ISTA

March 27, 2023

[19]:
```
# ISTA, FISTA and IRLS example

using LinearAlgebra, PyPlot, Printf, SparseArrays, FFTW, SeisProcessing,␣
 ↪SeisMain,SeisReconstruction
using BenchmarkTools
```

[21]:
```
include("fista.jl")
include("ista.jl")
include("power_iteration.jl")
include("irls.jl")
```

[21]:
```
IRLS (generic function with 1 method)
```

Below is the main script where I test ISTA, FISTA and IRLS for a simple CS problem. A random matrix of size N x M compresses a sparse signal x of lenght M via $y = Ax + e$. The observation vector y is of lenght $N < M$. We consider observations contaminated with errore e. Then we try to recover x via sparse solvers. In other words, we estimate x by minimizing the $l_2 - l_1$ cost function J given by

$$J = \|Ax - y\|_2^2 + \lambda \|x\|_1$$

The cost function above is minimized via ISTA, FISTA and IRLS. In the examples below, we use the same value of λ and the same maximum iteration number for the three tested methods. ISTA and FISTA apply one matrix $A[]$ and $A^T[]$ per iteration. IRLS in each iteration solves a linear system of equations. The IRLS is more expensive per iteration than ISTA and FISTA but in my example seems to converge much faster.

[32]:
```
# Example y = A x + e, with x sparse and A a random matrix
# Problem is underdetemined M>N,where size of A is N x M.


N = 40
M = 150
A = randn(N,M)


xt = (Array(sprandn(M,1,0.12)))             # True
```

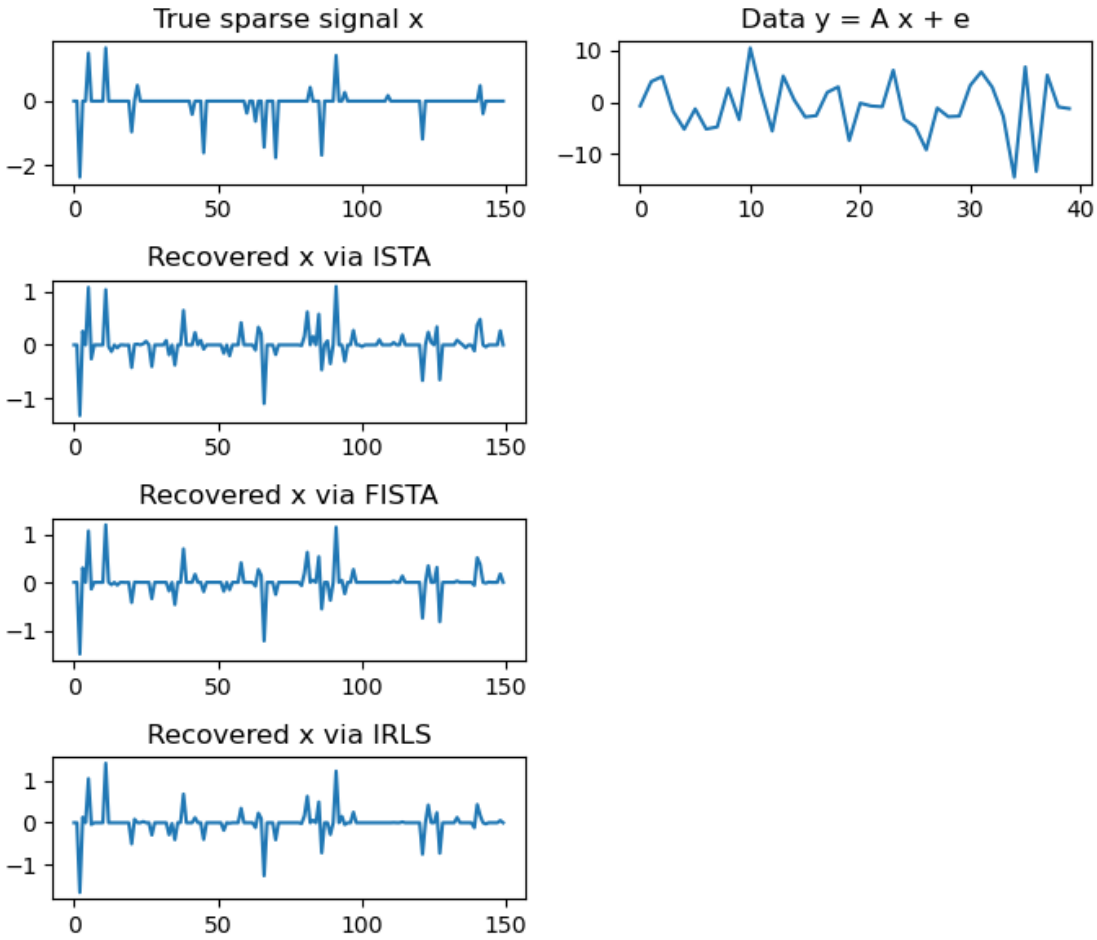
```
y = A*xt                                        # Compressed signal y (observations)
e = 0.0001*maximum(y)*randn(size(y))            # Add noise to observations
y = y + e

  = 1.0                                         # Trade-off parameter

NITER = 500                                     # Maximum number of iterations
```

