15-382 Collective Intelligence - S19

Lecture 22:
Swarm Intelligence 3 / Classical Optimization

Teacher:
Gianni A. Di Caro
WHAT IF WE HAVE ONE SINGLE AGENT ...

- PSO leverages the presence of a swarm: the outcome is truly a collective behavior.
- If left alone, each individual agent would behave like a hill-climber when moving in the direction of a local optimum, and then it will have a quite hard time to escape it.

\[ \vec{v}_i \leftarrow \chi \cdot \left( \vec{v}_i + \vec{U}(0, \phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2) \otimes (\vec{g}_i - \vec{x}_i) \right) \]

A single agent doesn’t look impressive 🤔 ...

How can a single agent be smarter?
A (FIRST) GENERAL APPROACH

- What is a good direction \( p \)?
- How small / large / constant / variable should be the step size \( \alpha_k \)?
- How do we check that we are at the minimum?
- This could work for a local optimum, but what about finding the global optimum?

\[
k = 0 \\
\text{while } x_k \text{ is not a minimum} \\
\text{compute step direction } p_k \text{ with } \|p_k\| = 1 \\
\text{compute step size } \alpha_k \\
x_{k+1} = x_k + \alpha_k p_k \\
k = k + 1 \\
\text{end.}
\]
IF WE HAVE THE FUNCTION (AND ITS DERIVATIVES)

\[ f(x) \approx f(x_0) + \nabla^T f(x_0) \cdot (x - x_0) \]

- From 1st order Taylor series: Equation of the tangent plane to \( f \) in \( x_0 \): the gradient vector is orthogonal to the tangent plane.
- The partial derivatives determine the slope of the plane.
- In each point \( x \), the gradient vector is orthogonal to the isocountours, \( \{ x : f(x) = c \} \).
- \( \Rightarrow \) The gradient vector points in the direction of maximal change of the function in the point. The magnitude of the gradient is the (max) velocity of change for the function.
GRADIENTS AND RATE OF CHANGE

• **Directional derivative** for a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the rate of change of the function along a given direction $v$ (where $v$ is a unit-norm vector):

$$D_vf(x) = \lim_{h \rightarrow 0^+} \frac{f(x + hv) - f(x)}{h}$$

• E.g., for a function of two variables and a direction $v=(v_x, v_y)$

$$D_vf(x_0, y_0) = \lim_{h \rightarrow 0^+} \frac{f(x_0 + hv_x, y_0 + hv_y) - f(x_0, y_0)}{h}$$

• **Partial derivatives** are directional derivatives along the vectors of the canonical basis of $\mathbb{R}^n$
GRADIENTS AND RATE OF CHANGE

- **Theorem:** Given a direction \( \mathbf{v} \) and a differentiable function \( f \), then in each point \( x_0 \) of the function domain the following relation exists between the gradient vector and the directional derivative:

\[
D_vf(x_0) = \nabla f(x_0) \cdot \mathbf{v}
\]

- **Corollary 1:**

\[
D_vf(x_0) = \nabla f(x_0) \cdot \mathbf{v} = ||\nabla f(x_0)|| \cos \theta \quad \Rightarrow \quad D_vf(x_0) \leq ||\nabla f(x_0)||
\]

- **Corollary 2:**

\[
D_vf(x_0) = ||\nabla f(x_0)|| \quad \text{only when the gradient is parallel to the directional derivative (} \theta = 0 \text{)}
\]

that is, the directional derivative in a point gets its maximum value when the direction \( \mathbf{v} \) is the same as the gradient vector

- \( \Rightarrow \) In each point, the gradient vector corresponds to the direction of maximal change of the function

- \( \Rightarrow \) The norm of the gradient corresponds to the max velocity of change in the point
- Move in the direction opposite to (min) to or aligned with (max) the gradient vector, the direction of maximal change of the function.
The final local optimum depends on where we start from (for non-convex functions)
GD run ~ Motion of a mass in a potential field towards the minimum energy configuration. At each point the gradient defines the attraction force, while the step $\alpha$ scales the force, to define the next point.
GRADIENT DESCENT ALGORITHM (MIN)

