Optimisation algorithms in Statistics II, lecture 2

Frank Miller, Department of Statistics; Stockholm University

April 13, 2021

Stockholm University

Course schedule

- Topic 1: **Stochastic gradient descent and quasi-Newton algorithms**
  Lectures: March 23; Time 10-12, 13-15 (online, Zoom)
- Topic 2: **Particle swarm optimisation and stochastic gradient descent with momentum**
  Lectures: April 13; Time 10-12, 13-15
- Topic 3: **Simulated annealing and genetic algorithms**
  Lectures: April 27; Time 10-12, 13-15

Course homepage:
http://gauss.stat.su.se/phd/oasi/optimisation2.html

Includes reading material, lecture notes, assignments

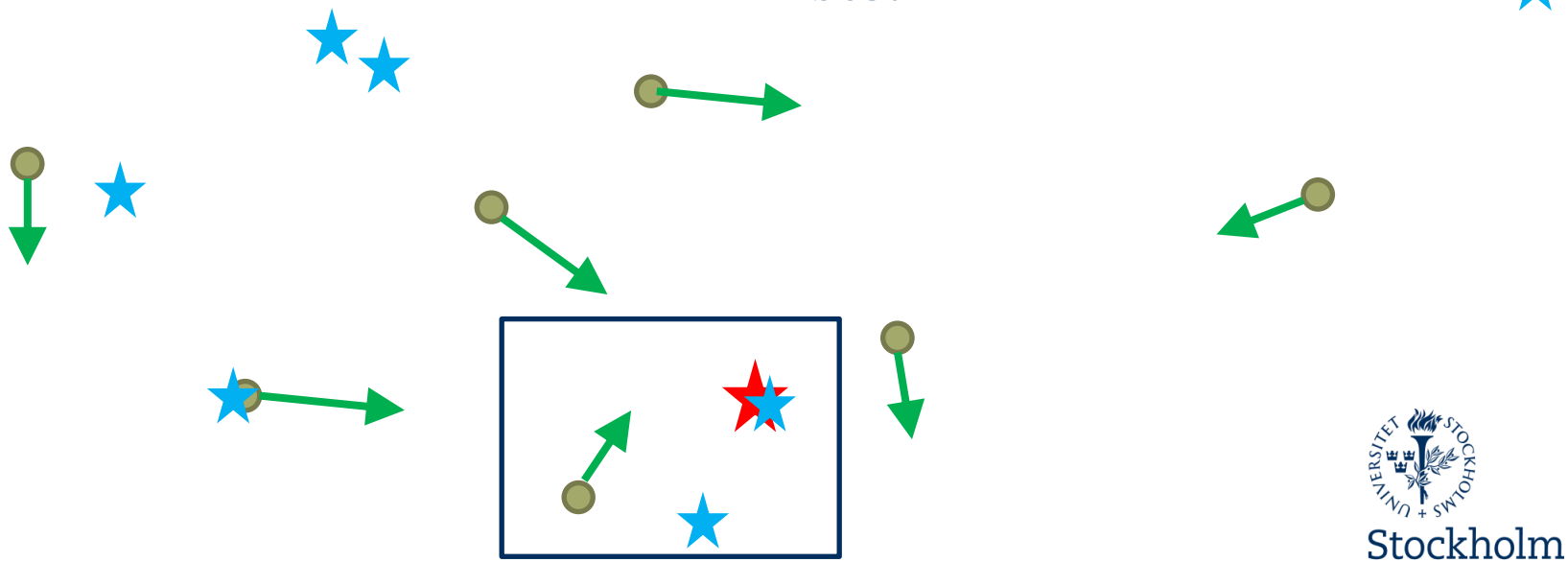
Stockholm University

Today's schedule

- Particle swarm optimisation (PSO)
  - Quick recap
  - Details on different versions
  - Stability analysis for PSO
  - Empirical studies
  - Exploration and exploitation
- Stochastic gradient descent (SGD) algorithms with momentum (SGDM)
  - Quick recap and some remarks

Particle swarm optimisation – basic versions

- Swarm of N particles
  - Position of particle i at iteration t+1: $\boldsymbol{x}_i^{(t+1)}$
  - Velocity of particle i at iteration t+1: $\boldsymbol{v}_i^{(t+1)}$
- Best positions found so far:
  - Best location found by particle i: $\boldsymbol{p}_{\text{best},\,i}^{(t)}$
  - Global best solution found: $\boldsymbol{g}_{\text{best}}^{(t)}$