[32]: 500

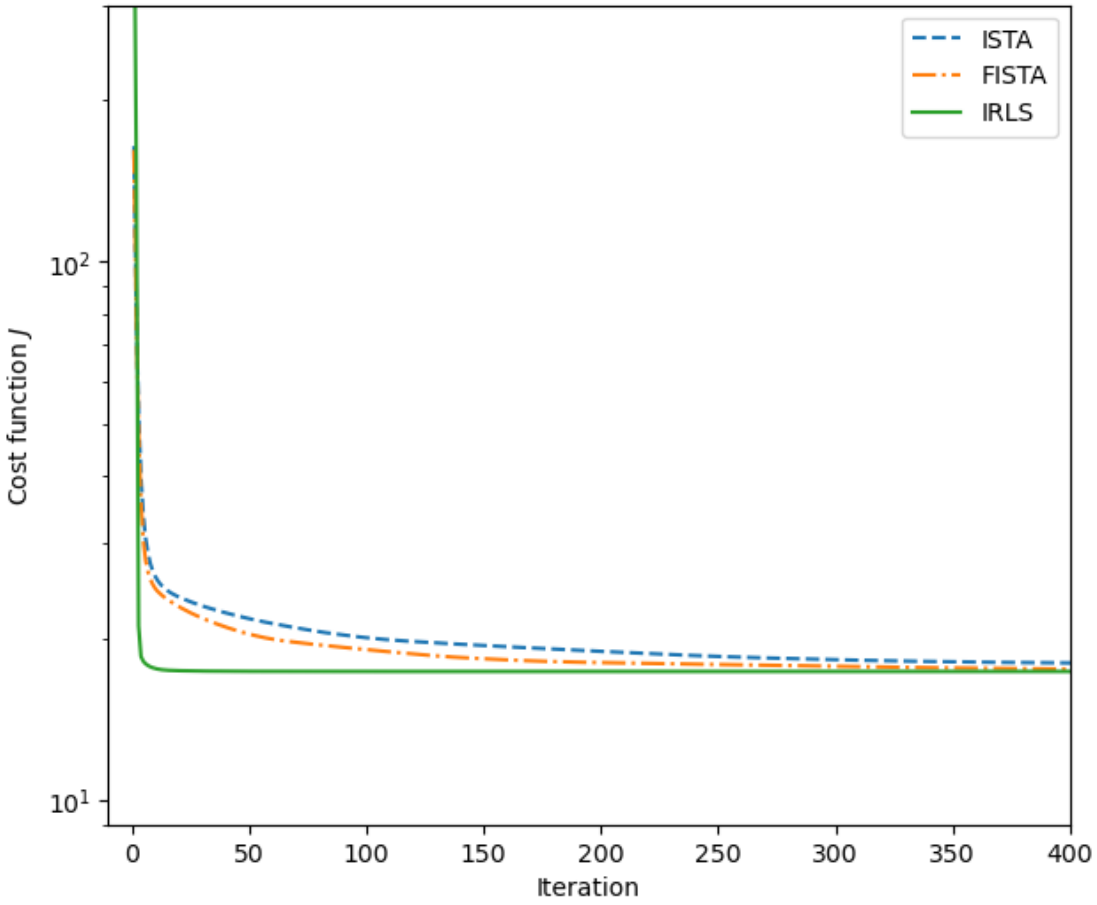
[38]:
```
x1, J1 =  ISTA(A, y, NITER,  )
x2, J2 = FISTA(A, y, NITER,  )
x3, J3 =  IRLS(A, y, NITER,  );
```

[34]:
```
figure(1,figsize=(7,6))

subplot(421); plot(xt);  title("True sparse signal x")
subplot(422); plot(y);   title("Data y = A x + e")
subplot(423); plot(x1);  title("Recovered x via ISTA")
subplot(425); plot(x2);  title("Recovered x via FISTA")
subplot(427); plot(x3);  title("Recovered x via IRLS")


tight_layout()
savefig("fig_1_comparison.pdf")
savefig("fig_1_comparison.PNG")
rmse(a,b) = sum((a-b).^2)/sum(b.^2)

@printf "rmse for ISTA    %0.12e  \n" rmse(x1,xt);
@printf "rmse for FISTA   %0.12e  \n" rmse(x2,xt);
@printf "rmse for IRLS    %0.12e  \n" rmse(x3,xt);
```

```
rmse for ISTA    4.973084100947e-01
rmse for FISTA   4.520282715773e-01
rmse for IRLS    3.569152205720e-01
```



[39]:
```
figure(2,figsize=(7,6))
k=collect(1:1:length(J1))
        semilogy(k,J1,"--",k,J2,"-.",k,J3)
axis([-10,400,9,300])
legend(["ISTA", "FISTA", "IRLS"])
minimum(J1)
xlabel("Iteration")
ylabel(L"Cost function $J$")
savefig("convergence.pdf")
savefig("convergence.PNG")
```



[30]:
```
size(J3)
```

[30]: (1000,)