Optimization

Class 10: Penalty & Barrier Functions

1 Constraints

Most real optimization problems come with constraints on the variables.

1.1 Monthly payment example

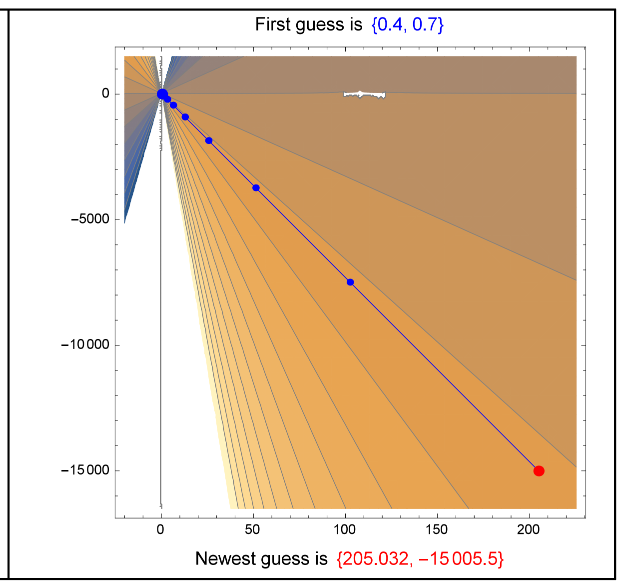
x is term in months and y is the rebate amount; and there practical constraints like the following on these variables:

→ $0 \leq x \leq 108$ (lenders limit the term)

→ $0 \leq y \leq 30$ (can't rebate more than the total cost of \$30 K)

Our methods so far have ignored this issue; we have simply applied a method and hoped for the best. However, we saw that if the first guess isn't chosen properly, both Newton's optimization method and the steepest descent method may be attracted to the arbitrarily small values of the monthly payment function that can be generated by variables that violate the constraints. Here's the output of Newton's optimization method applied to the monthly payment function with the default initial values.

newest 20 guesses	
0.4	0.7
0.800848	-28.7241
1.6018	-87.4638
3.20362	-204.93
6.40724	-439.859
12.8145	-909.719
25.6289	-1849.44
51.2579	-3728.87
102.516	-7487.75
205.032	-15005.5



The newest guess clearly violates both of the practical constraints above, and one way to avoid this phenomenon is to "penalize" constraint violation by adding value to the objective function.

2 Penalties

Idea: build constraints into the objective function by adding a penalty function.

Single-variable example:

$$\min f(x) \text{ over all real variables } x \text{ in the interval } [-1, 2].$$

The unconstrained *penalty problem*

$$\min f(x) + p(x) \text{ over all real variables } x$$

uses a *penalty function* $p(x)$ with relatively large values at points outside the interval. The *ideal penalty function* here is

$$p_{\text{ideal}}(x) = \begin{cases} 0 & \text{when } -1 \le x \le 2 \\ \infty & \text{otherwise} \end{cases}$$

Group Problem 1

(a) Sketch the graph of $(x+2)^2 + p_{\text{ideal}}(x)$ and identify two points of discontinuity.

(b) Explain why $p_{\text{ideal}}(x)$ is "ideal".

2.1 Continuous penalty

Step 1. Rewrite the constraints in standard form $g_i(\mathbf{x}) \ge 0$ using constraint functions g_i. For example,

$$x \in [-1, 2] \quad \Leftrightarrow \quad x \ge -1 \text{ and } x \le 2 \quad \Leftrightarrow \quad \underbrace{x+1}_{g_1(x)} \ge 0 \text{ and } \underbrace{2-x}_{g_2(x)} \ge 0$$

Step 2. Using constants $\lambda_i > 0$, add a penalty term like $-\lambda_i\, g_i(\mathbf{x})$ for each constraint function. When the constraint is violated $g_i(\mathbf{x}) < 0$, the product $-\lambda_i\, g_i(\mathbf{x})$ is positive (and hence a penalty for minimization). For example

$$p(x) = -\lambda_1\,(x+1) - \lambda_2(2-x).$$

Group Problem 2

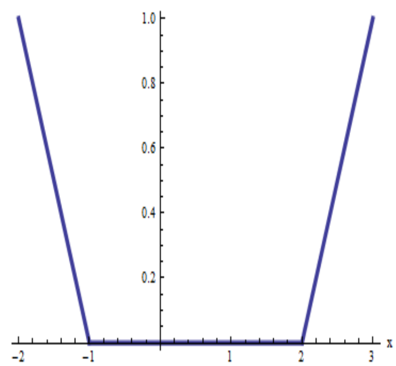
(a) Add this penalty function to the objective function $f(x) = -x/2$ and try to find a penalized minimizer. For what choices of constants will there be penalized minimizers?