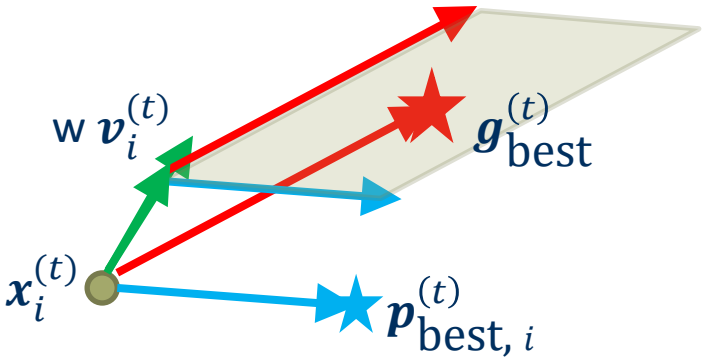


Particle swarm optimisation – basic versions

- Movement of particle i at iteration t+1:
  - $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}$
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$

inertia weight — cognitive component — social component

- $R_1^{(t+1)}$ and $R_2^{(t+1)}$ are uniformly distributed, **`runif()`**



Stockholm University

Particle swarm optimisation – basic versions

- PSO first suggested: 1995 by Kennedy and Eberhart
- Clerc (2016) distinguishes following (main) versions:
  - 1998. A basic version
  - SPSO 2007 (”Standard PSO”)
  - SPSO 2011

Particle swarm optimisation – inertia weight

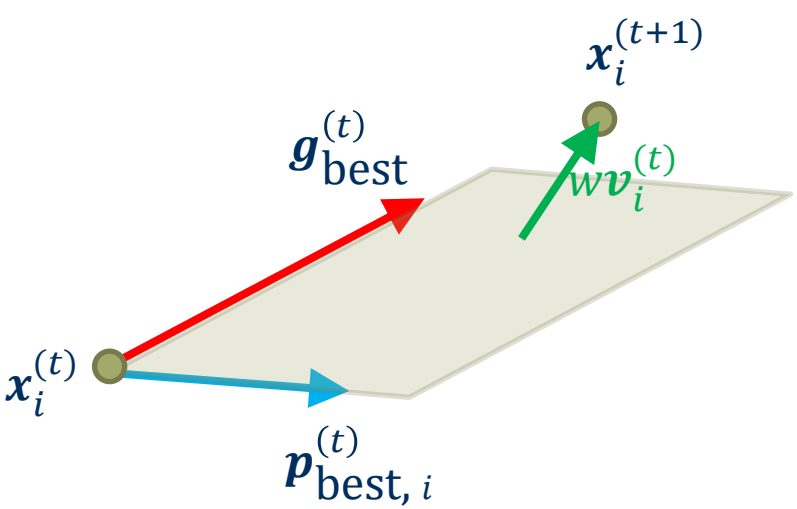
- Movement of particle i at iteration t+1:
  - $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}$
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- In the first version from 1995, the inertia weight w was not included
- Particle swarm might "explode"
- Explosion can be prevented by introducing maximum velocity
- Alternatively, inertia weight w < 1 can prevent explosion
- Included in basic version from 1998

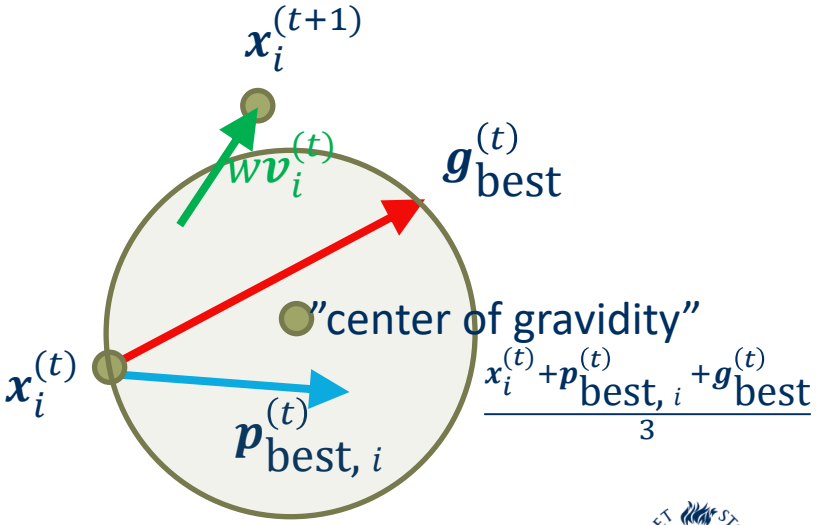
Particle swarm optimisation – dimensions

