CS 6776
Evolutionary Computation

February 13, 2014

Contents

• Assignment 1 grades discussions
• Assignment 2 discussions
• Particle swarm optimization
• Demo
• Algorithm details
• Velocity Clamping

Assignment 2

• Rosenbrock Function is a minimization problem:
  – Q1.4: selection methods should prefer solutions with a smaller \( f(x_1, x_2) \).
  – Q2.1: fitness sharing: \( f=0.5 \), \( sh = 1.87 \)
    • Convert \( f \): \( f'=1/f = 1/0.5 = 2 \)
    • Divide \( f' \) with \( sh \): \( f''=f'/sh = 2/1.87 = 1.07 \)
    • Convert the result back: \( f'''=1/f'' = 1/1.07= 0.935 \)

PSO: Origins

How do birds and fish demonstrate such a coordinated collective behavior?
Reynolds’ Model

- Global behavior is the result of local interaction.
- Each agent is a particle. The behavior of each particle is governed by the following 3 rules:
  - Collision Avoidance: avoid collision with nearby flock mates;
  - Velocity Matching: attempt to match velocity with nearby flock mates;
  - Flock Centering: attempt to stay close to nearby flock mates.

Heppner and Grenander’s Model

- Bird flocking is a local interaction process
- The model includes
  - a roost to attract the flock,
  - a nonlinear attraction to flock mates, preservation of flight velocity.
  - an n-dimensional Poisson stochastic process.

Particle Swarm Optimization

- Agents are collision-proof birds.
- Agents & their associated velocities are initialized randomly.
- There is a roost that attracts the agents.
- At each time step, each agent is stochastically updated using:
  - Its own historically best positions (pbest)
  - The best known position that one member of the flock finds (gbest)
- Simulation result: flocking agents circle around, realistically approaching the target and finally landed on the roost.

Citation and Demo

PSO Algorithm

- A population-based search algorithm.
- Each particle is a candidate solution in an $n$-dimensional search space.
- There is a fitness function to decide the quality of each particle solution.
- At each time step, each particle is stochastically updated:
  - First update the velocity
  - Next use the updated velocity to update the position

General PSO Algorithm

```plaintext
create and initialize an n-dimensional swarm, S;
repeat
  for each particle $i = 1, \ldots, n_s$ do
    // set the personal best position
    if $f(S.x_i) < f(S.y_i)$ then
      $S.y_i = S.x_i$; //y is pbest
    end
    // set the global best position if
    if $f(S.y_i) < f(S.\hat{y})$ then
      $S.\hat{y} = S.y_i$; //\hat{y} is gbest/ lbest
    end
  end
  for each particle $i = 1, \ldots, n_s$ do
    update the particle velocity;
    update the particle position;
  end
until stopping condition is true;
```

PSO vs. EA

- Similarities:
  - Population-based search algorithms.
  - Optimization is an iterative process.
- Differences:
  - Each potential solution in PSO also has a velocity.
  - No genetic inheritance: use social influence (gbest/lbest) to update particles.
  - No selection: all particles in a swarm are updated and carried over to the new generation.

Social vs. Genetic Interaction

- PSO algorithms change velocity to indirectly influence the particle values. At each time step:
  - Each particle’s velocity is updated toward its pbest and gbest/lbest (social interaction) locations.
  - A different weight is given to pbest and gbest/lbest locations.
  - Next, update the current particle position using the updated velocity.
- EA algorithms use crossover and mutation to directly inherit genetic materials of individuals to form new individuals.
Collective vs. Selective Intelligence

- In PSO, all particles in a swarm are updated and carried over to the new generation.
- All particles continue to exist and have potential to influence other particles in the swarm.
- In EA, only selected individuals are maintained to affect future generations.

