

Symbolic Subdifferentiation in Python

Maurizio Caló and Jaehyun Park

Project Proposal for EE364b, Stanford University, Spring 2010-11

April 27, 2011

1 Overview

Motivated by the first two weeks of lectures, we decided to create a Python package (tentative name: Subgradient-PY, or SPY) that solves optimization problems using subgradient methods. Using SPY, one will be able to formulate and solve an optimization problem using the following syntax (subject to change):

```
from spy import *
A = rand(20, 5)
x = vector('x', 5)
prob = spy_prob(minimize(norm(x, 1)), [A*x <= 1])
prob.solve()
```

2 Implementation

Along with CVX-like library functions, SPY will implement four important classes:

- **Expression:** Any real-valued mathematical expression is an object of Expression type. It can contain variables whose values are not predetermined.
- **Constraint:** A constraint is an inequality of the form (convex) \leq (concave) or an equality of the form (affine) = (affine).
- **Objective:** An objective is simply a pair of `minimize` and a convex function, or a pair of `maximize` and a concave function.
- **(Optimization) Problem:** A problem is a pair of objective and a list of constraints.

The essential part of the project is to compute subgradients correctly and efficiently, since all other methods will rely heavily on subgradients. Details are omitted.