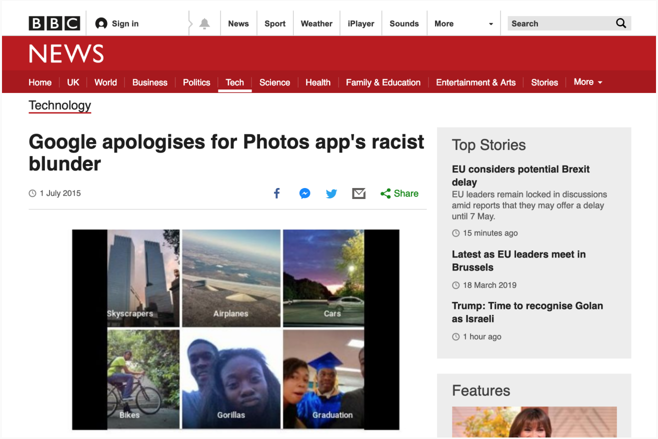
Evolutionary dataset optimisation: learning algorithm quality through evolution

Henry Wilde, Dr. Jonathan Gillard, Dr. Vincent Knight

Premise and motivation


BBC | Sign in | News | Sport | Weather | iPlayer | Sounds | More | Search

NEWS

Home | UK | World | Business | Politics | Tech | Science | Health | Family & Education | Entertainment & Arts | Stories | More

Technology

Google apologises for Photos app's racist blunder

1 July 2015

Share

Top Stories

EU considers potential Brexit delay
EU leaders remain locked in discussions amid reports that they may offer a delay until 7 May.
15 minutes ago

Latest as EU leaders meet in Brussels
18 March 2019

Trump: Time to recognise Golan as Israeli
1 hour ago

Features

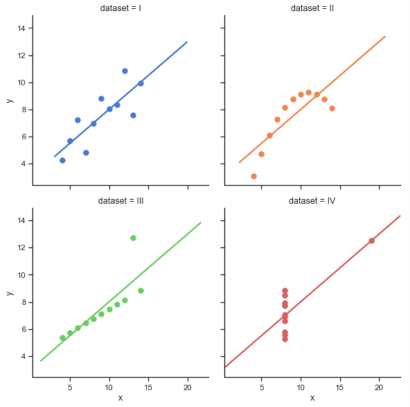
via: BBC News (`https://www.bbc.co.uk/news/technology-33347866`)

- Hyndman reference (reliability)
- A. Torralba and A. A. Efros. *Unbiased Look at Dataset Bias*. Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. 2011. DOI: 10.1109/CVPR.2011.5995347 (frailty)