Global Best PSO: Particle Update

- Let $x_i(t)$ denote the position of particle $i$ in the search space and $v_i(t)$ denote the velocity of $i$ at time step $t$.
- $c1$ & $c2$: weight
- $r1$ & $r2$: random value between 0 and 1.

$$v_i(t) = v_i(t-1) + c_1 \times r_1 \times (pbest(t-1) - x_i(t-1)) + c_2 \times r_2 \times (gbest(t-1) - x_i(t-1))$$

$$x_i(t) = x_i(t-1) + v_i(t)$$

Local Best PSO: Particle Update

- Instead of using the best particle in the entire swarm ($gbest$) to update each particle, the best particle in the local neighborhood (e.g. a ring network topology) ($lbest$) is used to update the particle.

$$v_i(t) = v_i(t-1) + c_1 \times r_1 \times (pbest(t-1) - x_i(t-1)) + c_2 \times r_2 \times (lbest(t-1) - x_i(t-1))$$

$$x_i(t) = x_i(t-1) + v_i(t)$$

PSO Optimization Process

- Driven by velocity (think of it as step size), velocity reflects:
  - Memory of the last flight direction, referred as inertia component.
  - Experiential knowledge of the particle ($pbest$), referred as cognition component.
  - Socially exchanged information from the particle’s neighborhood ($gbest/lbest$), referred as social component.
Velocity Components

- **Inertia component**: It is a term that biases towards the current direction.
- **Cognitive component**: It draws a particle back to its own previous best position.
- **Social component**: It is the group norm or standard that individuals seek to attain.

**Geometric Illustration**

- **Similarity**: The social component of the velocity impacts swarm movement towards the global best particle.
- **Differences**: *gbest* PSO converges faster than *lbest* PSO: each particle has a larger number of inter-connectivity. Information of the best particle quickly filtered through the social network.
  
  *lbest* PSO provides swarms with a larger diversity than *gbest* PSO: information is propagated in a slower pace.

**Particle Diverge**

- Velocity can quickly explode to a large value, especially for particles that are far from the gbest/lbest and pbest positions (why?).
- Consequently, particles have large position updates, which result in particles leaving the boundaries of the search space – the particle diverge.
Velocity Clamping

• One way to control the global exploration of particles, velocities are clamped to stay within a maximum velocity $V_{\text{max},j}$ for each dimension $j$ of a particle:

$v_j(t+1) = \begin{cases} 
  v_j(t+1), v_j(t+1) < v_{\text{max},j} \\
  v_{\text{max},j}, v_j(t+1) \geq v_{\text{max},j} 
\end{cases}$

– Large $v_{\text{max},j}$: exploration vs. exploitation?
– Small $v_{\text{max},j}$: exploration vs. exploitation?

Fixed Maximum Velocity

• Maximum velocity is normally set to be of a certain percentage of the value range.

$V_{\text{max},j} = \delta(x_{\text{max},j} - x_{\text{min},j}), \delta \in (0,1]$  

• Velocity clamping does not guarantee the particle always stay within the boundaries of the search space; when the particle value is outside the boundaries, map it to the boundary values.

Velocity Clamping Dynamics

• Velocity clamping changes both the step size and the direction of a particle’s movement.

– $x(t-1)=(3,2)$
– $v(t)=(3,10)$
– $V_{\text{max}}=5$
– $v'(t)=(3,5)$ //clamped $v$
– $x(t)=(3,2)+(3,10)=(6,12)$
– $x'(t)=(3,2)+(3,5)=(6,7)$

Dynamic $V_{\text{max}}$ Adjustment

• We can adjust velocity (step size) dynamically.

• For example, decrease $V_{\text{max},j}$ when gbest has no improvement after a certain number of iterations [Schutte and Groenwold, 2003]

• Rule of thumb: first explore and then gradually reduce $V_{\text{max},j}$ over time.
Linear vs. Exponential decreasing

• Exponential decreasing $V_{max,j}$[Fang, 2002]
  $V_{max,j}(t + 1) = (1 - (t/n)^α) \times V_{max,j}(t)$

• Linear decreasing:
  $V_{max}(t) = V_{max}(n) + (V_{max}(0) - V_{max}(n)) \times \frac{n-t}{n}$