- In first versions including 1998-basic version and SPSO 2007, random variables applied for each dimension separately:
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1\boldsymbol{R}_1^{(t+1)} \otimes \left(\boldsymbol{p}_{\mathrm{best},\,i}^{(t)} - \boldsymbol{x}_i^{(t)}\right) + c_2\boldsymbol{R}_2^{(t+1)} \otimes \left(\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)}\right)$
    where $\otimes$ is componentwise multiplication and $\boldsymbol{R}_k^{(t+1)}$ are vectors
- `v[i] <- w*v[i] + c1*runif(p)*(pbest[i]-x[i]) + c2*runif(p)*(gbest-x[i])`
  where `v[i], x[i], pbest[i], gbest` vectors for each `i`
- In SPSO 2011, same random variable used for all dimensions leading to movement in hyperspheres:
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\,i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- `v[i] <- w*v[i] + c1*runif(1)*(pbest[i]-x[i]) + c2*runif(1)*(gbest-x[i])`

Particle swarm optimisation – dimensions

- Velocity of particle i at iteration t+1:
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\,i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- In SPSO 2011, same random variable used for all dimensions leading to movement in hyperspheres





Stockholm University

Particle swarm optimisation – dimensions

- With second version (SPSO 2011), particles can move only in hyperspace spanned by starting particles
- Disadvantages:
  - If dimension of problem p is large in relation to swarm size s, e.g. p>s, optimisation done only in a subspace and high risk that optimum is missed
  - Even if starting particles well distributed, they might become close to a hyperspace after some iterations
- Advantages:
  - Problem with dependence on coordinate system and with "biased search" is reduced; it has been shown that optima along axes and diagonal found more easily (Clerc, 2016)
  - Linearly constrained problems can easily be handled (see next slide from OASI, L4)

Optimisation with equality constraints – modification of algorithms

From OASI, L4

- Constrained optimisation problem:
  - $\boldsymbol{x}$ p-dimensional vector, g: $\mathbb{R}^p \to \mathbb{R}$ function
  - We search $\boldsymbol{x}^*$ with $g(\boldsymbol{x}^*) = \max g(\boldsymbol{x})$
  - Subject to $\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b} = \boldsymbol{0}$, $\boldsymbol{A} \in \mathbb{R}^{m \times p}, \boldsymbol{b} \in \mathbb{R}^m$ (linear equality constraints)
- Example: Particle Swarm Optimisation (see L3)
- Movement of particle *i* at iteration *t*+1:
  - $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}$
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- $R_1^{(t+1)}$ and $R_2^{(t+1)}$ are uniformly distributed, `runif()`
- Ensure that $\boldsymbol{A}\boldsymbol{x}_i^{(0)} = \boldsymbol{b}$ and $\boldsymbol{A}\boldsymbol{v}_i^{(0)} = \boldsymbol{0}$, then $\boldsymbol{A}\boldsymbol{x}_i^{(t)} = \boldsymbol{b}$ for all *i* and *t*

Particle swarm optimisation – choice of hyperparameters using stability analyses

- Velocity of particle i at iteration t+1:
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- Hyperparameters to choose: w, c_1, c_2
- Particles should not diverge
- "Stability analyses" had been done – these are simplified analytical computations, **for example**:
  - Assume one dimensional case,
  - Assume static $\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} = \boldsymbol{p}_{\mathrm{best},\, i}$ and $\boldsymbol{g}_{\mathrm{best}}^{(t)} = \boldsymbol{g}_{\mathrm{best}}$ ("stagnation assumption")
  - Ignore randomness (replace $R_k^{(t+1)}$ by expected value ½)
- Derive requirements for w, c_1, c_2 such that $\boldsymbol{x}_i^{(t)}$ "converges"

Particle swarm optimisation – choice of hyperparameters using stability analyses

- Velocity of particle i at iteration t+1:
  - $\boldsymbol{v}_i^{(t+1)} = \mathrm{w}\boldsymbol{v}_i^{(t)} + c_1 R_1^{(t+1)} (\boldsymbol{p}_{\mathrm{best},\, i}^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 R_2^{(t+1)} (\boldsymbol{g}_{\mathrm{best}}^{(t)} - \boldsymbol{x}_i^{(t)})$
- Standard choice in SPSO 2007, based originally on stability analyses from Clerc and Kennedy (2002):
  - $w = \frac{1}{2\ln(2)} = 0.721,$
  - $c_1 = c_2 = \frac{1}{2} + \ln(2) = 1.193$
- Since deterministic $R_k^{(t+1)} = \frac{1}{2}$ and static $\boldsymbol{p}_{\mathrm{best}}$, $\boldsymbol{g}_{\mathrm{best}}$ are used in stability analyses, no distinctive requirements for c_1 and c_2 are obtained and a default is often just $c_1 = c_2$
- Write now $C_k^{(t+1)} = c_k R_k^{(t+1)} \sim Unif[0, c_k], k = 1,2.$

