Automated optimization of a course of actions in a stochastic environment

Thomas Vicente

General principle

Explain how it works in sentences

Algorithm 1 General version

Initialized:
  X: empty m-columns feature matrix
  y: empty target vector
  X': empty m-columns simulated states matrix
  $w^* = w_0$: random weights for the regressor
  $\boldsymbol{\gamma} \in [0, 1]$ discount rate
  $\Delta = 0$: number of actions
Other variables:
  $f()$: regressor function
  m: number of features
  $p_t(x^*)$: stochastic process, function of the last existing chosen state
  $x_t(p_t) \in \mathbb{R}^m$: current state vector
  $\Omega_t(x_t)$: set of possible actions
  $s(x_t) \in \mathbb{R}$: signal value triggered by the modified state
WHILE the learning process is on:
  record x_t
  $X' :=$ empty matrix
  FOR x' in $\Omega_t(x_t)$:
    append x' to X'
  $x^* := \arg\max_{x' \in X'} f(w^*, X')$
  append x^* to X
  IF $s(x^*)$!= 0 :
    FOR δ in 1:Δ:
      $value = \gamma^{\delta-1} s(x^*)$
      append $value$ to y
    $w^* := \arg\min_w(l(f(w, X), y))$
    $\Delta := 0$
  ELSE:
    Δ += 1

Explanation of some elements:

- Ω can be set deterministically. In the context of a game, there is a well determined space of actions. It also can be infinite. A mobile robot can explore the real 3D world in almost infinite ways.
- The signal value function assumes the existence of one or multiple sensors, or the manual assignement of a value. It is conditional on the current environment and takes positive value if a "gain" is sensed, and negative value if a "pain" is sensed.
- The minimization process is a tedious part of the algorithm. When dealing with unstructured features, we might want to use a neural network-type regressor. In that case, W, the set of the hidden layers' weights is initialized with the values of the previous iterations. The optimal W^* should converge as X grows if there are patterns in the stochastic processes.

Remark on the function of the regressor's role

The regressor's task is not to predict accurately, even though it is closely related, but rather to give the "real value" of an action.

Analysis of why RL+NN converge, why it works

It needs to have a good balance between gains and pains

Application

Remark on the function of the regressor's role

Two intermediary movements can ultimately lead to both winning and losing. Such states will get a non-tendencial value as X grows, which is a good thing.

Analysis of game

Do some graph based on data.txt's target variable, observe time effects, wins evolution