Automated optimization of a course of actions in a stochastic environment

Thomas Vicente

General principle

Algorithm 1 General version

initialize:
$f()$: regressor function
x_t: current stochastic environment
$\Omega_t(x_t)$: set of possible actions
$s(x_t)$: signal activated or not by the stochastic environment, initialized to 0
$W^* = W_0$: random weights for the regressor
X: empty training feature matrix
y: empty training target vector
A: empty feature matrix for simulated actions
$\gamma \in [0, 1]$ discount rate
$\Delta = 0$: number of actions
while the learning process is on:
record x_t
A = empty matrix
for a' simulated action in $\Omega_t(x_t)$
$A.append(a')$
$a^* = \arg\max_{a' \in A} f(W^*, A)$
$X.append(a^*)$
if $s(a^*)$!= 0 :
for a in Δ:
$value = \gamma^{a-1} s(a^*)$
$y.append(value)$
$W^* := \arg\min_W (l(f(W, X), y))$
$\Delta = 0$
else:
Δ += 1
continue

The algorithm assumes two underlying function:

- 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.

- The signal 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.

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