Particle swarm optimisation – stability analyses

- Movement of specific particle at iteration t+1 (drop index i):
  - $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \boldsymbol{v}^{(t+1)}$
  - $\boldsymbol{v}^{(t+1)} = \mathrm{w}\boldsymbol{v}^{(t)} + C_1^{(t+1)}(\boldsymbol{p}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)}) + C_2^{(t+1)}(\boldsymbol{g}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)})$
- Focusing on particle locations, we can describe PSO as:

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \boldsymbol{v}^{(t+1)}$$
$$= \boldsymbol{x}^{(t)} + \mathrm{w}\boldsymbol{v}^{(t)} + C_1^{(t+1)}\left(\boldsymbol{p}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)}\right) + C_2^{(t+1)}\left(\boldsymbol{g}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)}\right)$$
$$= \boldsymbol{x}^{(t)} + \mathrm{w}\left(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\right) + C_1^{(t+1)}\left(\boldsymbol{p}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)}\right) + C_2^{(t+1)}\left(\boldsymbol{g}_{\text{best}}^{(t)} - \boldsymbol{x}^{(t)}\right)$$
$$= \boldsymbol{x}^{(t)}\left(1 + w - C_1^{(t+1)} - C_2^{(t+1)}\right) - \mathrm{w}\boldsymbol{x}^{(t-1)} + C_1^{(t+1)}\boldsymbol{p}_{\text{best}}^{(t)} + C_2^{(t+1)}\boldsymbol{g}_{\text{best}}^{(t)}$$

$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \boldsymbol{v}^{(t)}$

- Therefore, a single equation is sufficient to describe the PSO iterations ($\boldsymbol{x}^{(t+1)}$ depends then on both $\boldsymbol{x}^{(t)}$ and $\boldsymbol{x}^{(t-1)}$)

Stockholm University

Particle swarm optimisation – stability analyses

- Movement of specific particle at iteration t+1 with PSO:
  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}\left(1 + w - C_1^{(t+1)} - C_2^{(t+1)}\right) - \mathrm{w}\boldsymbol{x}^{(t-1)} + C_1^{(t+1)}\boldsymbol{p}_{\mathrm{best}}^{(t)} + C_2^{(t+1)}\boldsymbol{g}_{\mathrm{best}}^{(t)}$
- Stability analyses were improved during the two previous decades, see Bonyadi and Michalewicz (2016) and Cleghorn and Engelbrecht (2018); definitions below follow the latter
- Order-1 stability
  A sequence $(\boldsymbol{x}^{(t)})$ of p-dimensional random variables is called *order-1 stable* if $\mathrm{E}\left[\boldsymbol{x}^{(t)}\right] \to \boldsymbol{x}_E$ for some $\boldsymbol{x}_E$
- Order-2 stability
  A sequence $(\boldsymbol{x}^{(t)})$ of p-dimensional random variables is called *order-2 stable* if $\mathrm{Var}\left[\boldsymbol{x}^{(t)}\right] \to \boldsymbol{x}_V$ for some $\boldsymbol{x}_V$

Particle swarm optimisation – stability analyses

- Movement of specific particle at iteration t+1 with PSO:
  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}\left(1 + w - C_1^{(t+1)} - C_2^{(t+1)}\right) - \mathrm{w}\boldsymbol{x}^{(t-1)} + C_1^{(t+1)}\boldsymbol{p}_{\text{best}}^{(t)} + C_2^{(t+1)}\boldsymbol{g}_{\text{best}}^{(t)}$
- Bonyadi and Michalewicz (2016) interpret each of $C_1^{(t+1)}, C_2^{(t+1)}$, $\boldsymbol{p}_{\text{best}}^{(t)}, \boldsymbol{g}_{\text{best}}^{(t)}$ as iid random variables
- This generalises assumptions that these values are fixed values; it weakens the stagnation assumption
- The iid assumption for $\boldsymbol{p}_{\text{best}}^{(t)}, t = 1, \ldots$ and for $\boldsymbol{g}_{\text{best}}^{(t)}, t = 1, \ldots$ still need to be seen as approximations

