Bob Ghosh, 42039157, j0k0b

Assignment 2, CPSC 406, January 29

Collaborations

Shrey Verma, 24552151: Question 1
Shikhar Nandi, 51931153: Question 2

Libraries

In [119]:
```julia
using JLD
using LinearAlgebra
using Convex
using SCS
using Plots
```

Question 1

Proof by Contradiction:

Let us assume that $null(A) \cap null(\mathcal{L}) \neq \{0\}$. That is to state that a non-zero solution $\tilde{x}$ exists in both $null(A)$ and $null(\mathcal{L})$.

Secondly, let us assume that $null(A) \cap null(\mathcal{L}) = \{0\}$ results in a unique solution to the problem defined x.

Now, if we define $x^* = \tilde{x} + x$, then:
$f(x^*) = ||Ax^* - b||_2^2 + \lambda||\mathcal{L}x^*||_2^2$
= $||A\tilde{x} + Ax - b||_2^2 + \lambda||\mathcal{L}\tilde{x} + \mathcal{L}x||_2^2$
= $||Ax - b||_2^2 + \lambda||\mathcal{L}x||_2^2$

Since $\tilde{x}$ is a non zero solution in both $null(A) \cap null(\mathcal{L})$, we get the same solution as $f(x)$. Thus we have reached a point of contradiction. A zero unique solution x and $\tilde{x} + x$ result in the same solution.

Proof by Cases:

Let us now take the gradient of the function $f(x)$:
$\nabla f(x) = 2A^T(Ax - b) + 2\lambda\mathcal{L}^T\mathcal{L}x$
Let $\nabla f(x) = 0$

We get, $(A^TA + \lambda\mathcal{L}^T\mathcal{L})x = A^TAb$.

We know that A^TA and $\mathcal{L}^T\mathcal{L}$ are both > 0. So, in order to prove that they both are positive definite we need to prove u^TA^TAu and $u^T\mathcal{L}^T\mathcal{L}u$ to be > 0 for some non-zero u. If we are able to prove that they are positve-definite, then we will know for sure that the problem has an unique solution iff $null(A) \cap null(\mathcal{L}) = \{0\}$. We also know that $\lambda > 0$.

Case 1: $(u \in null(A)) \cap (u \notin null(\mathcal{L}))$
$u^T(A^TA + \lambda\mathcal{L}^T\mathcal{L})u$
$= u^T\lambda\mathcal{L}^T\mathcal{L} > 0$

Case 2: $(u \notin null(A)) \cap (u \in null(\mathcal{L}))$
$u^T(A^TA + \lambda\mathcal{L}^T\mathcal{L})u$
$= u^TA^TAu > 0$

Case 3: $(u \notin null(A)) \cap (u \notin null(\mathcal{L}))$
$u^T(A^TA + \lambda\mathcal{L}^T\mathcal{L})u$
$= u^TA^TAu + u^T\lambda\mathcal{L}^T\mathcal{L}u > 0.$

Case 4: $(u \in null(A)) \cap (u \in null(\mathcal{L}))$
This is the default case given to us, as $null(A) \cap null(\mathcal{L}) = \{0\}$.

Hence, we can now claim that $u^T(A^TA + \lambda\mathcal{L}^T\mathcal{L})u$ is a positive definite, and so there is an unique solution iff $null(A) \cap null(\mathcal{L}) = \{0\}$.

Question 2

a

We are given that $g = Q^TA^Tb$ and $A^TA = QDQ^T$
For the sake of simplicity, let us also declare:
$QG = A^Tb$ [1]
$b^TA = G^TQ^T$ [2]