1. **Initialization**
   
   (a) Definition of a starting point \( x_0 \)
   
   (b) Definition of a tolerance parameter for convergence \( \epsilon \)
   
   (c) Initialization of the iteration variable, \( k \leftarrow 0 \)

2. **Computation of a feasible direction for moving**
   
   \( d_k \leftarrow -\nabla_k f(x_k) \)

3. **Definition of the feasible (max) length of the move**
   
   \( \alpha_k \leftarrow \min_{\alpha} f(x_k + \alpha d_k) \)
   
   (1-dimensional problem in \( \alpha \in \mathbb{R} \), the farthest point to where \( f(x_k + \alpha d_k) \) keeps increasing)

4. **Move to new point in the direction of gradient descent**
   
   \( x_{k+1} \leftarrow x_k + \alpha_k d_k \)

5. **Check for convergence**
   
   - If \( \|\alpha_k d_k\| < \epsilon \) [or, if \( \alpha_k \leq c\alpha_0 \), where \( c > 0 \) is a small constant] (i.e., gradient becomes zero)
     
     (a) Output: \( x^* = x_{k+1}, f(x^*) = f(x_{k+1}) \)
     
     (b) Stop

   - Otherwise, \( k \leftarrow k + 1 \), and go to Step 2
STEP BEHAVIOR

• If the gradient in $x_0$ points to the local minimum, then a line search would determine a step size $\alpha_k$ that would take directly to the minimum

• However, this lucky case only happens in perfectly conditioned functions, or for a restricted set of points

• It might be heavy to solve a 1D optimization problem at each step, such that some approximate methods can be preferred

• In the general case, the moving directions $d_k$ are perpendicular to each other:

$$df(x_k + \alpha d_k) \over d\alpha = \sum^n \partial f(x_k + \alpha d_k) \over \partial x^i d(x_k^i + \alpha d_k^i) \over d\alpha$$

$$= \sum^n \nabla_i f(x_k + \alpha d_k)d_k^i = \nabla f(x_k + \alpha d_k)^T d_k$$

$\alpha^*$ minimizes $f(x_k + \alpha^* d_k)$, such that $df/d\alpha$ must be zero in $\alpha^*$

$$\Rightarrow \nabla f(x_k + \alpha^* d_k)^T d_k = 0 \quad \text{but,} \quad d_{k+1} = -\nabla f(x_k + \alpha^* d_k), \quad \Rightarrow d_{k+1}^T d_k = 0$$
If the function is very *anisotropic*, then the problem is said **ill-conditioned**, since the gradient vector doesn’t point to the direction of the local minimum, resulting into a zig-zagging trajectory.

- Ill-conditioning can be determined by computing the **ratio between the eigenvalues of the Hessian matrix** (the matrix of the second partial derivatives).
WHAT ABOUT A CONSTANT STEP SIZE?

Small, good $\alpha$, convergence

$\alpha$ is too large, divergence
WHAT ABOUT A CONSTANT STEP SIZE?

- Adapting the step size *may* be necessary to avoid either a too slow progress, or overshooting the target minimum.
Gradient descent to solve optimization problem

\[
\text{minimize } \sum_{i=1}^{m} \left( (x^{(i)T}\theta - y^{(i)}) \right)^2
\]

- Loss function: Sum of squared errors:
- \( m \) labeled training samples (examples)
- \( y^{(i)} \) = known correct value for sample \( i \)
- \( x^{(i)} \cdot \theta = \) linear hypothesis function, \( \theta \) vector of parameters
- **Goal:** find the value of the parameter vector \( \theta \) such that the loss (errors in classification / regression) is minimized (over the training set)
- Any analogy with PSO?