Particle swarm optimisation – stability analyses

- We consider the one-dimensional case (p=1) now
- Movement of specific particle at iteration t+1 with PSO:
  $x^{(t+1)} = x^{(t)}\left(1 + w - C_1^{(t+1)} - C_2^{(t+1)}\right) - \mathrm{w}x^{(t-1)} + C_1^{(t+1)} p_{\text{best}}^{(t)} + C_2^{(t+1)} g_{\text{best}}^{(t)}$

- To write the iterations as a linear one-step relation, we write
  $$\boldsymbol{z}^{(t+1)} = (x^{(t+1)}, x^{(t)})^T, \qquad U = 1 + w - C_1^{(t+1)} - C_2^{(t+1)},$$
  and
  $$\boldsymbol{z}^{(t+1)} = \begin{pmatrix} U & -w \\ 1 & 0 \end{pmatrix} \boldsymbol{z}^{(t)} + \begin{pmatrix} C_1^{(t+1)} p_{\text{best}}^{(t)} + C_1^{(t+1)} g_{\text{best}}^{(t)} \\ 0 \end{pmatrix}$$
- Since U and $\boldsymbol{z}^{(t)}$ are independent, we have
  $$E\boldsymbol{z}^{(t+1)} = \begin{pmatrix} EU & -w \\ 1 & 0 \end{pmatrix} E\boldsymbol{z}^{(t)} + \begin{pmatrix} E\left[C_1^{(t+1)} p_{\text{best}}^{(t)}\right] + E\left[C_1^{(t+1)} g_{\text{best}}^{(t)}\right] \\ 0 \end{pmatrix}$$
- ➢ Sequence $E\boldsymbol{z}^{(t+1)}$ is of form $E\boldsymbol{z}^{(t+1)} = \boldsymbol{M}E\boldsymbol{z}^{(t)} + \boldsymbol{b}$

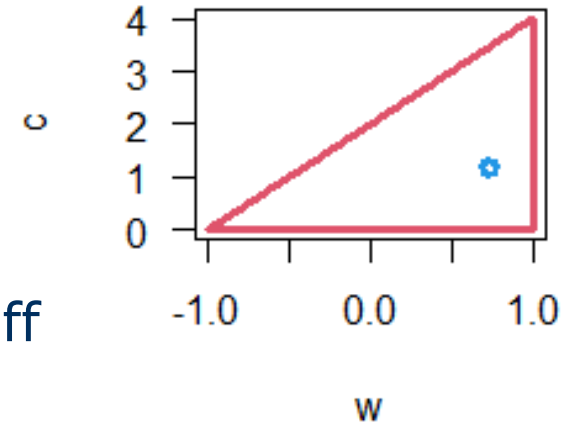
Particle swarm optimisation – stability analyses

- Sequence $E\boldsymbol{z}^{(t+1)}$ is of form $E\boldsymbol{z}^{(t+1)} = \boldsymbol{M}E\boldsymbol{z}^{(t)} + \boldsymbol{b}$
- Functional analysis says that $E\boldsymbol{z}^{(t)}$ converges if the spectral radius of $\boldsymbol{M}$ is <1, see Bonyadi and Michalewicz (2016)'s Lemma 1
- Spectral radius $\rho(\boldsymbol{M})$ of $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ is $\rho(\boldsymbol{M}) = \max\{|\lambda_1|, \ldots, |\lambda_p|\}$ where λ_j are the p (real or complex) eigenvalues of $\boldsymbol{M}$
- Recall that a non-symmetric $\mathbb{R}^{p\times p}$ matrix still has p eigenvalues as long as we allow for complex eigenvalues
- If $\lambda = r + c\mathrm{i}$ then $|\lambda| = \sqrt{r^2 + c^2}$; `R` can cope with this easily:

```
> M <- matrix(c(-0.66, 1, -0.72, 0), ncol=2)
> eigen(M)$values
[1] -0.33+0.7817289i -0.33-0.7817289i
> max(abs(eigen(M)$values))  # spectral radius
[1] 0.8485281
```

Particle swarm optimisation – stability analyses

- We have
$$E\boldsymbol{z}^{(t+1)} = \begin{pmatrix} EU & -w \\ 1 & 0 \end{pmatrix} E\boldsymbol{z}^{(t)} + \begin{pmatrix} E\left[C_1^{(t+1)} p_{\text{best}}^{(t)}\right] + E\left[C_1^{(t+1)} g_{\text{best}}^{(t)}\right) \\ 0 \end{pmatrix}$$
- Compute spectral radius of $\begin{pmatrix} EU & -w \\ 1 & 0 \end{pmatrix}$
- Eigenvalues:
$$0 = \det\begin{pmatrix} \lambda - EU & w \\ -1 & \lambda \end{pmatrix} = \lambda^2 - \lambda EU + w \quad \Rightarrow \quad \lambda_{1,2} = \frac{EU \pm \sqrt{EU^2 - 4w}}{2}$$
- $EU = 1 + w - EC_1^{(t+1)} - EC_2^{(t+1)} = 1 + w - \frac{c_1 + c_2}{2}$