$||x||_2 = ((A^TA + \gamma I)^{-1}A^Tb)^T(A^TA + \gamma I)^{-1}A^Tb$
$= b^TA(A^TA + \gamma I^T)^{-1}(A^TA + \gamma I)^{-1}A^Tb$
$= b^TA((A^TA + \gamma I^T)(A^TA + \gamma I))^{-1}A^Tb$
$= b^TA(A^TAA^TA + A^TA\gamma I^T + \gamma IA^TA + \gamma^2 I)^{-1}A^Tb$
$= b^TA(A^TAA^TA + 2\gamma IA^TA + \gamma^2 I)^{-1}A^Tb$
$= b^TA(A^TAA^TA + 2\gamma IA^TA + \gamma^2 I)^{-1}A^Tb$
$= G^TQ^T(QDQ^TQDQ^T + 2\gamma IQDQ^T + \gamma^2 I)^{-1}QG$ [From 1 and 2]
$= G^TQ^T(QD^2Q^T + 2\gamma IQDQ^T + \gamma^2 I)^{-1}QG$
$= G^TQ^T(QDQ^T + \gamma I)^{-2}QG$

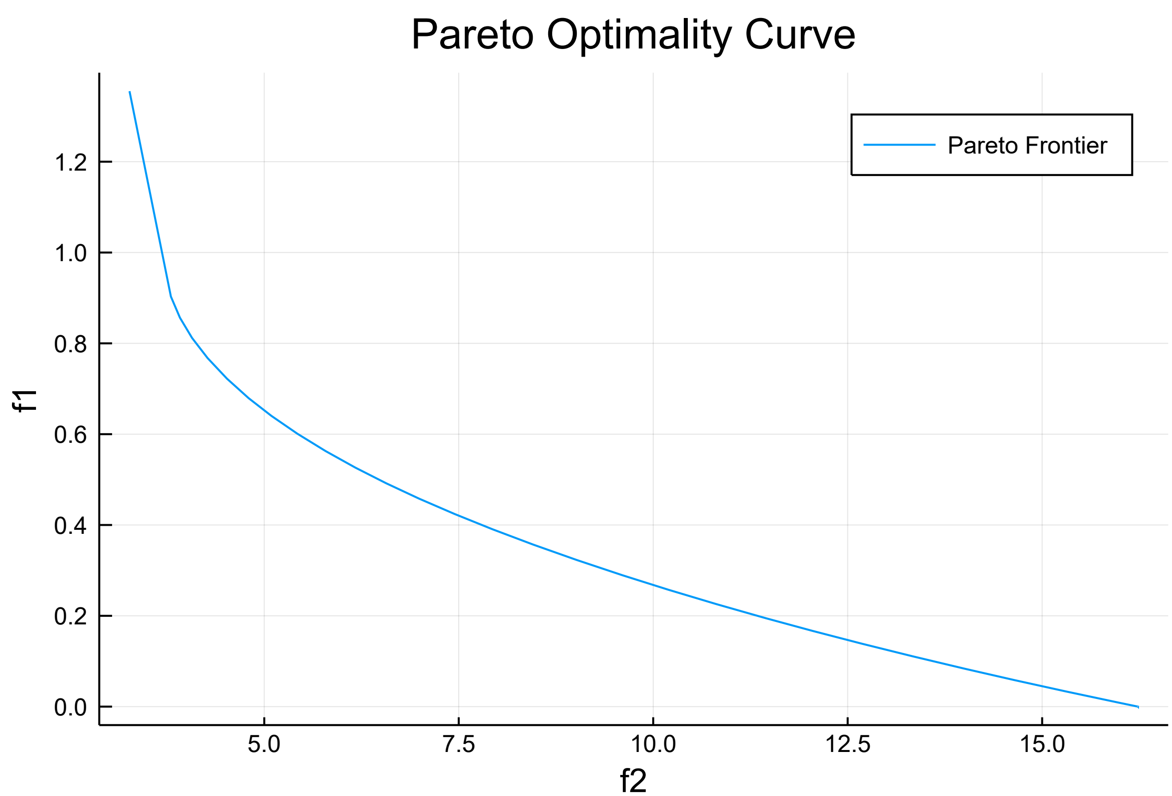
$$= \frac{G^T Q^T QG}{(QDQ^T + \gamma I)^2}$$

$$= \frac{G^T G}{(QDQ^T + \gamma I)^2}$$

In [230]:

```
A = rand(100,100);
b = rand(100,1);
f1 = Array{Float64}(undef,100);
f2 = Array{Float64}(undef,100);
for i in 1:100
    x = Variable(size(A)[2],1);
    g = i;
    sol = minimize(0.5*sumsquares(A*x - b)+g*norm(x,1));
    solve!(sol, SCSSolver(verbose = 0));
    f1[i] = 0.5*norm((A*x.value)-b)^2;
    f2[i] = norm(x.value, 1);
end
plot(f1, f2, title="Pareto Optimality Curve", label="Pareto Frontier",
    xaxis= "f2", yaxis = "f1")
```