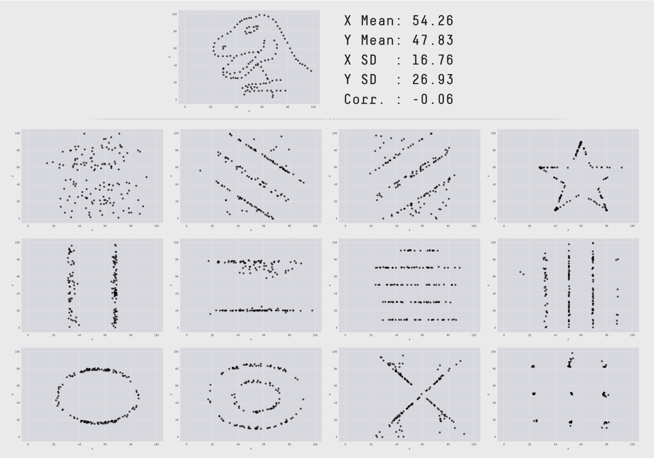
Paradigm flip diagram

Premise and motivation

Generating artificial data



via: `https://seaborn.pydata.org/examples/anscombes_quartet.html`

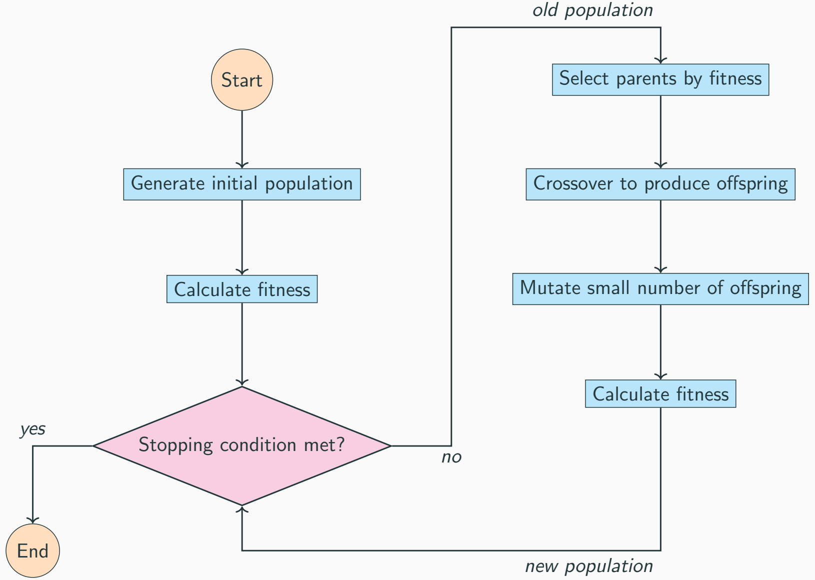


via: https://www.autodeskresearch.com/publications/samestats

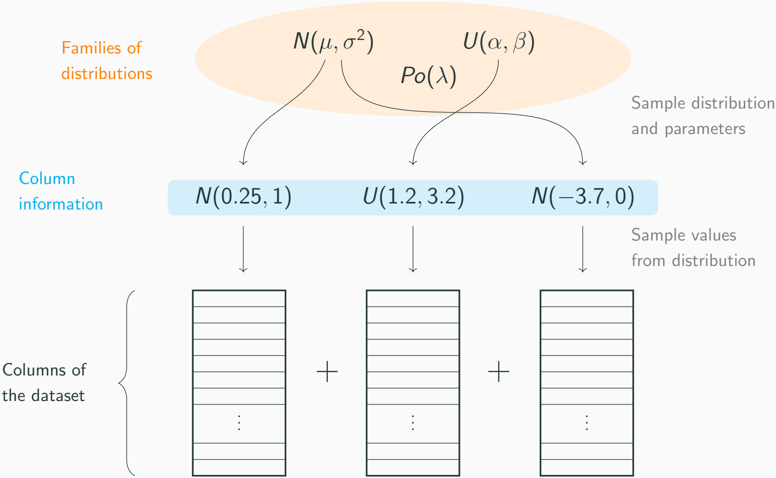
Dataset similarity diagram

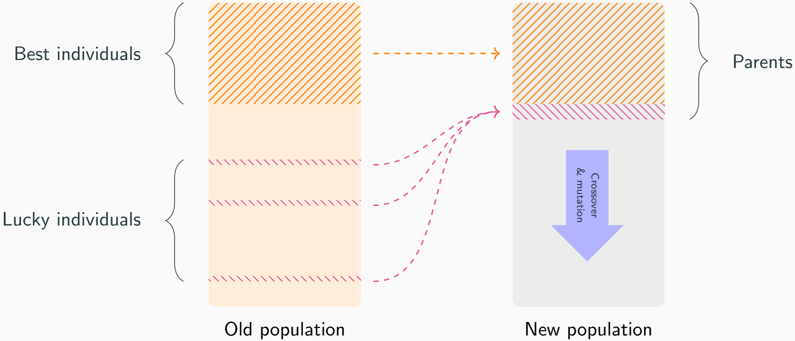
Given some algorithm, how can one find data for which it performs well?

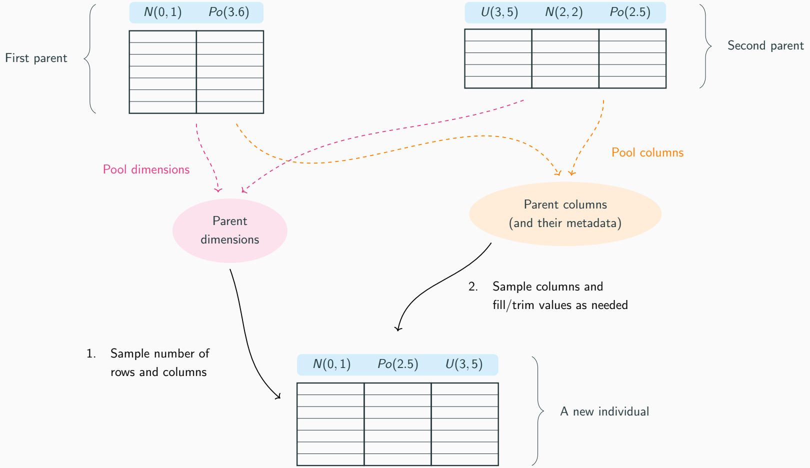
What is an evolutionary algorithm?

