, 0.0001$ are not relevant. Moreover some true kinks are missing: $3.75, 0.42, \ldots$.

The solution of OSCAR at $\gamma \in \{8.75, 7.92, 6.87, 6.75, 6.42, 6.17, 6, 5\}$ are correct but wrong when 0.0001.

Overall, the solution path of OSCAR is correct when $\gamma \geq 5$ but wrong when $\gamma < 5$.

Solution path of OSCAR applied on the red Vinho Verde wines data set

The data set can be downloaded from http://archive.ics.uci.edu/ml/datasets/Wine+Quality. Hereafter we standardize these data.

```r
library(radarBoxplot)
data("winequality_red")
y=winequality_red[,12]-mean(winequality_red[,12])
X=matrix(nrow=1599,ncol=11)
for (j in (1:11))
{
  X[,j]=sqrt(1599/1598)*(winequality_red[,j]-mean(winequality_red[,j]))/sd(winequality_red[,j])
}
```

We construct the matrix D for the generalized LASSO for which $\|Db\|_1 = 4|b|_{\downarrow 1} + \cdots + |b|_{\downarrow 11}$.

```
p=ncol(X)
I=diag(p)
A=matrix(nrow=p*(p-1),ncol=p,rep(0,p^2*(p-1)))

i=1
for ( k in (1:(p-1)))
{
  for ( l in ((k+1):p))
  {
    A[i,k]=1/2
    A[i+1,k]=1/2
    A[i,l]=1/2
    A[i+1,l]=-1/2
    i=i+2
  }
}
D=rbind(I,0.3*A)
```

The running time to solve the OSCAR solution path, numerical minima when $\gamma \in \{J^*_\lambda(X'y)/2, J^*_\lambda(X'y)/10\}$ and a graphical illustration of this path are reported below

```
# Running time
t_in = Sys.time()
Sol=genlasso(y,X,D)
t_out = Sys.time()
cat("The running time is:",  t_out - t_in, "\n")
```

```
## The running time is: 0.5078666
```

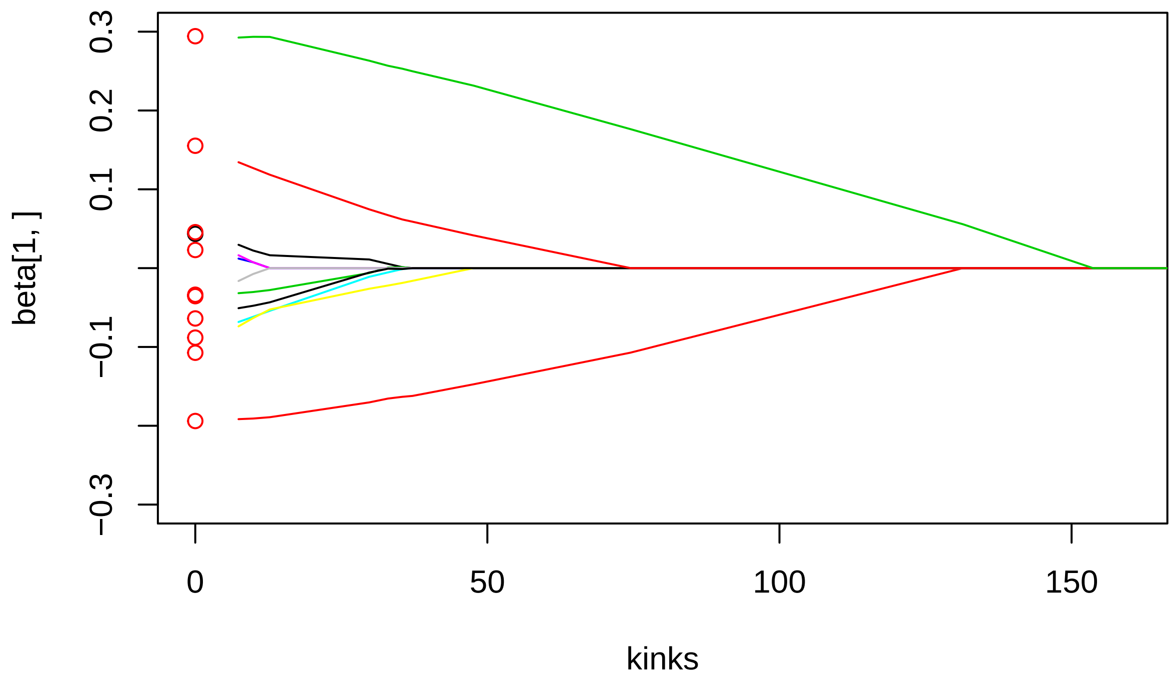
```
# Graphical illustration
ols = solve(t(X)%*%X,t(X)%*%y)
kinks = Sol$lambda
beta = Sol$beta
finished = Sol$completepath
cat("Does genelasso compute entirely the path?",  finished, "\n" )
```

```
## Does genelasso compute entirely the path? TRUE
```

```
cat("Smallest kink:",min(kinks), "\n")
```

```
## Smallest kink: 7.404376
```

```
plot(kinks,beta[1,],type="l",xlim=c(0,160),ylim=c(-0.3,0.3))
points(0,ols[1])
for (i in (2:11))
{
lines(kinks,beta[i,],col=i)
points(0,ols[i],col="red")
}
```




```
# Minima

gamma_max=153.623
gamma1=gamma_max/2
gamma2=gamma_max/10

alpha1 = (gamma1-kinks[60])/(kinks[59]-kinks[60])
beta1 = beta[,59]*alpha1+(1-alpha1)*beta[,60]
min1 =  0.5*sum((y-X%*%beta1)^2)+gamma1*sum(abs(D%*%beta1))
cat("Minimum at J*(X'y)/2:",min1 ,"\n")
```

```
## Minimum at J*(X'y)/2: 483.4111
```

```
alpha2 = (gamma2-kinks[114])/(kinks[113]-kinks[114])
beta2 = beta[,113]*alpha2+(1-alpha2)*beta[,114]
min2 =  0.5*sum((y-X%*%beta2)^2)+gamma2*sum(abs(D%*%beta2))
cat("Minimum at J*(X'y)/10:",min2 , "\n")
```

```
## Minimum at J*(X'y)/10: 379.8436
```

Some comments on the solution path provided by this package:

Actually $\gamma = 7.40$ is not the smallest kink of OSCAR solution path; there are 7 kinks smaller than 7.40 (the smallest kink is 0.26). Surprisingly, despite genlasso does not solve entirely the solution path, this algorithm claims that the path is entirely computed.