Out[230]:



Using random matrices, we get the above Pareto Optimality Cuve. When using specific values we will get a similar graph.

b

In [120]:

```
A = load("hw2_p2_sparse_A.jld", "data");
b = load("hw2_p2_sparse_b.jld", "data");
x0 = load("hw2_p2_sparse_signal.jld", "data");
```

In [121]:
```
x = Variable(size(A)[2],1);
ans_1 = minimize(0.5*sumsquares(A * x - b)+1.0*norm(x,1));
solve!(ans_1, SCSSolver())
```

```
----------------------------------------------------------------------------
	SCS v2.1.1 - Splitting Conic Solver
	(c) Brendan O'Donoghue, Stanford University, 2012
----------------------------------------------------------------------------
Lin-sys: sparse-indirect, nnz in A = 5256, CG tol ~ 1/iter^(2.00)
eps = 1.00e-005, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 10, rho_x = 1.00e-003
Variables n = 103, constraints m = 206
Cones:	primal zero / dual free vars: 1
	linear vars: 101
	soc vars: 104, soc blks: 2
Setup time: 9.93e-004s
----------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
----------------------------------------------------------------------------
     0|7.17e+018 1.95e+018 1.00e+000 -6.00e+020 1.27e+018 4.47e+020 4.22e-004
   100|2.18e-003 2.71e-003 6.78e-005 3.88e+000 3.88e+000 3.22e-016 1.69e-002
   200|6.80e-005 8.12e-005 1.17e-005 3.88e+000 3.88e+000 3.25e-016 3.06e-002
   260|8.64e-006 9.57e-006 3.75e-006 3.88e+000 3.88e+000 1.32e-016 3.66e-002
----------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 3.66e-002s
	Lin-sys: avg # CG iterations: 15.74, avg solve time: 1.14e-004s
	Cones: avg projection time: 2.16e-007s
	Acceleration: avg step time: 2.31e-005s
----------------------------------------------------------------------------
Error metrics:
dist(s, K) = 1.2939e-015, dist(y, K*) = 1.1102e-016, s'y/|s||y| = 1.3827e-016
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 8.6422e-006
dual res:   |A'y + c|_2 / (1 + |c|_2) = 9.5667e-006
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 3.7545e-006
----------------------------------------------------------------------------
c'x = 3.8802, -b'y = 3.8803
============================================================================
```

In [122]:
```
res = A*x.value;
res = 0.5*(norm(res-b))^2
```

Out[122]:
```
0.8029071285364764
```

In [123]:
```
norm(x.value, 1)
```

Out[123]:
```
3.077440717262689
```

The **optimal value** = 3.8802.
We also find that **accuracy** to be 0.8029071285364764, and the **sparsity metric** to be 3.077440717262689.