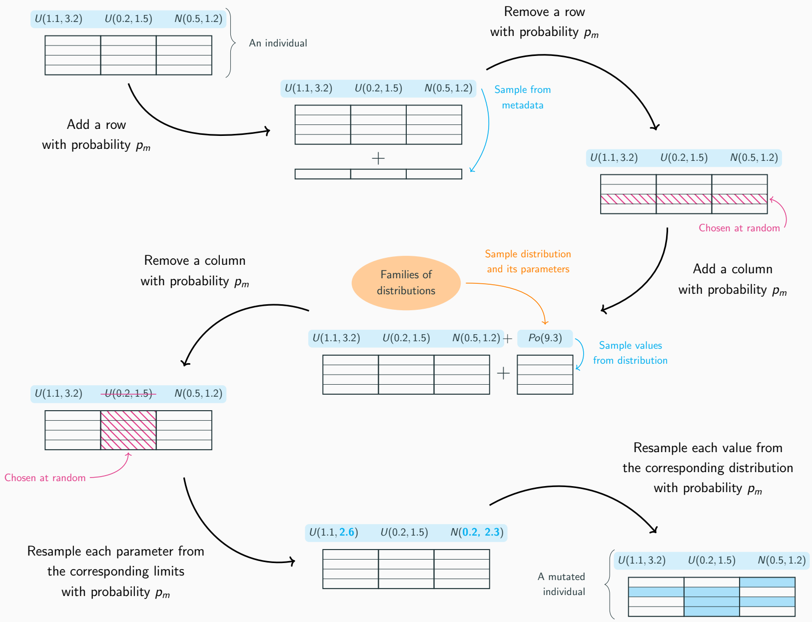

old population
Start
Select parents by fitness
Generate initial population
Crossover to produce offspring
Calculate fitness
Mutate small number of offspring
Calculate fitness
yes
Stopping condition met?
no
End
new population

GA example


Families of distributions
$N(\mu, \sigma^2)$
$U(\alpha, \beta)$
$Po(\lambda)$
Sample distribution and parameters
Column information
$N(0.25, 1)$
$U(1.2, 3.2)$
$N(-3.7, 0)$
Sample values from distribution
Columns of the dataset
+
+


Best individuals
Lucky individuals
Parents
Crossover
& mutation
Old population
New population


First parent
$N(0,1)$ $Po(3.6)$
$U(3,5)$ $N(2,2)$ $Po(2.5)$
Second parent
Pool dimensions
Pool columns
Parent dimensions
Parent columns (and their metadata)
2. Sample columns and fill/trim values as needed
1. Sample number of rows and columns
$N(0,1)$ $Po(2.5)$ $U(3,5)$
A new individual



U(1.1, 3.2) U(0.2, 1.5) N(0.5, 1.2)
An individual
Add a row
with probability p_m
Sample from
metadata
Remove a row
with probability p_m
Chosen at random
Add a column
with probability p_m
Sample distribution
and its parameters
Families of
distributions
Po(9.3)
Sample values
from distribution
Remove a column
with probability p_m
U(1.1, 3.2) ~~U(0.2, 1.5)~~ N(0.5, 1.2)
Chosen at random
Resample each parameter from
the corresponding limits
with probability p_m
U(1.1, 2.6) U(0.2, 1.5) N(0.2, 2.3)
Resample each value from
the corresponding distribution
with probability p_m
A mutated
individual

- A fitness function, f, which acts on a single dataset
- A population size, $N \in \mathbb{N}$
- Limits on the number of rows a dataset can have:

$$R \in \left\{(r_{\text{min}}, r_{\text{max}}) \in \mathbb{N}^2 \mid r_{\text{min}} \leq r_{\text{max}}\right\}$$

