Estimation-of-Distribution Algorithms.
Continuous Domain.

Petr Pošík
Intro to EDAs
Black-box optimization
GA vs. EDA
- GA approach: select — crossover — mutate
- EDA approach: select — model — sample

EDA with binary representation
- the best possible (general, flexible) model: joint probability
  - determine the probability of each possible combination of bits
  - $2^D - 1$ parameters, exponential complexity
- less precise (less flexible), but simpler probabilistic models

Content of the lectures

Binary EDAs
- Without interactions
  - 1-dimensional marginal probabilities $p(X = x)$
  - PBIL, UMDA, cGA
- Pairwise interactions
  - conditional probabilities $p(X = x|Y = y)$
  - sequences (MIMIC), trees (COMIT), forrest (BMDA)
- Multivariate interactions
  - conditional probabilities $p(X = x|Y = y, Z = z, \ldots)$
  - Bayesian networks (BOA, EBNA, LFDA)

Continuous EDAs
- Histograms, mixtures of Gaussian distributions
- Analysis of a simple Gaussian EDA
- Remedies for premature convergence
  - Evolutionary strategies
  - AMS, Weighting, CMA-ES, classification
The difference of binary and real space

**Binary space**
- Each possible solution is placed in one of the corners of $D$-dimensional hypercube
- No values lying between them
- Finite number of elements
- Not possible to make 2 or more steps in the same direction

**Real space**
- The space in each dimension need not be bounded
- Even when bounded by a hypercube, there are infinitely many points between the bounds (theoretically; in practice we are limited by the numerical precision of given machine)
- Infinitely many (even uncountably many) candidate solutions

Local neighborhood

How do you define a local neighborhood?
- …as a set of points that do not have the distance to a reference point larger than a threshold?
  - The volume of the local neighborhood relative to the volume of the whole space exponentially drops
  - With increasing dimensionality the neighborhood becomes increasingly more local
- …as a set of points that are closest to the reference point and their unification covers part of the search space of certain (constant) size?
  - The size of the local neighborhood rises with dimensionality of the search space
  - With increasing dimensionality of the search space the neighborhood is increasingly less local

Another manifestation of the curse of dimensionality!
Taxonomy

2 basic approaches:
- discretize the representation and use EDA with discrete model
- use EDA with natively continuous model

Again, classification based on the interactions complexity they can handle:
- Without interactions
  - UMDA: model is product of univariate marginal models, only their type is different
  - Univariate histograms?
  - Univariate Gaussian distribution?
  - Univariate mixture of Gaussians?
- Pairwise and higher-order interactions:
  - Many different types of interactions!
  - Model which would describe all possible kinds of interaction is virtually impossible to find!

No Interactions Among Variables

UMDA: EDA with marginal product model $p(x) = \prod_{d=1}^{D} p(x_d)$

Lessons learned:
- If a separable function is rotated, UMDA does not work.
- If there are nonlinear interactions, UMDA does not work.
- EDAs with univariate marginal product models are not flexible enough!
- We need EDAs that can handle some kind of interactions!
Distribution Tree

Distribution Tree-Building Real-valued EA [Poš04]

- Identifies hyper-rectangular areas of the search space with significantly different densities
- Can handle certain type of interactions

Lessons learned:
- Cannot model promising areas not aligned with the coordinate axes.
- *We need models able to rotate the coordinate system!*

Global Coordinate Transformations

**Algorithm 1:** EDA with global coordinate transformation

```plaintext
begin
  Initialize the population.
  while termination criteria are not met do
    Select parents from the population.
    Transform the parents to a space where the variables are independent of each other.
    Learn a model of the transformed parents distribution.
    Sample new individuals in the transformed space.
    Transform the offspring back to the original space.
    Incorporate offspring into the population.
end
```

The individuals are
- Evaluated in the original space (where the fitness function is defined), but
- Bred in the transformed space (where the dependencies are reduced).