(b) What is the minimizer of $f(x) = -x/2$ over the interval $[-1, 2]$?

2.2 Removing rewards

Instead of $-\lambda_i\, g_i(\mathbf{x})$, use

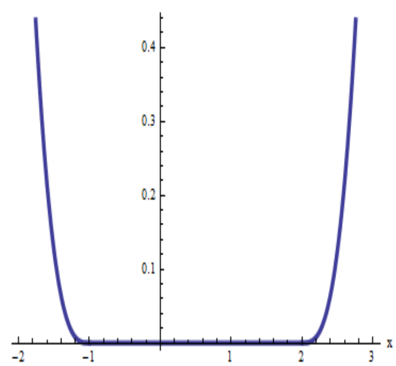
$$-\lambda_i\ \min\{g_i(\mathbf{x}), 0\} = \begin{cases} -\lambda_i\, g_i(\mathbf{x}) & \text{if } g_i(\mathbf{x}) < 0 \text{ (constraint violated)} \\ 0 & \text{if } g_i(\mathbf{x}) \geq 0 \text{ (constraint satisfied)} \end{cases}$$



Q: Trouble for Newton's optimization method?

2.2.1 Twice-differentiable penalty

Smooth the corners by cubing: $-\lambda_i\ \min\{g_i(\mathbf{x}), 0\}^3$.



3 Barriers

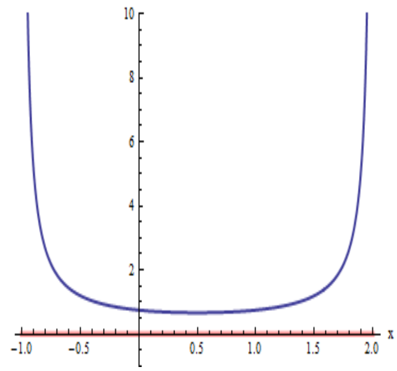
We will see in lab that penalty functions do not necessarily enforce the constraints. If constraints are truly inviolable (e.g. we can't have negative term on a loan), we can use a different kind of penalty function, called a *barrier* function (because it attempts to create a barrier at the boundary of the constraint).

3.1 Inverse barrier function

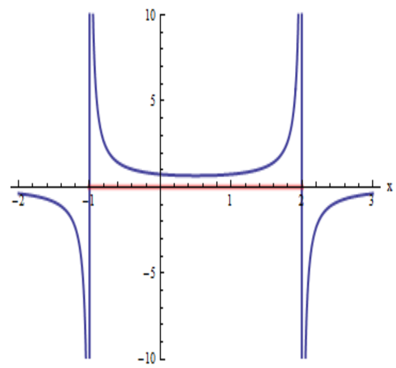
To enforce $g_i(\mathbf{x}) \geq 0$, we could use the inverse $\lambda_i / g_i(\mathbf{x})$ for a constant $\lambda_i > 0$, since this gets very large (hence penalizes) as $g_i(\mathbf{x}) > 0$ approaches the boundary value 0. For our single-variable example with constraint $-1 \leq x \leq 2$, this corresponds to a (barrier) penalty function of the form

$$p(x) = \frac{\lambda_1}{x+1} + \frac{\lambda_2}{2-x}$$

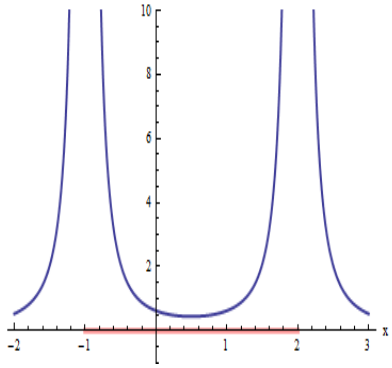
graphed below with $\lambda_1 = \lambda_2 = 0.5$:



This clearly creates a very steep penalty at the boundary of the constraint, but it also can reward constraint violation; as a wider-view of the graph shows:



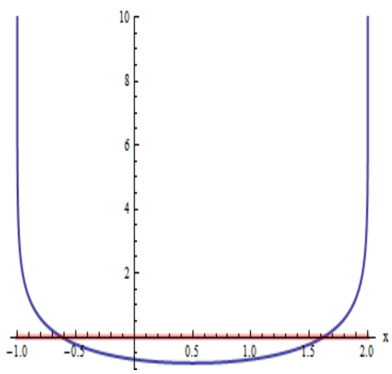
Thus, any optimization method that can skip past the steep penalty at the boundary of the constraint, can get trapped outside the constraint. This reward can be avoided through careful choice of step-factors to avoid skipping over the boundary (there is a formula for this that involves the second-derivative matrix), or it can be removed entirely by squaring the denominator of each inverse barrier function $\lambda_i / \big(g_i(\mathbf{x})\big)^2$:



Of course this latter situation doesn't assign a very steep penalty as long as the boundary is avoided, whether or not x is within the constraint interval; so it would still be desirable to avoid skipping well past the boundary through careful choice of step-factors.

3.2 Logarithmic barrier function

To enforce $g_i(\mathbf{x}) \geq 0$, we could instead use the negative logarithm $-\lambda_i \log\big(g_i(\mathbf{x})\big)$. This barrier function is undefined when $g_i(\mathbf{x}) \leq 0$, so again we need to avoid overshooting the boundary.



3.3 Warning: barrier functions fail for equation constraints

Barrier functions cannot be used for equation constraints, which are effectively all "boundary". Since barrier functions are undefined when $g_i(\mathbf{x}) = 0$, the penalized objective corresponding to an equation constraint is also undefined. In contrast to this, the penalty functions we developed last time *can* be used for equation constraints. The progression below shows the cubed min-type penalty functions associated with the constraints $-1 \leq x \leq 2$, $-1 \leq x \leq 0$, and $-1 = x$ respectively:

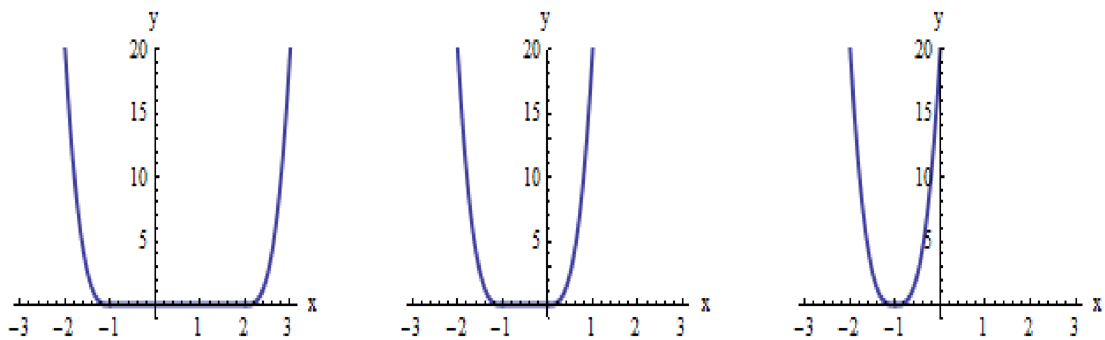


Figure 1: cubed min-type penalties for $-1 \le x \le 2$, $-1 \le x \le 0$, and $-1 = x$

We can build analogous barrier functions for the first two cases (of inequalities), but not for the equation constraint $-1 = x$ in the third case.

4 Lab Exercise

Pick constants $\lambda_1 > 0$ and $\lambda_2 > 0$ and use smooth penalties $-\lambda_i \min\{g_i(\mathbf{x}), 0\}^3$ to construct a penalty function $p_3(x)$ corresponding to $[-1, 2]$, and plot the penalized function $f(x) + p_3(x)$ for $f(x) = (x+2)^2$. (Use *Mathematica*'s `Min` command to construct $p_3(x)$.)

(a) Confirm that the penalized minimizer does not satisfy the constraints.

(b) Use *Mathematica*'s `Manipulate` command to experiment with choices of constants $\lambda_1 > 0$ and $\lambda_2 > 0$ that move the penalized minimizer closer to the true minimizer.