- Limits on the number of columns a dataset can have:

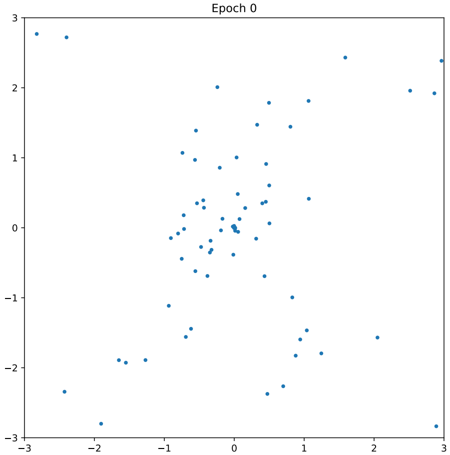
$$C := \left(C_1, \ldots, C_{|\mathcal{P}|}\right) \text{ where } C_j \in \left\{(c_{\text{min}}, c_{\text{max}}) \in (\mathbb{N} \cup \{\infty\})^2 \mid c_{\text{min}} \leq c_{\text{max}}\right\}$$

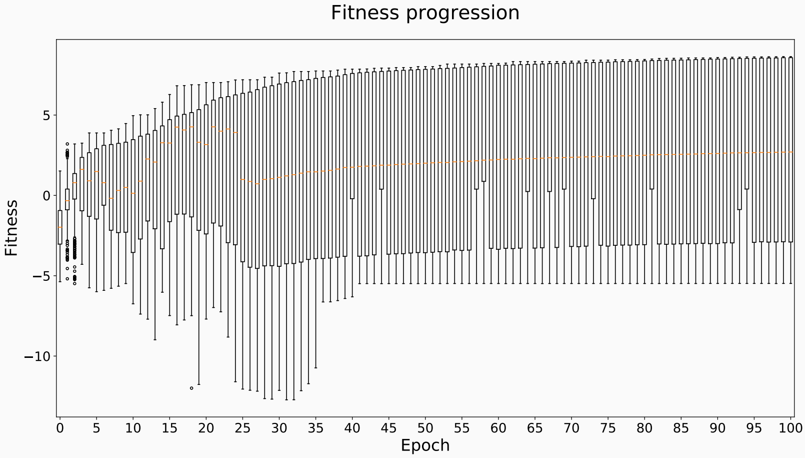
  for each $j = 1, \ldots, |\mathcal{P}|$
- A set of probability distribution families, $\mathcal{P}$. Each family in this set has some parameter limits which form a part of the overall search space
- A probability vector to sample distributions from $\mathcal{P}$, $w = \left(w_1, \ldots, w_{|\mathcal{P}|}\right)$
- A maximum number of iterations, $M \in \mathbb{N}$
- Two selection parameters: one to indicate the proportion of the fittest individuals to carry forward, $b \in [0, 1]$, and the other to allow for a small proportion of "lucky" individuals in the next generation, $l \in [0, 1]$
- A mutation probability, $p_m \in [0, 1]$
- A shrink factor, $s \in [0, 1]$. The relative size of a component of the search space to be retained after adjustment

Some example use cases

Given a dataset X, maximise f where:

$$f(X) := \max_{a,b\ \in\{(0,1),(1,0)\}} \left\{ Var\left(X^{(a)}\right) - \max_i \left| X_i^{(b)} - 1 \right| \right\}$$


Epoch 0


Fitness progression
Fitness
Epoch
5
0
−5
−10
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100

Input: A dataset (column) X, some sampling fraction p

Output: An estimate for the mean of X

$Y \leftarrow$ a random sample of $\lfloor p|X| \rfloor$ entries from X;

evaluate the mean of Y:

$$\bar{Y} = \frac{1}{|Y|} \sum_{i=1}^{|Y|} Y_i$$

$$\iff f(X) = \bar{Y}$$

Anscombe's quartet

```
$ pip install edo
```

Any questions?

Henry Wilde

Twitter: @daffidwilde

Email: henrydavidwilde@gmail.com

GitHub repository: `https://github.com/daffidwilde/edo`

Documentation: `https://edo.readthedocs.io`