Demo_3_Lapis_2019_Steepest_Descent

April 8, 2019

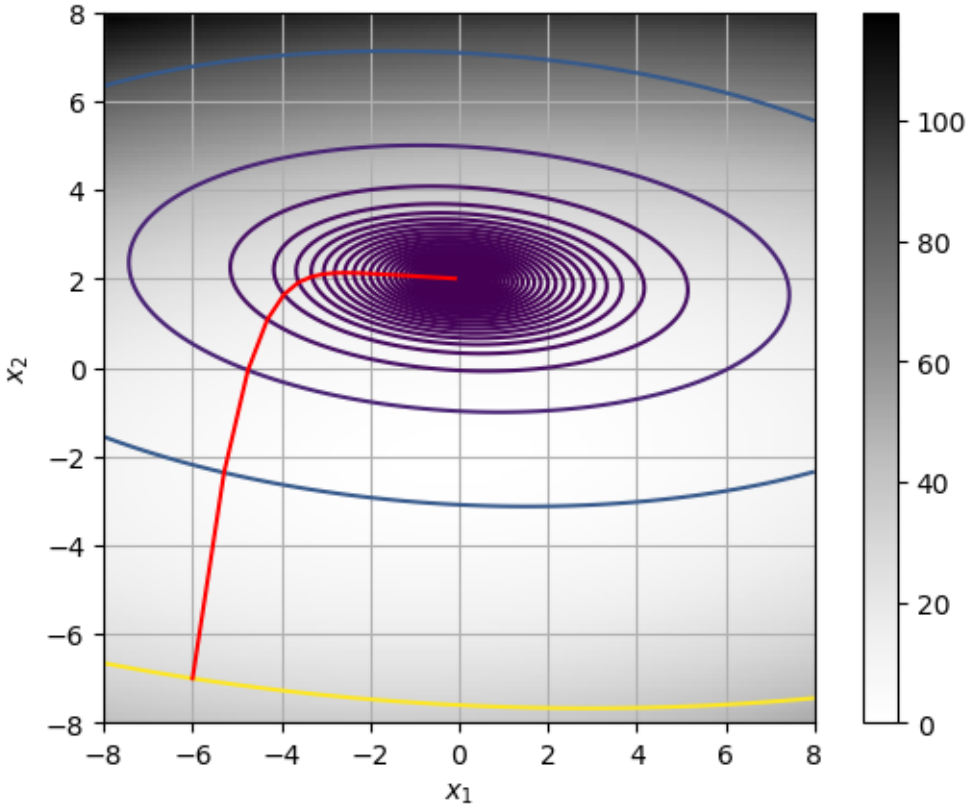
In [1]:
```
using PyPlot, LinearAlgebra, Random
```

In [7]:
```
# Make a matrix...

Random.seed!(1234)
A=randn(5,2).^3
U,S,V=svd(A)
S=[1,0.4]
A=U*diagm(0 =>S)*V';
```

0.1 Steepest descent method with Matrices (Operators)

In [5]:
```
x0 = [0.0,2.0]
 y = A*x0

N,M = size(A)

x1 = collect(-8:0.1:8)
x2 = collect(-8:0.1:8)
J = zeros(length(x2),length(x1))
for i = 1:length(x1)
    for j=1:length(x2)
        J[j,i] = sum((A*[x1[i],x2[j]]-y).^2)
    end
end


 K = 50
v1 = zeros(K)
v2 = zeros(K)
 x = [-6,-7]

v1[1] = x[1];
v2[1] = x[2];
cc=  zeros(K)
cc[1] = sum((A*x-y).^2)
for k = 1:K-1
```

```
        e = A*x-y
        g = A'*e
        alpha=0.5
        x = x - alpha*g
        v1[k+1]=x[1]
        v2[k+1]=x[2]
        cc[k+1] = sum((A*x-y).^2)
    end

    contour(x1,x2,J,sort(cc))
    imshow(J, extent=(-8,8,-8,8), cmap="Greys")
    colorbar()
    plot(v1,v2,"r")
    xlabel(L"$x_1$")
    ylabel(L"$x_2$")
    grid(true);
```




```
In [6]: x
```

```
Out[6]: 2-element Array{Float64,1}:
         -0.09169476740058996
          2.005368985494076
```