Implementing the Levenberg-Marquardt method

Picaud

June 29, 2016

Problem description

This is a **nonlinear least squares** problem

$$\theta = \arg\min_{\theta} \frac{1}{2}\|f(\theta)\|_2^2$$

Mathetamica test

Preamble

Gradient computation can be done as follow

```
grad[f_, var_List] := D[f, #] & /@ var;
```

Function to fit

```
(* Function to fit  *)
f[x_]:=a0+a1*x+a2*x*x;

(* Parameters  *)
theta={a0,a1,a2};

(* Parameter value *)
thetaTrueValue={1,2,-1};
```

Data

```
nSample = 10;
xSample = Table[i/nSample,{i,0,nSample-1}];
ySample = Table[f[xSample[[i]]]  /. Thread[theta ->thetaTrueValue],{i,1,nSample}];
```

```
grad[f_, var_List] := D[f, #] & /@ var;
(* Function to fit  *)
f[x_]:=a0+a1*x+a2*x*x;

(* Parameters  *)
theta={a0,a1,a2};

(* Parameter value *)
thetaTrueValue={1,2,-1};
nSample = 10;
xSample = Table[i/nSample,{i,0,nSample-1}];
ySample = Table[f[xSample[[i]]]  /. Thread[theta ->thetaTrueValue],{i,1,nSample}];

\[Theta] = ySample
```

$$1 \quad 119/100 \quad 34/25 \quad 151/100 \quad 41/25 \quad 7/4 \quad 46/25 \quad 191/100 \quad 49/25 \quad 199/100$$

```
(* Gradient Computation *)
grad[f_, var_List] := D[f, #] & /@ var;

(* Function to fit *)
f[x_] := a0 + a1*x + a2*x*x;

(* Parameters *)
theta = {a0, a1, a2};

(* Generate data to fit from theta true value *)

thetaTrueValue = {1, 2, -1};
nSample = 10;
xSample = Table[i/nSample, {i, 0, nSample - 1}];
ySample =
  Table[f[xSample[[i]]] /. Thread[theta -> thetaTrueValue], {i, 1,
    nSample}];

(* Define objective function *)

evalObjectiveGradHessian[f_] :=
  Module[{F, dF},
```

```
   F = Table[ySample[[i]] - f[xSample[[i]]], {i, 1, nSample}];
   dF = grad[F, theta]; Return[{F.F/2, dF.F, dF.Transpose[dF]}]];

evalObjectiveGradHessian[f]
```

((1 - a0) ^2 + (199/100 - a0 - (9*a1) /10 - (81*a2) /100) ^2 + (49/25 - a0 - (4*a1) /5 - (16*a2) /25)