- One can show:

$$\rho(M) = \max\left\{\frac{\left|EU + \sqrt{EU^2 - 4w}\right|}{2}, \frac{\left|EU - \sqrt{EU^2 - 4w}\right|}{2}\right\} < 1 \text{ iff}$$
$$-1 < w < 1 \text{ and } 0 < \frac{c_1 + c_2}{2} < 2(w + 1)$$

- Assume $c = c_1 = c_2$

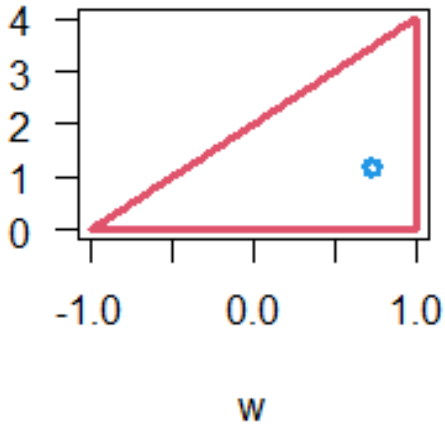
Stockholm University

Particle swarm optimisation – stability analyses

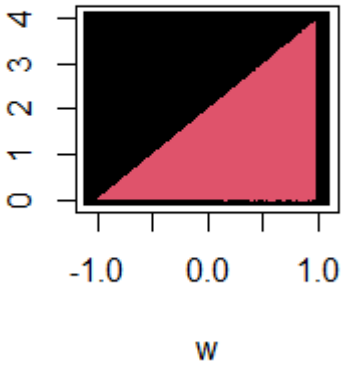
- Assume $c = c_1 = c_2$. $EU = 1 + w - c$
- One can show:

$$\rho(M) = \max\left\{\frac{\left|EU+\sqrt{EU^2-4w}\right|}{2}, \frac{\left|EU-\sqrt{EU^2-4w}\right|}{2}\right\} < 1 \text{ iff}$$

$-1 < w < 1$ and $0 < c < 2(w + 1)$



- If it would be too difficult to show the above, one can calculate the maximum eigenvalue for a grid of (w, c)-pairs and plot the cases when it is <1 (see R code on homepage)



Stockholm University

Particle swarm optimisation – stability analyses

- To do stability analyses for order-2 stability (about the limit of the variance $V(\boldsymbol{z}^{(t+1)})$), we can investigate

$$\boldsymbol{z}^{(t+1)} = (x^{(t+1)}, x^{(t)}, \left(x^{(t+1)}\right)^2, \left(x^{(t)}\right)^2, x^{(t+1)}x^{(t)})^T$$

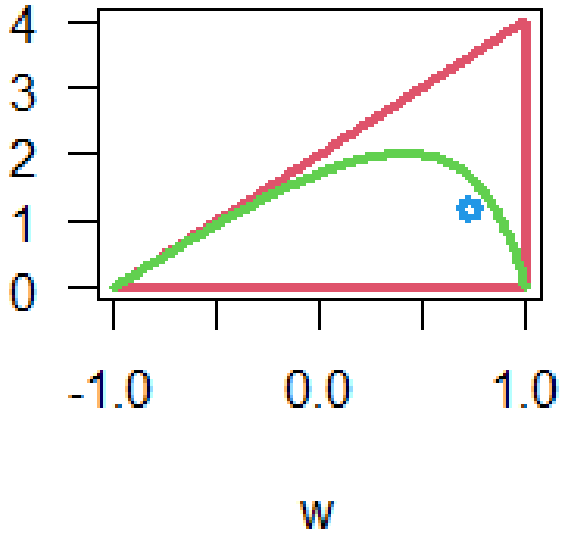
- The iterations can be written as system

$$E\boldsymbol{z}^{(t+1)} = \begin{pmatrix} EU & -w & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 2E[UP] & -2wEP & E[U^2] & w^2 & -2wEU \\ 0 & 0 & 1 & 0 & 0 \\ EP & 0 & EU & 0 & -w \end{pmatrix} E\boldsymbol{z}^{(t)} + \boldsymbol{b}$$

where $P = C_1^{(t+1)} p_{\text{best}}^{(t)} + C_2^{(t+1)} g_{\text{best}}^{(t)}$

