Statistical Techniques in Robotics (16-831, F08)	Lecture #8 (Thursday September 17)

Convex Optimization and Subgradients

Lecturer: Drew Bagnell *Scribe: Jared Goerner*

1 Experts and Online Learning

1.1 Review

We want low regret R, or zero time-averaged regret. This means there can be different rates of 'no regret', such as R = $\sqrt[2]{T}$ or R = $T^{7/8}$. Regret itself is expressed by ths sum of the difference in loss functions:

$$Regret = \sum_t L_t\left(e_t\right) - \; L_t(e_t^*)$$

Where Lt is the loss function at time t, et is the expert we chose at time t, and e*t is the best expert at time t. Our *instentaneous regret* is the portion within the sum. We make no explicit assumptions for this algorithm, but there are some implicit assumptions. Namely, that our choice does not affect the world, or, if we had changed our choice then it would be better. A counterexample is any game, where our choice of move influences the opponent.

1.2 Example

Robot Vision

A vision system estimates objects in front of it very well up to 20 meters. It is much less accurate at 70 meters. So, we use local vision to train the far vision. This is bad when we have an expert that determines "stay away from big green things", which might be trees (which we can't drive through) or shrubs (which we can). It's bad because we never get the chance to correct our mistake, as we're always staying away from anything classified that way.

1.3 Big Picture

We want to turn learning into optimization to make our lives (and the computation) easier. If we can just run the weighted majority algorithm, then great! However, the computation for that is difficult. So what can we do?

1. Linear Optimization
2. Convex Functions

2 Convex Sets and Functions

2.1 Definitions

A convex set is a set such that any linear combination of two points in the set is also in the set. **R**n is convex, as are all regular polygons. {||x||2 = c} is convex (it resembles a filled-in circle) but {||x||2 = c} is not, since the points on the interior of the circle are not included in the set. Likewise, a star-shaped set would not be convex, since if you picked a point in one star tip and a point in another star tip, not all of the linear combinations of those two points would be in the set.

Convex functions are functions such that the epigraph (everything above the function) is a convex set. Intuitively, you can think about it as if the function will hold water if it was poured in from the top of the graph. Also, if there exists a line such that the entire function is above (=) the line, there exists a sub-gradient. If for all points there is an intersecting sub-gradient, the function is convex. As long as the function is differentiable, the sub-gradient will simply be the gradient of the function. Also, note the following two properties of convex functions: 1. Argmin f(x) is not always unique. 2. f(x) has no local minima.

2.2 Jensen's Inequality

Another property of convex functions is that for any x1 = x2 and any line between f(x1) and f(x2), f(x) is below that line when x1 = x = x2. Jensen's Inequality states this formally:

$$if \sum_i \theta_i = 1$$

$$then\ f\left(\theta_1 x_1 + \ \cdots + \ \theta_n x_n\right) \leq \ \sum_i \theta_i f(x_i)$$

3 Online Convex Programming Problem

3.1 Algorithm

ft(x) **R** where for all t, ft is convex. (At every time, there is a function). The set of experts consists of one expert for each point in the set; the set G is convex, and all x G. Our regret is $\sum_t L_t\left(x_t\right) - \ L_t(x_t^*)$ where x* is whatever minimuzes the sum. For example, if ft = ft-1, then x* = argminx ft(x). Note that x* is always the same point – x* = argminx $\sum_t f_t(x)$.

3.2 Example

If x ranges between 0 and 1, and ft(x) follows the pattern f1(x) = x, f2(x) = 1 – x, f3(x) = x, and so on and so forth, then x* will be equal to 1.