Linear Coordinate Transformations
UMDA with equi-height histogram models [?]:
- No transformation vs. PCA vs. ICA
- PCA and ICA are used to find a suitable rotation of the space, not to reduce the space dimensionality

Different results: the difference does not matter.

Lessons learned:
- The global information extracted by linear transformations was often not useful.
- We need non-linear transformations or local transformations!!!


Mixture of Gaussians
Gaussian mixture model (GMM):

\[ P(x) = \sum_{k=1}^{K} a_k N(x | \mu_k, \Sigma_k) \]  \hspace{1cm} (1)

Normalization and the requirement of positivity:

\[ \sum_{k=1}^{K} a_k = 1 \]
\[ 0 \leq a_k \leq 1 \]

Model learned by EM algorithm.

Lessons learned:
- GMM is able to model locally linear dependencies.
- We need to specify the number of components beforehand!
- If the optimum is not covered by at least one of the Gaussian peaks, the EA will miss it!
Non-linear global transformation

Kernel PCA as the transformation technique in EDA [?] 

Works too well:
- It reproduces the pattern with high fidelity
- If the population is not centered around the optimum, the EA will miss it

Lessons learned:
- Continuous EDA must be able to effectively move the whole population!!!
- Is the MLE principle actually suitable for model building in EAs???

P. Pošik © 2014

Simple Gaussian EDA

Consider a simple EDA with the following settings:

Algorithm 2: Gaussian EDA

\begin{algorithm}
\begin{algorithmic}[1]
\State \{\mu_1, \Sigma_1\} \leftarrow \text{InitializeModel}()
\State \tau \leftarrow 1
\While{\text{not TerminationCondition}()}
\State X \leftarrow \text{SampleGaussian}(\mu^g, k \cdot \Sigma^g)
\State f \leftarrow \text{Evaluate}(X)
\State X_{sel} \leftarrow \text{Select}(X, f, \tau)
\State \{\mu^{g+1}, \Sigma^{g+1}\} \leftarrow \text{LearnGaussian}(X_{sel})
\State \tau \leftarrow \tau + 1
\EndWhile
\end{algorithmic}
\end{algorithm}

Gaussian distribution:

\[ \mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^\frac{1}{2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right) \]

Maximum likelihood (ML) estimates of parameters

\[ \mu_{\text{ML}} = \frac{1}{N} \sum_{i=1}^{N} x_i, \text{ where } x_i \in X_{\text{sel}} \]

\[ \Sigma_{\text{ML}} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_{\text{ML}})(x_i - \mu_{\text{ML}})^T \]

P. Pošik © 2014
Premature convergence

Using Gaussian distribution and ML estimation seems as a good idea…

…but it is actually very bad optimizer!!!

Two situations:

Population centered around optimum (population in the valley):

Algorithm works:
- the optimum is located
- the algorithm focuses the population on the optimum

Population far away from optimum (population on the slope):

Algorithm fails:
- the optimum is far away
- the algorithm is not able to shift the population towards optimum
What happens on the slope?
The change of population statistics in 1 generation:

Expected value:

\[ \mu_{t+1} = E(X|X > x_{\text{min}}) = \mu^t + \sigma^t \cdot d(\tau), \]

Variance:

\[ (\sigma_{t+1}^2) = \text{Var}(X|X > x_{\text{min}}) = (\sigma^t)^2 \cdot c(\tau), \]

where

\[ d(\tau) = \frac{\phi(\Phi^{-1}(\tau))}{\tau}, \]

where

\[ c(\tau) = 1 + \frac{\Phi^{-1}(1 - \tau) \cdot \phi(\Phi^{-1}(\tau))}{\tau} - d(\tau)^2. \]
What happens on the slope (cont.)