Gradient:

\[
\nabla_{\theta} \sum_{i=1}^{m} \left( (x^{(i)T}\theta - y^{(i)}) \right)^2 = \sum_{i=1}^{m} \nabla_{\theta} \left( (x^{(i)T}\theta - y^{(i)}) \right)^2 = 2 \sum_{i=1}^{m} x^{(i)} \left( x^{(i)T}\theta - y^{(i)} \right)
\]

Gradient descent, repeat:

\[
\theta \leftarrow \theta - \alpha \sum_{i=1}^{m} x^{(i)} \left( x^{(i)T}\theta - y^{(i)} \right)
\]

If the averaging factor \( 1/2m \) is used, then the update action becomes:

\[
\theta \leftarrow \theta - \frac{\alpha}{m} \sum_{i=1}^{m} x^{(i)} \left( x^{(i)T} \cdot \theta - y^{(i)} \right)
\]
function $\theta = \text{Gradient-Descent}(\{(x^{(i)}, y^{(i)})\}, h_\theta, \ell, \alpha)$

Initialize: $\theta \leftarrow 0$

Repeat until convergence

$g \leftarrow 0$

For $i = 1, \ldots, m$:

$g \leftarrow g + \nabla_\theta \ell(h_\theta(x^{(i)}), y^{(i)})$

$\theta \leftarrow \theta - \alpha g$

return $\theta$

---

function $\theta = \text{SGD}(\{(x^{(i)}, y^{(i)})\}, h_\theta, \ell, \alpha)$

Initialize: $\theta \leftarrow 0$

Repeat until convergence

For $i = 1, \ldots, m$ (randomly shuffle the order):

$\theta \leftarrow \theta - \alpha \nabla_\theta \ell(h_\theta(x^{(i)}), y^{(i)})$

return $\theta$

---

- If the number of samples $m$ is large, computing a single gradient step is costly.
- An alternative approach, **stochastic gradient descent (SGD)**, update the parameters based upon gradient each sample:

- It takes many more steps along noisy estimates of the gradient, and often converges to a "good" parameter value after relatively few passes over the data set.

Function value does not decrease monotonically.
ADAPTIVE STEP SIZE?

- Problem \( \min_{\alpha} f(x_k + \alpha d_k) \) can be tackled using any algorithm for one-dimensional search: Fibonacci, Golden section, Newton’s method, Secant method, ...

- If \( f \in C^2 \), it’s possible to compute the Hessian \( H_k \), and \( \alpha_k \) can be determined analytically:
  
  - Let \( y_k = \alpha d_k \) be the (feasible) small displacement applied in \( x_k \), such that from Taylor’s series about \( x_k \), truncated at the 2nd order (quadratic approximation):
    
    \[
    f(x_k + y_k) \approx f(x_k) + y_k^T \nabla_k + \frac{1}{2} y_k^T H_k y_k
    \]

  - In the algorithm, \( d_k \) is the direction of steepest descent, \( y_k = -\alpha \nabla_k \), such that:
    
    \[
    f(x_k - \alpha \nabla_k) \approx f(x_k) - \alpha \nabla_k^T \nabla_k + \frac{1}{2} \alpha^2 \nabla_k^T H_k \nabla_k
    \]

  - We are looking for \( \alpha \) minimizing \( f \rightarrow \) first order conditions must be satisfied, \( \frac{df}{d\alpha} = 0 \):
    
    \[
    \frac{df(x_k - \alpha \nabla_k)}{d\alpha} \approx -\nabla_k^T \nabla_k + \alpha \nabla_k^T H_k \nabla_k = 0 \quad \Rightarrow \quad \alpha = \alpha_k \approx \frac{\nabla_k^T \nabla_k}{\nabla_k^T H_k \nabla_k}
    \]

  - Updating rule of the gradient descent algorithm becomes:

    \[
    x_{k+1} \leftarrow x_k - \frac{\nabla_k^T \nabla_k}{\nabla_k^T H_k \nabla_k} \nabla_k
    \]
Conditions on the Hessian Matrix

- The value found for $\alpha_k$ is a sound estimate of the "correct" value of $\alpha$ as long as the quadratic approximation of the Taylor series is an accurate approximation of the $f(x)$.

