DM812
METAHEURISTICS

Lecture 12

Cross Entropy Method
Continuous Optimization

Marco Chiarandini

Department of Mathematics and Computer Science
University of Southern Denmark, Odense, Denmark
<marco@imada.sdu.dk>

Outline

Outline

Cross Entropy Method

Key idea: use rare event-simulation and importance sampling to proceed towards good solutions

- generate random solution samples according to a specified mechanism
- update the parameters of the random mechanism to produce a better "sample"

CE for Optimization

Notation:

- $\mathcal{S}$ finite set of states
- f real valued performance functions on $\mathcal{S}$
- $\max_{s\in\mathcal{S}} f(s) = \gamma^* = f(s^*)$ (our problem)
- $\{p(s,\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$ family of discrete probability mass function on $s \in \mathcal{S}$
- $E_{\boldsymbol{\theta}}[f(s)] = \sum_{s\in\mathcal{S}} f(s)p(s,\boldsymbol{\theta})$

We are interested in the probability that $f(s)$ is greater than some threshold γ under the probability $p(\cdot,\boldsymbol{\theta}^*)$:

$$\ell = \Pr(f(s) \geq \gamma) = \sum_s I\{f(s) \geq \gamma\} p(s,\boldsymbol{\theta}') = E_{\boldsymbol{\theta}'}\left[I\{f(s) \geq \gamma\}\right]$$

if this probability is very small then we call $\{f(s) \geq \gamma\}$ a rare event

Estimation

$$\ell = \sum_s I\{f(s) \geq \gamma\} p(s,\boldsymbol{\theta}') = E_{\boldsymbol{\theta}'}\left[I\{f(s) \geq \gamma\}\right]$$

Monte-Carlo simulation:

- draw a random sample
- compute unbiased estimator of ℓ: $\hat{\ell} = \frac{1}{N}\sum_{i=1}^N I\{f(s_i) \geq \gamma\}$
- if probability to sample $I\{f(s_i) \geq \gamma\}$ the estimation is not accurate

Importance sampling:

- use a different probability function g on $\mathcal{S}$ to sample the solutions
- $\ell = \sum_s I\{f(s) \geq \gamma\} \frac{p(s,\boldsymbol{\theta}')}{g(s)} g(s) = E_g\left[I\{f(s) \geq \gamma\} \frac{p(s,\boldsymbol{\theta}')}{g(s)}\right]$
- compute unbiased estimator of ℓ:

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^N I\{f(s_i) \geq \gamma\} \frac{p(s,\boldsymbol{\theta}')}{g(s)}$$

- How to determine g?
  Best choice would be:

$$g^*(s) := \frac{I\{f(s) \geq \gamma\} p(x,\boldsymbol{\theta}')}{l},$$

  as substituting $\hat{\ell} = \frac{1}{N}\sum_{i=1}^N I\{f(s_i) \geq \gamma\} \frac{p(s,\boldsymbol{\theta}')}{g^*(s)} = \ell$.
  But ℓ is unknwon.

- It is convinient to choose g from $\{p(\cdot,\boldsymbol{\theta})\}$
- Choose the parameter $\boldsymbol{\theta}$ such that the difference of $g = p(\cdot,\boldsymbol{\theta})$ to g^* is minimal
- Cross entropy or Kullback Leibler distance, measure of the distance between two probability distribution functions,

$$\mathcal{D}(g^*, g) = E_{g^*}\left[\ln \frac{g^*(s)}{g(s)}\right]$$

- Generalizing to probability density functions and Lebesque integrals

$$\min \mathcal{D}(g^*, g) = \min_{\boldsymbol{\theta}} \int g^*(s) \ln g^*(s) ds - \int g^*(s) \ln g(s,\boldsymbol{\theta}) ds$$

- Minimizing the distance by means of sampling estimation leads to:

$$\widehat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}''} I\{f(s_i) \geq \gamma\} \frac{p(s,\boldsymbol{\theta}')}{p(s,\boldsymbol{\theta}'')} \ln p(s,\boldsymbol{\theta})$$

  stochastic program (convex).
  In some cases can be solved in closed form (eg, exponential, Bernoulli).