Population statistics in generation $t$:

$$
\mu^t = \mu^0 + \sigma^0 \cdot d(\tau) \cdot \sum_{i=1}^{l} \sqrt{c(\tau)^i}
$$

$$
\sigma^t = \sigma^0 \cdot \sqrt{c(\tau)}^l
$$

Convergence of population statistics:

$$
\lim_{t \to \infty} \mu^t = \mu^0 + \sigma^0 \cdot d(\tau) \cdot \frac{1}{1-\sqrt{c(\tau)}}
$$

$$
\lim_{t \to \infty} \sigma^t = 0
$$

The **distance** the population can “travel” in this algorithm is bounded!

Premature convergence!

Lessons learned:

- Maximum likelihood estimates are suitable in situations when the model fits the fitness function well (at least in local neighborhood).
- Gaussian distribution may be suitable in the neighborhood of optimum.
- Gaussian distribution is not suitable on the slope of fitness function!
- We need something different from MLE to traverse the slopes!!!

---

Variance Enlargement in a Simple EDA

What happens if we enlarged the MLE estimate of variance with a constant multiplier $k$? [?]  
- What is the minimal value $k_{\text{min}}$ ensuring that the model will not converge on the slope?  
- What is the maximal value $k_{\text{max}}$ ensuring that the model will not diverge in the valley?  
- Is there a single value $k$ of the multiplier for MLE variance estimate that would ensure a reasonable behavior in both situations?  
- Does it depend on the type of the single-peak distribution being used?

- For Gaussian and “isotropic Gaussian”, allowable $k$ is hard or impossible to find.  
- For isotropic Cauchy, allowable $k$ seems to always exist…  
- …but this does not guarantee a reasonable behavior.

---

Summary of Continuous EDAs So Far

Initially, high expectations:
- Started with structurally simple models for complex objective functions.
  - They did not work, partially because of the discrepancy between the complexities of the model and the function.
- Used increasingly complex and flexible models.
  - Some improvements were gained, but even the most complex models did not fulfill the expectations.
- Realized that a fundamental mistake was present all the time:
  - MLE principle builds models which try to reconstruct the points they were build upon.
  - This allows to focus on already covered areas, but not to shift the population to unexplored places.

Current research directions:
- Aimed at understanding and developing principles critical for successful continuous EDAs.
- Studying behavior on simple functions first.
- Using simple, single-peak models so that the resulting algorithm behave (more or less) as local search procedures.

State of the Art

Current Trend: Population-based Adaptive Local Search

There's something about the population:
- data set forming a basis for offspring creation
- allows for searching the space in several places at once
  (replaced by restarted local search with adaptive neighborhood)

Hypothesis:
- The data set (population) is very useful when creating (sometimes implicit) global model of the fitness landscape or a local model of the neighborhood.
- It is often better to have a robust adaptive local search procedure and restart it, than to deal with a complex global search algorithm.
Preventing the Premature Convergence

- self-adaptation of the variance \([\gamma]\) (let the variance be part of the chromosome)
- adaptive variance scaling when population is on the slope, ML estimate of variance when population is in the valley
- anticipate the shift of the mean and move part of the offspring in the anticipated direction
- use weighted estimates of distribution parameters
- do not estimate the distribution of selected points, but rather a distribution of selected mutation steps
- use a different principle to estimate the parameters of the Gaussian


Adaptive Variance Scaling

AVS \([\gamma]\):

- Enlarge the ML estimate of \(\Sigma\) by an adaptive coefficient \(c_{\text{AVS}}\)
- If an improvement was not found in the current generation, we explore too much, thus decrease \(c_{\text{AVS}}: c_{\text{AVS}} \leftarrow \eta_{\text{DEC}} c_{\text{AVS}}\), \(\eta_{\text{DEC}} \in (0, 1)\).
- If an improvement was found in the current generation, we may get better results with increased \(c_{\text{AVS}}: c_{\text{AVS}} \leftarrow \eta_{\text{INC}} c_{\text{AVS}}\), \(\eta_{\text{INC}} > 1\).
- \(c_{\text{AVS}}\) is bounded: \(c_{\text{AVS,MIN}} \leq c_{\text{AVS}} \leq c_{\text{AVS-MAX}}\)
- Stimulate exploration: if \(c_{\text{AVS}} < c_{\text{AVS-MIN}}\), reset it to \(c_{\text{AVS-MAX}}\).