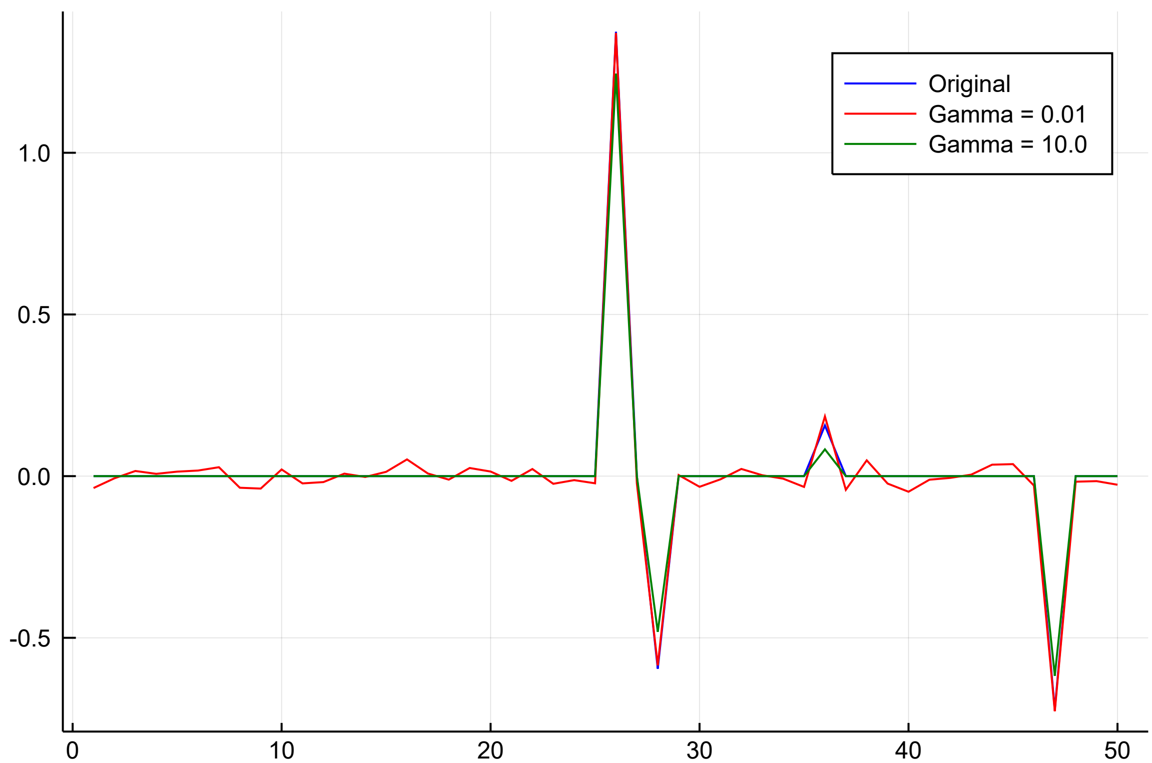
c

In [124]:
```
x = Variable(size(A)[2],1);
ans_2 = minimize(0.5*sumsquares(A * x - b)+0.01*norm(x,1));
solve!(ans_2, SCSSolver(verbose = 0))
x_low = x.value;
```

In [125]:
```
x = Variable(size(A)[2],1);
ans_3 = minimize(0.5*sumsquares(A * x - b)+10*norm(x,1));
solve!(ans_3, SCSSolver(verbose = 0))
x_high = x.value;
```

In [126]:
```
plot(x0, label = "Original", color = "blue")
plot!(x_low, label = "Gamma = 0.01", color = "red")
plot!(x_high, label = "Gamma = 10.0", color = "green")
```

Out[126]:

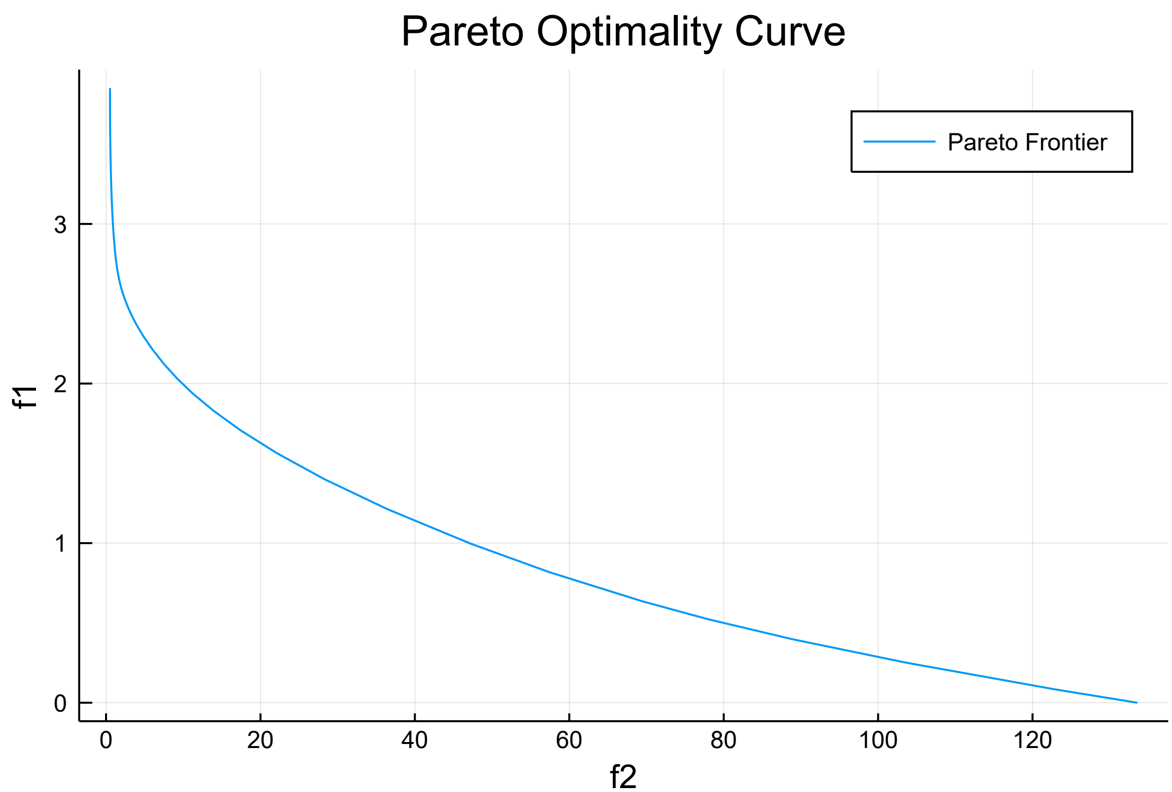


It is evident from the graph that when γ=10.0, we see a better fit that corresponds to the original signal. At γ=0.01 we see a lot of noise and less congruency when compared to the original.

d

In [127]:
```
gm = exp10.(range(-3, stop=3,length=100));
f1 = Array{Float64}(undef,100);
f2 = Array{Float64}(undef,100);
for i in 1:100
    x = Variable(size(A)[2],1);
    ans_ = minimize(0.5*sumsquares(A*x - b)+gm[i]*norm(x,1));
    solve!(ans_, SCSSolver(verbose = 0));
    f1[i] = 0.5*norm((A*x.value)-b)^2;
    f2[i] = norm(x.value, 1);
end
plot(f1, f2, title="Pareto Optimality Curve", label="Pareto Frontier",
    xaxis= "f2", yaxis = "f1")
```

Out[127]:



e

In [128]:

```
A = load("hw2_p2_smooth_A.jld", "data");
b = load("hw2_p2_smooth_b.jld", "data");
x0 = load("hw2_p2_smooth_signal.jld", "data");
n = 50;
x = Variable(size(A)[2],1);

D = Bidiagonal(ones(n), -ones(n-1), :U);
D = D[1:n-1,:];
gamma = 1.0;
ans_4 = minimize(0.5*sumsquares(A * x - b)+gamma*norm(D*x,1));
solve!(ans_4, SCSSolver());
```

```
----------------------------------------------------------------------------
	SCS v2.1.1 - Splitting Conic Solver
	(c) Brendan O'Donoghue, Stanford University, 2012
----------------------------------------------------------------------------
Lin-sys: sparse-indirect, nnz in A = 5349, CG tol ~ 1/iter^(2.00)
eps = 1.00e-005, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 10, rho_x = 1.00e-003
Variables n = 102, constraints m = 204
Cones:	primal zero / dual free vars: 1
	linear vars: 99
	soc vars: 104, soc blks: 2
Setup time: 4.86e-004s
----------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
----------------------------------------------------------------------------
     0|7.14e+018 1.85e+018 1.00e+000 -1.37e+021 4.34e+017 1.02e+021 4.19e-004
   100|7.41e-003 1.15e-002 1.94e-004 3.85e+000 3.85e+000 2.33e-016 1.59e-002
   200|1.99e-003 2.94e-003 3.72e-005 3.86e+000 3.86e+000 2.56e-016 3.26e-002
   300|5.06e-004 7.18e-004 3.80e-005 3.88e+000 3.88e+000 4.23e-017 4.90e-002
   400|1.56e-003 2.48e-003 8.15e-006 3.88e+000 3.88e+000 1.88e-016 6.47e-002
   500|1.37e-005 2.10e-005 6.08e-006 3.88e+000 3.88e+000 9.11e-017 7.88e-002
   560|5.41e-006 6.03e-006 6.61e-008 3.88e+000 3.88e+000 1.23e-016 8.65e-002
----------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 8.65e-002s
	Lin-sys: avg # CG iterations: 18.50, avg solve time: 1.32e-004s
	Cones: avg projection time: 2.35e-007s
	Acceleration: avg step time: 1.85e-005s
----------------------------------------------------------------------------
Error metrics:
dist(s, K) = 2.8936e-015, dist(y, K*) = 0.0000e+000, s'y/|s||y| = -7.2795e-017
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 5.4131e-006
dual res:   |A'y + c|_2 / (1 + |c|_2) = 6.0294e-006
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 6.6107e-008
----------------------------------------------------------------------------
c'x = 3.8763, -b'y = 3.8763
============================================================================
```