Particle swarm optimisation – stability analyses

- $c = c_1 = c_2$
- $-1 < w < 1$ and
  $0 < c < 2(w + 1)$



- Sequence $\left(\boldsymbol{z}^{(t+1)}\right)$ is order-2 stable if:
  $-1 < w < 1$ and

$$0 < c < \frac{12(w^2 - 1)}{5w - 7}$$

- Default in **R**–package **pso** based on Clerc and Kennedy (2002):
  $w = \frac{1}{2\ln(2)} = 0.721,\ c = c_1 = c_2 = \frac{1}{2} + \ln(2) = 1.193$

Stockholm University

Particle swarm optimisation – choice of hyperparameters

- Based on stability analysis, choose w, c_1, c_2 respecting $-1 < w < 1$ and $0 < c_1 + c_2 < \frac{24(w^2-1)}{5w-7}$
- w>0 is in spirit of the algorithm's idea

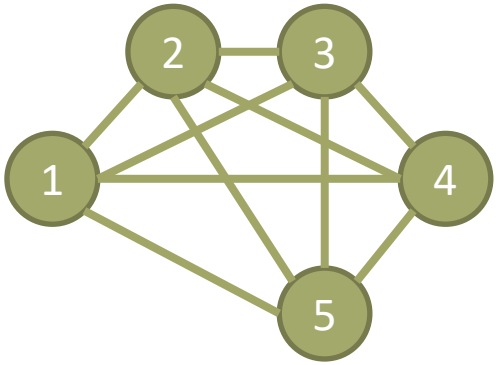
- Another hyperparameter to be chosen: swarm size
- Swarm size motivated by empirical studies based on standard optimisation problems
- SPSO 2007: $10 + \lceil 2\sqrt{p} \rceil$
- Clerc (2012) shows with 12 standard optimisation problems:
  - usually swarm sizes larger than $10 + \lceil 2\sqrt{p} \rceil$ better,
  - dependence on dimension p is weak
- SPSO 2011: choice of user; suggested: 40

Particle swarm optimisation – choice of hyperparameters using empirical studies

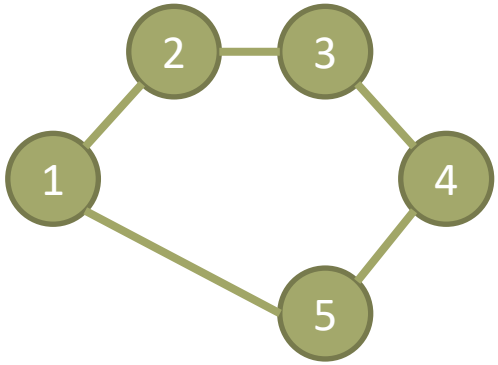
- Example: Problem 3.2 from OASI
  - one global and three other local optima
  - use different swarm sizes (e.g. 10, 20, 50, 100) and different average percentage of informants (e.g. 0.1, 0.2, 0.5, 1), run 100 times each, and report percentage identifications of global maximum
  - here: fixed function given to be optimised
- In general, one might want to compare algorithms for a set of easy and difficult optimisation problems
- For comparability, often ”standard optimisation problems” used; see e.g. Liang et al. (2013)
- Can be mathematical functions or statistical optimisation problems

Particle swarm optimisation – topologies for particles

- Particles ”inform” other particles about their results
- In the original PSO, each particle informs all others
- To ensure that not all particles are attracted prematurely by particle at a local optimum, do not inform all particles
- The structure how information flows is specified in ”topologies”
- Global top. (all inform all)



Ring top. (all inform their two ”neighbours”)



Stockholm University

Particle swarm optimisation – exploration versus exploitation

- Exploration of the search space
- Exploitation around a promising position

- The topology
  - A sparce topology (e.g. ring top.) ensures more exploration compared to a dense one (e.g. global top.)
- Parameter w
  - Larger w leads to more exploration
- Parameters c_1 and c_2
  - Smaller c_2 (and c_1) lead to more exploration
- Clerc (2016; Section 8.6.4.1): The experimental evidence for such dependencies [on w, c_1, c_2] is weak

Stochastic gradient descent with momentum

Stochastic gradient descent with momentum

- Gradient descent with (Polyak's) momentum (GDM)

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right) + \beta_t(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})$

  can be written also as

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha_t \boldsymbol{v}^{(t+1)}$

  $\boldsymbol{v}^{(t+1)} = \beta_t \boldsymbol{v}^{(t)} - \boldsymbol{g}'\left(\boldsymbol{x}^{(t)}\right)$

- Stochastic gradient descent with momentum (SGDM)

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) + \beta_t(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})$

  can be written also as

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha_t \boldsymbol{v}^{(t+1)}$

  $\boldsymbol{v}^{(t+1)} = \beta_t \boldsymbol{v}^{(t)} - \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)$

Stochastic gradient descent with momentum (SGDM)

- "The idea behind SGDM originates from Polyak's heavy-ball method for deterministic optimization. For strongly convex and smooth objectives, heavy-ball method enjoys an accelerated linear convergence rate over gradient descent.
- However, the theoretical understanding of its stochastic counterpart is far from being complete." (Liu et al., 2020).