AVS Triggers

With AVS, all improvements increase $c_{AVS}$:
- This is not always needed, especially in the valleys.
- Trigger AVS when on slope; in the valley, use ordinary MLE.

Correlation trigger for AVS (CT-AVS) [?]:
- Compute the ranked correlation coefficient of p.d.f. values and function values, $p(x_i)$ and $f(x_i)$.
- If the distribution is placed around optimum, function values increase with decreasing p.d.f., correlation will be large. Use ordinary MLE.
- If the distribution is on a slope, correlation will be close to zero. Use AVS.

Standard-deviation ratio trigger for AVS (SDR-AVS) [?]:
- Compute $x_{IMP}$ as the average of all improving individuals in the current population
- If $p(x_{IMP})$ is "low" (the improvements are found far away from the distribution center), we are probably on a slope. Use AVS.
- If $p(x_{IMP})$ is "high" (the improvements are found near the distribution center), we are probably in a valley. Use ordinary MLE.


Anticipated Mean Shift

Anticipated mean shift (AMS) [?]:
- AMS is defined as: $\hat{\mu}_{shift} = \hat{\mu}(t) - \hat{\mu}(t - 1)$
- AMS is an estimate of the direction of improvement
- 100% of offspring are moved by certain fraction of AMS: $x = x + \delta \hat{\mu}_{shift}$
- When centered around optimum, $\hat{\mu}_{shift} = 0$ and the original approach is unchanged.
- Selection must choose parent from both the old and the shifted regions to adjust $\Sigma$ suitably.

### Weighted ML Estimates

Account for the values of p.d.f. of the selected parents $X_{\text{sel}}$ [7]:

- assign weights inversely proportional to the values of p.d.f.

![Weighted ML Estimates](image)

**Weighted (ML) estimates of parameters**

$$
\mu_W = \frac{1}{V_1} \sum_{i=1}^{N} w_i x_i, \text{ where } x_i \in X_{\text{sel}}
$$

$$
\Sigma_W = \frac{V_1}{V_1 - V_2} \sum_{i=1}^{N} w_i (x_i - \mu_{ML})(x_i - \mu_{ML})^T
$$

where

$$
w_i = \frac{1}{p(x_i)}
$$

$$
V_1 = \sum w_i
$$

$$
V_2 = \sum w_i^2
$$

---

**CMA-ES**

Evolutionary strategy with cov. matrix adaptation [7]

- $(\mu / \mu, \lambda)$-ES (recombinative, mean-centric)
- model is adapted, not built from scratch each generation
- accumulates the successful steps over many generations

Compare:

- Simple Gaussian EDA estimates the distribution of selected individuals (left fig.)
- CMA-ES estimates the distribution of successful mutation steps (right fig.)

![CMA-ES](image)
Optimization via Classification

Build a quadratic classifier separating the selected and the discarded individuals [7]

- Classifier built by modified perceptron algorithm or by semidefinite programming
- Works well for pure quadratic functions
- If the selected and discarded individuals are not separable by an ellipsoid, the training procedure fails to create a good model
- Work in progress; not solved yet

Remarks on SotA

- Many techniques to fight premature convergence
- Although based on different principles, some of them converge to similar algorithms (weighted MLE, CMA-ES, NES)
- Only a few sound principles; the most of them are heuristic approaches
### Real-valued EDAs

- much less developed than EDAs for binary representation
- the difficulties are caused mainly by
  - much more severe effects of the curse of dimensionality
  - many different types of interactions among variables
- Gaussian distribution used most often, but pure maximum-likelihood estimates are BAD! Some other remedies are needed.
- Despite of that, EDA (and EAs generally) are able to gain better results then conventional optimization techniques (line search, Nelder-Mead search, …)

*P. Pošík © 2014*