- At the beginning of the iterations, $\|y_k\|$ will be quite large since the approximation will be inaccurate, but getting closer to the minimum $\|y_k\|$ will decrease accordingly, and the accuracy will keep increasing.

- If $f$ is a quadratic function, the Taylor series is exact, and we can use $\approx$ instead of $\Rightarrow$ $\alpha_k$ is exact at each iteration

- The Hessian matrix is the matrix of 2nd order partial derivatives.
  E.g., for $f : X \subseteq \mathbb{R}^3 \mapsto \mathbb{R}$, the Hessian matrix computed in a point $x^*$:

$$
H|_{x^*} = \begin{pmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\
\frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\
\frac{\partial^2 f}{\partial x_1 \partial x_3} & \frac{\partial^2 f}{\partial x_2 \partial x_3} & \frac{\partial^2 f}{\partial x_3^2}
\end{pmatrix}
$$
PROPERTIES OF THE HESSIAN MATRIX

- **Theorem (Schwartz):** If a function is $C^2$ (twice differentiable with continuity) in $\mathbb{R}^n \rightarrow$ The order of derivation is irrelevant. $\Rightarrow$ The Hessian matrix is symmetric

- **Theorem (Quadratic form of a matrix):** Given a matrix $H \in \mathbb{R}^{n \times n}$, squared and symmetric, the associated quadratic form is defined as the function:

$$q(x) = \frac{1}{2} x^T H x$$

The matrix is said:

- **Positive definite** if $x^T H x > 0$, $\forall x \in \mathbb{R}^n$, $x \neq 0$
- **Positive semi-definite** if $x^T H x \geq 0$, $\forall x \in \mathbb{R}^n$
- **Negative definite** if $x^T H x < 0$, $\forall x \in \mathbb{R}^n$, $x \neq 0$
- **Negative semi-definite** if $x^T H x \leq 0$, $\forall x \in \mathbb{R}^n$
- **Indefinite** if $x^T H x > 0$ for some $x$ and $x^T H x < 0$ for others $x$
**Theorem (Use of the eigenvalues):** A matrix \( H \in \mathbb{R}^{n \times n} \) is positive definite (semi-definite) if and only if all its eigenvalues are positives (non negatives).

*Proof sketch:* For a diagonal matrix \( D = \text{diag}(d_1, \ldots, d_n) \) where each diagonal entry is positive, the theorem holds, since \( x^T D x = \sum d_i x_i^2 > 0 \) (unless \( x = 0 \)).

Since \( H \) is real and symmetric (Hermitian), the spectral theorem says that it is diagonalizable using the eigenvector matrix \( E \) as orthonormal basis for the transformation: \( E^{-1} A E = \Lambda \), where the elements of the diagonal matrix \( \Lambda \) are the eigenvalues of \( A \).

Therefore, \( A \) is positive (negative) definite if all its eigenvalues are positive (negative).

**Theorem (Hessian and min/max points):** Given \( x^* \), a *stationary point of* \( f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R} \) (i.e., a point where \( \nabla f(x^*) = 0 \)), and given \( f \)'s Hessian matrix \( H(f) \), evaluated in in \( x^* \), the following conditions are *sufficient* to determine the nature of \( x^* \) as an extremum of the function:

1. If \( H \) is positive definite \( \rightarrow x^* \) is a *minimum* (local or global)
2. If \( H \) is negative definite \( \rightarrow x^* \) is a *maximum* (local or global)
3. If \( H \) has eigenvalues of opposite sign \( \rightarrow x^* \) is (in general), a *saddle*

If \( H \) semi-definite, positive or negative, is more complex to analyze . . .
CONVERGENCE SPEED OF GRADIENT DESCENT

- Previous theorem says that in the neighborhood of a local minimum any function $C^2$ is approximated by a positive definite quadratic form: the function is strictly convex.