- Same result can be obtained by maximum likelihood estimation over the solutions s_i with performance $\geq \gamma$

$$L = \max_{\boldsymbol{\theta}} \prod_{i=1}^N p(s_i,\boldsymbol{\theta})$$

- Estimation via stochastic counterpart:

$$\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} I\{f(s_i) \geq \gamma)\} \frac{p(s_i, \boldsymbol{\theta}')}{p(s_i, \boldsymbol{\theta}'')} \ln p(s_i, \boldsymbol{\theta})$$

where $s_1, \ldots, s_N$ is a random sample from $p(\cdot, \boldsymbol{\theta}'')$.

- But still problems with sampling due to rare events.
  Solution: Two-phase iterative approach:
  - construct a sequence of levels $\widehat{\gamma}_1, \widehat{\gamma}_2, \ldots, \widehat{\gamma}_t$
  - construct a sequence of parameters $\widehat{\boldsymbol{\theta}}_1, \widehat{\boldsymbol{\theta}}_2, \ldots, \widehat{\boldsymbol{\theta}}_t$

  such that $\widehat{\gamma}_t$ is close to optimal
  and $\widehat{\boldsymbol{\theta}}_t$ assigns maximal probability to sample high quality solutions

Model Based Metaheuristics
Continuous Optimization
CEM

Cross Entropy Method (CEM):

Define $\widehat{\boldsymbol{\theta}}_0$. Set $t = 1$

while *termination criterion* is not satisfied **do**

generate a sample $(s_1, s_2, \ldots s_N)$ from the pdf $p(\cdot; \hat{\boldsymbol{\theta}}_{t-1})$

set $\widehat{\gamma}_t$ equal to the $(1-\rho)$-quantile with respect to f

$(\widehat{\gamma}_t = s^{(\lceil (1-\rho)N \rceil)})$

use the same sample $(s_1, s_2, \ldots, s_N)$ to solve the stochastic program

$$\widehat{\boldsymbol{\theta}}_t = \arg\max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} I_{\{f(s_i) \leq \widehat{\gamma}_t\}} \ln p(s_i; \boldsymbol{\theta})$$

Model Based Metaheuristics
Continuous Optimization
CEM

- **Termination criterion**: if for some $t \geq d$ with, *e.g.*, $d = 5$,
  $\widehat{\gamma}_t = \widehat{\gamma}_{t-1} = \ldots = \widehat{\gamma}_{t-d}$
- **Smoothed Updating**: $\widehat{\boldsymbol{\theta}}_t = \alpha \widehat{\boldsymbol{\theta}}'_t + (1-\alpha)\widehat{\boldsymbol{\theta}}_{t-1}$ with $0.4 \leq \alpha \leq 0.9$
  θ'_t from the stochastic counterpart
- **Parameters**:
  - $N = cn$, n size of the problem (number of choices available for each solution component to decide)
  - $c > 1$ $(5 \leq c \leq 10)$;
  - $\rho \approx 0.01$ for $n \geq 100$ and $\rho \approx \ln(n)/n$ for $n < 100$

Model Based Metaheuristics
Continuous Optimization
CEM

Example: TSP

- Solution representation: permutation representation
- Probabilistic model: matrix P where p_{ij} represents probability of vertex j after vertex i
- Tour construction: specific for tours
  Define $P^{(1)} = P$ and $X_1 = 1$. Let $k = 1$
  while $k < n-1$ **do**
  obtain $P^{(k+1)}$ from $P^{(k)}$ by setting the X_k-th column of $P^{(k)}$ to zero and normalizing the rows to sum up to 1.
  Generate X_{k+1} from the distribution formed by the X_k-th row of $P^{(k)}$
  set $k = k + 1$
- Update: take the fraction of times transition i to j occurred in those paths the cycles that have $f(s) \leq \gamma$

Outline

Continuous Optimization

Model Based Metaheuristics
Continuous Optimization
Numerical Analysis

- We look at unconstrained optimization of continuous, non-linear, non-convex, non-differentiable functions
- Many applications above all in statistical estimation, (eg, likelihood estimation)
- Typically few variables (curse of dimensionality)