Gradient descent method (with/without momentum) – convergence

- Convergence rate for $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$:
  - Gradient descent (GD): $\rho = \frac{\kappa-1}{\kappa+1}$
  - Gradient descent with momentum (GDM): $\rho = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)$

- $\lim_{t\to\infty} \|x^{(t+1)}-x^*\| / \|x^{(t)}-x^*\|^{\beta} = \rho$

  convergence order; here $\beta = 1$

  convergence rate

- Example $\kappa = 100$ ("ill-conditioned"):
  - $\frac{\kappa-1}{\kappa+1} = \frac{99}{101}$; $(\frac{\kappa-1}{\kappa+1})^t = 1, 0.98, \ldots, \underset{\text{t=10}}{0.82}, \ldots, \underset{\text{t=100}}{0.14}, \ldots$
  - $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{9}{11}$; $(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^t = 1, 0.82, \ldots, 0.13, \ldots, 1.9 \cdot 10^{-9}, \ldots$

Stochastic gradient descent with momentum (SGDM)

- GD: $\left\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\right\|$ decreases like $(\frac{\kappa-1}{\kappa+1})^t$, i.e. driven by κ
- GDM: $\left\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\right\|$ decreases like $(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^t$, i.e. driven by $\sqrt{\kappa}$

- SGD (for strongly convex functions): $\boldsymbol{x}^{(t)}$ fluctuates finally around $\boldsymbol{x}^*$; until it arrives in this neighbourhood, the decrease is driven by κ
- SGDM (for least squares regression): $\boldsymbol{x}^{(t)}$ fluctuates finally around $\boldsymbol{x}^*$; until it arrives in this neighbourhood, the decrease is driven by $\sqrt{\kappa\tilde{\kappa}}$ with $\tilde{\kappa} \leq \kappa$ being a *statistical condition number*

- For more details see Jain et al. (2018)

Stochastic gradient descent with momentum (SGDM)

- Stochastic gradient descent with (Polyak's) momentum

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \alpha_t \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) + \beta_t(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})$

  can be written also as

  $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha_t \boldsymbol{v}^{(t+1)}$

  $\boldsymbol{v}^{(t+1)} = \beta_t \boldsymbol{v}^{(t)} - \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right)$

- For $\beta_t = \beta$ constant, the direction of the iterations is

  $\boldsymbol{v}^{(t+1)} = \beta\left(\beta \boldsymbol{v}^{(t-1)} - \boldsymbol{g}'_{R(t-1)}\left(\boldsymbol{x}^{(t-1)}\right)\right) - \boldsymbol{g}'_{R(t)}\left(\boldsymbol{x}^{(t)}\right) = \cdots$

  $= -\sum_{\boldsymbol{k}=\mathbf{0}}^{\boldsymbol{t}} \boldsymbol{g}'_{R(t-k)}\left(\boldsymbol{x}^{(t-k)}\right)\beta^{\boldsymbol{k}}$

➢ direction of movement is moving average of preceding gradients with decreasing weights for earlier gradients

References

- Bonyadi MR, Michalewicz Z (2016). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE transactions on evolutionary computation*, 20(5):814-819.
- Cleghorn CW, Engelbrecht AP (2018). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, 12(1):1–22. Here an author version.
- Clerc M (2012). *Standard Particle Swarm Optimisation*. hal-00764996.
- Clerc M (2016). Chapter 8: Particle swarms. In: *Metaheuristics.* (Siarry P ed.).
- Clerc M, Kennedy J (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58-73.
- Jain P, Kakade SM, Kidambi R, Netrapalli P, Sidford A (2018). Accelerating stochastic gradient descent for least squares regression. *arXiv preprint* arXiv:1704.08227v2
- Liang JJ, Qu BY, Suganthan PN (2013). Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical report.
- Liu Y, Gao Y, Yin W (2020). An improved analysis of stochastic gradient descent with momentum. *arXiv preprint* arXiv:2007.07989.