In [129]:

```
res = A*x.value;
res = 0.5*(norm(res-b))^2
```

Out[129]:

```
0.856494473282389
```

In [130]:
```
norm(D*x.value,1)
```

Out[130]:
```
3.0200670722151903
```

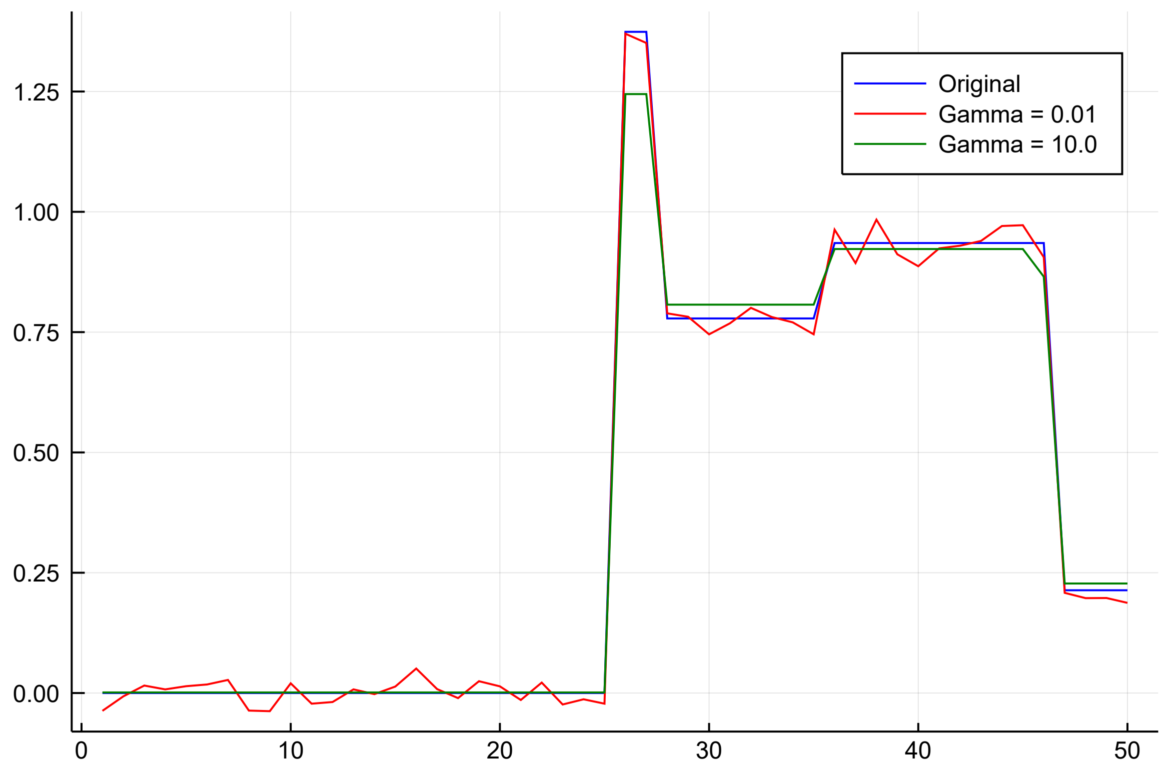
The **optimal value** = 3.8763.
We also find that **accuracy** to be 0.856494473282389, and the **sparsity metric** to be 3.0200670722151903.