Standard Test Functions

Model Based Metaheuristics
Continuous Optimization
Numerical Analysis

- Rosenbrock's banana function

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2$$

  Global minimum at $(x,y) = (1,1)$ where $f(x,y) = 0$

  Multidimensional extension is

$$f(x) = \sum_{i=1}^{N-1} \left[(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2\right] \quad \forall x \in \mathbb{R}^N.$$

  Global minimum at $(x_1, \ldots, x_N) = (1, \ldots, 1)$
- Rastrigin's
- Schwefel's
- Sphere

Continue at: http://www.cs.bham.ac.uk/research/projects/ecb/

Smooth Functions

Differentiable

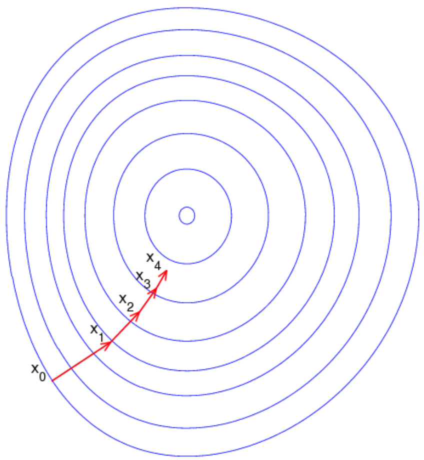
Model Based Metaheuristics
Continuous Optimization
Numerical Analysis

Gradient Descent $f(\mathbf{x})$ decreases fastest moving in the direction of the negative gradient of f Hence,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \nabla f(\mathbf{x}_n)$$

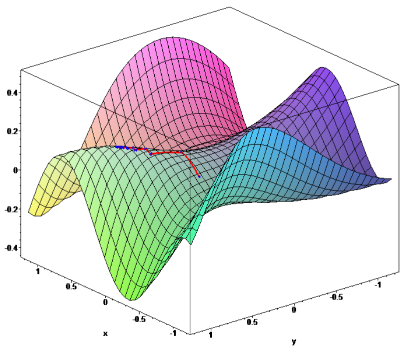
converges for appropriate x_0 and for $\gamma_n > 0$ small enough numbers.

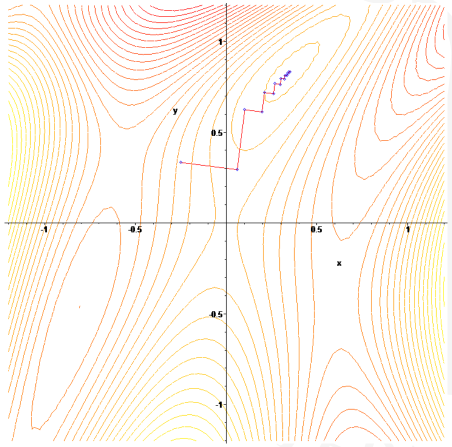
- Problem is choosing γ



Secant Method
If only one-dimension and f hard to differentiate:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n).$$





Smooth functions

Twice differentiable

Newton's method in one dimension
Taylor expansion of $f(x)$:

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2,$$

attains its extremum when Δx solves the linear equation:

$$f'(x) + f''(x)\Delta x = 0 \qquad \text{and} \, f''(x) > 0$$

Hence, if x_0 is chosen appropriately, the sequence below converges to x^*

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, \ n \geq 0$$

Newton's method generalized to several dimensions
first derivative $\longleftarrow$ gradient $\nabla f(\mathbf{x})$,
reciprocal of the second derivative $\longleftarrow$ inverse of Hessian matrix, $Hf(\mathbf{x})$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [Hf(\mathbf{x}_n)]^{-1} \nabla f(\mathbf{x}_n), \ n \geq 0.$$

- Newton's method converges much faster towards a local maximum or minimum than gradient descent.
- However, finding the inverse of the Hessian may be an expensive operation, so approximations may be used instead
  Quasi-Newton methods
  - Conjugate Gradient [Fletcher and Reeves (1964)]
  - BFGS (variable metric algorithm) [Broyden, Fletcher, Goldfarb and Shanno (1970)]