- The Hessian matrix is symmetric $\rightarrow$ The quadratic form is also symmetric.

- In 2D $\rightarrow$ The isocurves are ellipsoids:
  - Axes are oriented as the eigenvectors of $H$.
  - Length is proportional to $1/\sqrt{\lambda_1}$ $1/\sqrt{\lambda_2}$.

Convergence speed depends on the shape of the ellipsoids:

- *Spherical*: fast convergence
- *Unbalanced*: slow convergence

Convergence rate ($p=1$, linear):

$$\lim_{k \to +\infty} \frac{|x_{k+1} - \hat{x}|}{|x_k - \hat{x}|^p} = \rho, \quad \rho \in [0, +\infty[$$

For a quadratic function, $p=1$:

$$\rho = \left(\frac{1-r}{1+r}\right)^2, \quad r = \frac{\lambda_{\min}(H_k)}{\lambda_{\max}(H_k)}$$
• $H = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, from the characteristic equation, $\det(H - \lambda I) = 0$ we get: $\lambda^2 - 4\lambda + 3 = 0 \rightarrow \lambda_1 = 1, \lambda_2 = 3$.

• For the eigenvectors, $Hx = \lambda x$:

  $\lambda_1 = 1$: $Hx = x \rightarrow (H - I)x = 0$, that reduces to equation $x_2 = -x_1$
  $\Rightarrow$ all $\lambda_1$’s eigenvectors are $\begin{pmatrix} 1 \\ -1 \end{pmatrix} x_1$

  $\lambda_2 = 3$: $Hx = x \rightarrow (H - 3I)x = 0$, that reduces to equation $x_2 = x_1$
  $\Rightarrow$ all $\lambda_2$’s eigenvectors are $\begin{pmatrix} 1 \\ 1 \end{pmatrix} x_1$

• There are two directions defined by the eigenvectors: $x_2 = -x_1$ and $x_2 = x_1$

• Isolines of the quadratic form: $x^T H x = k \rightarrow 2x_1^2 + 2x_1x_2 + 2x_2^2 = k$

• We can transform it in the canonical form, in the eigenvector basis $\rightarrow \frac{x}{\lambda_1} + \frac{y}{\lambda_2} = k$, which is an ellipse oriented in the direction of the eigenvectors and semi-axes of length $\sqrt{k/\lambda_1}, \sqrt{k/\lambda_2}$
Let’s assume to have a local estimate $\hat{\alpha}$ of the values $\alpha_k$, and let’s denote $f(\mathbf{x}_k)$ as $f_k$, and $f(\mathbf{x}_k - \hat{\alpha}\nabla_k)$ as $\hat{f}_k$. From the previous 2nd order truncated Taylor’s series:

$$\hat{f}_k = f(\mathbf{x}_k - \hat{\alpha}\nabla_k) \approx f_k - \hat{\alpha}\nabla^T_k\nabla_k + \frac{1}{2}\nabla^T_k H_k \nabla_k$$

$$\Rightarrow \nabla^T_k H_k \nabla_k \approx \frac{2(\hat{f}_k - f_k + \hat{\alpha}\nabla^T_k\nabla_k)}{\hat{\alpha}^2}$$

$$\Rightarrow \alpha_k \approx \frac{\nabla^T_k \nabla_k \hat{\alpha}^2}{2(\hat{f}_k - f_k + \hat{\alpha}\nabla^T_k\nabla_k)}$$

$$\Rightarrow \mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \frac{2(\hat{f}_k - f_k + \hat{\alpha}\nabla^T_k\nabla_k)}{\hat{\alpha}^2} \nabla_k$$

A reasonable choice for $\hat{\alpha}$ is $\alpha_{k-1}$: the (optimal) value taken by $\alpha$ in the previous iteration, starting with $\alpha_0 = 1$

Also in this case the approximation is more or less good depending on how the 2nd order Taylor series is an accurate approximation of $f(\mathbf{x})$

**Moral:** Isn’t easier (more effective / general) to use and play with PSO?