2.3

In [42]:

```
using Gurobi, StatsBase, CSV, DataFrames, JuMP, LinearAlgebra, Distributions, Ra
```

a)

In [51]:

```
function solve_inner_problem(X,Y,s,γ,lam1)
    m = Model(Gurobi.Optimizer)
    set_optimizer_attribute(m, "OutputFlag", 0)
    n,p = size(X)

    @variable(m, a[1:p])
    @variable(m, u[1:n])

    @constraint(m,[k=1:p],dot(X[:,k],u)+a[k] <= lam1)
    @constraint(m,[k=1:p],-(dot(X[:,k],u)+a[k]) <= lam1)

    @objective(m, Max, (0.5 * sum(Y[i]^2 for i=1:n)) - (0.5 * sum((Y[k]-u[k])^2

    optimize!(m)
    alpha_i = value.(a)
    obj = objective_value(m)
    grad_s = -(γ/2)*alpha_i.^2

  return obj, grad_s
end;
```

In [50]:

```
function sparse_regression(X,Y,k,γ,lam1,s0=[])

    m = Model(Gurobi.Optimizer)
    n,p = size(X)
    set_optimizer_attribute(m, "OutputFlag", 0)

    ###
    # Step 1: Define the Variables:
    ###
    @variable(m, s[1:p], Bin)
    @variable(m, t >= 0)

    ###
    # Step 2: Set Up Constraints and Objective
    ###
    @constraint(m, sum(s) <= k)
    # Initial solution: if none is provided, start at arbitrary point
    if length(s0) == 0
        s0 = zeros(p)
        s0[1:k] .= 1
    end
    obj0, grad0 = solve_inner_problem(X,Y, s0, γ, lam1)
    @constraint(m, t >= obj0 + dot(grad0, s - s0))
    # Objective
    @objective(m, Min, t)

    ###
    # Step 3: Define the outer approximation function
    ###
    function outer_approximation(cb_data)
        s_val = []
        for i = 1:p
            s_val = [s_val;callback_value(cb_data, s[i])]
        end
        obj, grad = solve_inner_problem(X,Y, s_val, γ, lam1)
        # add the cut: t >= obj + sum(∇s * (s - s_val))
        offset = sum(grad .* s_val)
        con = @build_constraint(t >= obj + sum(grad[j] * s[j] for j=1:p) - offse
        MOI.submit(m, MOI.LazyConstraint(cb_data), con)
    end
    MOI.set(m, MOI.LazyConstraintCallback(), outer_approximation)

    ###
    # Step 4: Solve
    ###
    optimize!(m)
    s_opt = JuMP.value.(s)
    s_nonzeros = findall(x -> x>0.5, s_opt)
    β = zeros(p)
    X_s = X[:, s_nonzeros]
    # Formula for the nonzero coefficients
    β[s_nonzeros] = γ * X_s' * (Y - X_s * ((I / γ + X_s' * X_s) \ (X_s'* Y)))

    return Dict("support" => s_opt, "coefs" => β, "selected_features" => s_nonze

end;
```

b)

In [47]:

```
train = CSV.read("/Users/bennetthellman/Desktop/OneDrive - Massachusetts Institu
```

In [48]:

```
# Load and center responses (so no intercept term is needed)
Y = Vector(train[:,1])
Y = Y .- mean(Y)

# Load and standardize data
X = Array{Float64,2}(train[:,2:end])
X = (X.-mean(X,dims=1))./std(X,dims=1)
n,p = size(X);
```

In [49]:

```
for k in 1:4
    betas = sparse_regression(X,Y,k,1,.05)
    print(betas)
end
```

```
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Dict{String, Vector{T} where T}("coefs" => [0.0, 0.0, 0.6908866666470792, 0.0,
0.0, 0.0], "support" => [0.0, 0.0, 1.0, 0.0, 0.0, 0.0], "selected_features" =>
[3])Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Dict{String, Vector{T} where T}("coefs" => [0.0, 0.0, 0.7040890970475672, -0.667
6505624020903, 0.0, 0.0], "support" => [-0.0, -0.0, 1.0, 1.0, -0.0, -0.0], "sele
cted_features" => [3, 4])Academic license - for non-commercial use only - expire
s 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Dict{String, Vector{T} where T}("coefs" => [0.0, 0.0, 0.7153821581288513, -0.628
1568916307213, 0.0, 0.17168416423705912], "support" => [-0.0, -0.0, 1.0, 1.0, 0.
0, 1.0], "selected_features" => [3, 4, 6])Academic license - for non-commercial
use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Academic license - for non-commercial use only - expires 2022-08-19
Dict{String, Vector{T} where T}("coefs" => [0.0, 0.0, 0.7986529556124364, -0.641
9951433047906, 0.15400682525028425, 0.16513267852008184], "support" => [-0.0, -
0.0, 1.0, 1.0, 1.0, 1.0], "selected_features" => [3, 4, 5, 6])
```

```
In [ ]:
```