In [131]:
```
gamma = 0.01;
ans_5 = minimize(0.5*sumsquares(A * x - b)+gamma*norm(D*x,1));
solve!(ans_5, SCSSolver(verbose = 0));
x_low = x.value;
```

In [132]:
```
gamma = 10;
x = Variable(size(A)[2],1);
ans_6 = minimize(0.5*sumsquares(A * x - b)+gamma*norm(D*x,1));
solve!(ans_6, SCSSolver(verbose = 0))
x_high = x.value;
```

In [133]:
```
plot(x0, label = "Original", color = "blue")
plot!(x_low, label = "Gamma = 0.01", color = "red")
plot!(x_high, label = "Gamma = 10.0", color = "green")
```

Out[133]:

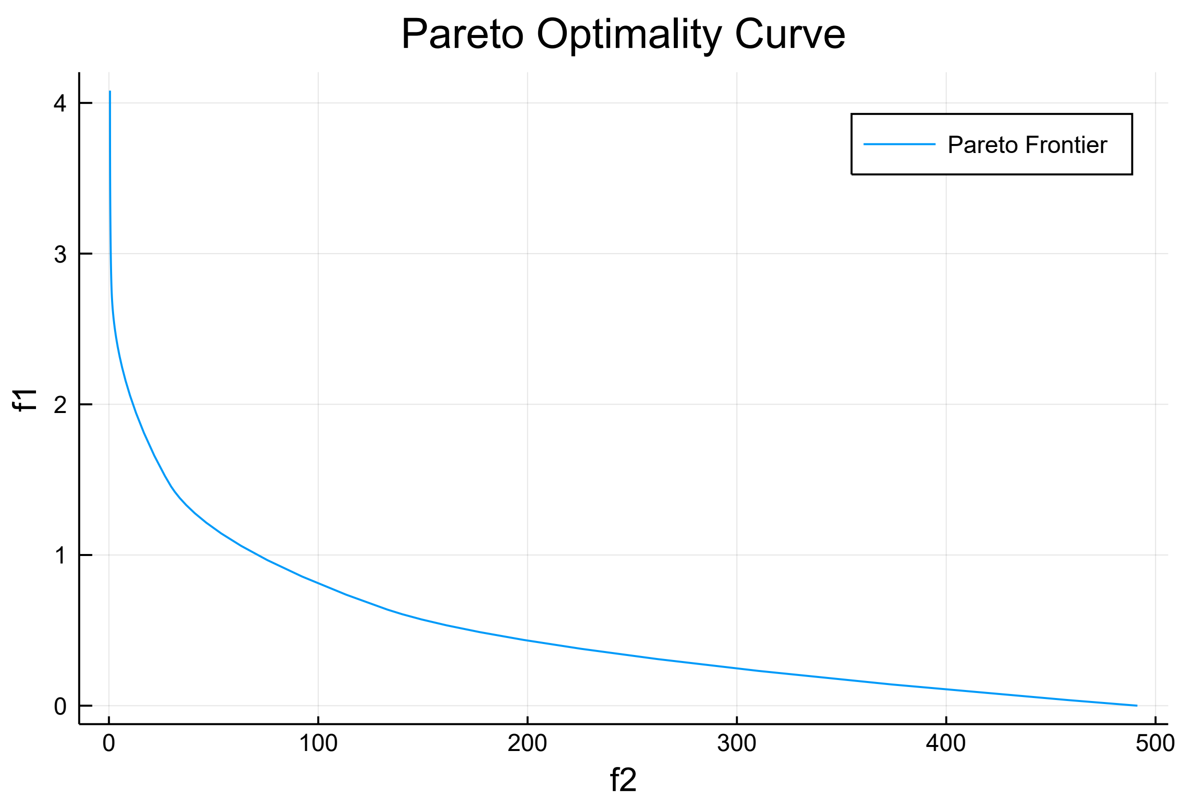


Just like the last comparison we made, for γ=10.0, we see a better fit that corresponds to the original signal. At γ=0.01 we see a lot of noise and less congruency when compared to the original, although it matches exactly in some parts.

f

In [134]:
```
gx = exp10.(range(-3, stop=3,length=100));
f1 = Array{Float64}(undef,100);
f2 = Array{Float64}(undef,100);
i = 1;
for g in gx
    x = Variable(size(A)[2],1);
    ans = minimize(0.5*sumsquares(A * x - b)+g*norm(D*x,1));
    solve!(ans, SCSSolver(verbose=0));
    f1[i] = 0.5*norm((A*x.value)-b)^2;
    f2[i] = norm(D*x.value, 1);
    i += 1;
end
plot(f1, f2, title="Pareto Optimality Curve", label="Pareto Frontier",
    xaxis= "f2", yaxis = "f1")
```

Out[134]:



Question 3

a

We are given that $f(x) := \sum_{i=1}^{m}(||x - c_i||_2^2 - d_i^2)^2$.

$\therefore\ f(x) = (||x - c_1||_2^2 - d_1^2)^2 + (||x - c_2||_2^2 - d_2^2)^2 + \cdots + (||x - c_m||_2^2 - d_m^2)^2$

We are assuming $n = 2$.

Hence, $f(x) = ((x - c_{1.1})^2 + (x - c_{2.1})^2 - d_1^2)^2 + \cdots + ((x - c_{1.m})^2 + (x - c_{2.m})^2 - d_m^2)^2$

$$\nabla f(x) = 2((x - c_{1,1})^2 + (x - c_{2,1})^2 - d_1^2) \times (2(x - c_{1,m}) + 2(x - c_{2,m})) + \cdots$$
$$+ 2((x - c_{1,m})^2 + (x - c_{2,m})^2 - d_m^2) \times (2(x - c_{1,m}) + 2(x - c_{2,m}))$$

Finally,
$\nabla f(x) = 2 \sum_{i=1}^{m} (2(x - c_{1,i}) + 2(x - c_{3,i}))(||x - c_i||_2^2 - d_i^2)$.
$\nabla f(x) = 4 \sum_{i=1}^{m} ((x - c_{1,i}) + (x - c_{2,i}))(||x - c_i||_2^2 - d_i^2)$.

b

We can define $r(x) = ||x - c_i||_2^2 - d_i^2$. Therefore,
$\nabla r(x) = 2(x - c_{1,i}) + 2(x - c_{2,i})$.

$\therefore$ We can write
$$J(x) = \begin{bmatrix} 2(x_1 - c_{1,1}) & 2(x_2 - c_{2,1}) \\ \vdots & \vdots \\ 2(x_1 - c_{1,m}) & 2(x_2 - c_{1,m}) \end{bmatrix}$$

c

d

(i) Gradient Descent

In [219]:
```
c = load("hw2_p3_C.jld", "data");
d = load("hw2_p3_d.jld", "data");
x = load("hw2_p3_signal.jld", "data");
xk = zeros(Float64,(101,2));
alp  = 10^(-7.301365);
xk[1,1] = 1000;
xk[1,2]  = -500;
for i in 1:100
    fx1 = 0;
    fx2 = 0;
    for j in 1:5
        fx1 += ((xk[i,1]-c[1,j]+xk[i,1]-c[2,j]))*
        (norm(xk[i,1]-c[j])^2-d[j]^2);
        fx2 += ((xk[i,2]-c[1,j]+xk[i,2]-c[2,j]))*
        (norm(xk[i,2]-c[j])^2-d[j]^2);
    end
    xk[i+1, 1] = xk[i,1]-(alp*4*fx1);
    xk[i+1, 2] = xk[i,2]-(alp*4*fx2);
end
display(xk');
```

```
2×101 Adjoint{Float64,Array{Float64,2}}:
 1000.0  -1000.68    999.624  -998.8    …  -49.4586  -49.2223  -48.9894
 -500.0   -250.749  -219.382  -198.388     -49.5371  -49.2997  -49.0657
```

We find that if we set $\bar{\alpha} = 10^{-7.301365}$, we get a non-diverging solution. As evident from the last 2 values, that we see a convergance.