THE UNIVERSITY OF MELBOURNE

Stress-testing Algorithms via Instance Space Analysis

Presented by

Professor Kate Smith-Miles

University of Melbourne

smith-miles@unimelb.edu.au

Lecture 1 (Part 1)

- Introduction to Instance Space Analysis
- Introduction to MATILDA
- Case Study: Optimisation (University Timetabling)

Lecture 2

- Case Study: Computer Vision (Facial Age Estimation)

Lecture 3

- Evolving new instances to fill an instance space (Black-box optimisation and new artwork!)

THE UNIVERSITY OF MELBOURNE

Part 2

How to perform an Instance Space Analysis

Mario Andrés Muñoz : munoz.m@unimelb.edu.au

https://matilda.unimelb.edu.au/matilda/matildadata/tutorials/matilda-technical-details.mp4

- Using the MATLAB code/live script
- Using MATILDA's web Interface
- Case Study: Machine Learning (Classification)

THE UNIVERSITY OF MELBOURNE

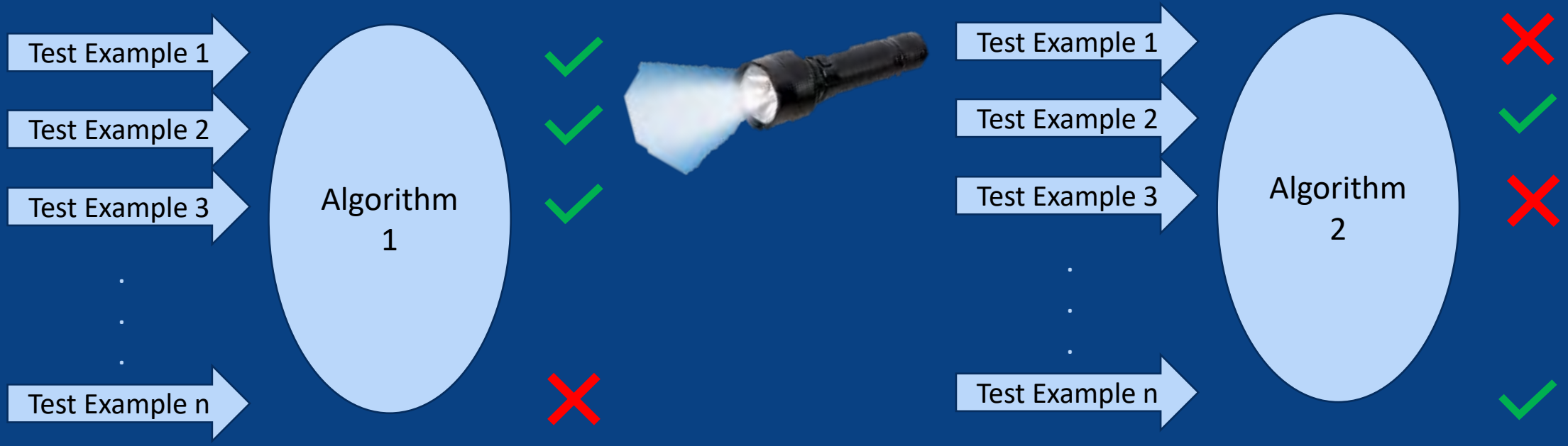
Lecture 1

Introduction to Instance Space Analysis and MATILDA


WARNING!
THIS ALGORITHM SHOULD ONLY BE USED FOR THE SCENARIOS MOST SIMILAR TO THOSE DESCRIBED IN THE TERMS AND CONDITIONS WHICH ESTABLISH ITS GUARANTEED RELIABILITY.

Establishing the T&Cs … a mathematical challenge

- Choice of test examples is critical for rigorous and trustworthy “stress-testing”



- Trustworthy test examples must be unbiased, diverse, challenging, discriminating and real-world-like … inconclusive without such guarantees

Instance Space Analysis: Motivation

- Standard Research Practice
  - Reporting algorithm performance averaged across a set of chosen test instances
  - If an algorithm's performance is demonstrably better (on average) than competing algorithms, then the conclusion is typically drawn that the algorithm is successful.
- What's wrong with that?
  - Concern #1: Which test instances were chosen, and why?
    - Hand-selected? Weaknesses rarely reported.
    - Common benchmarks are important for fair comparisons, but are they adequate?
    - Where did they come from? Are they fit for purpose now? Unbiased? Representative?

Test instances should ideally be: (i) demonstrably diverse; (ii) lacking bias; (iii) sufficiently challenging; (iv) discriminating of algorithms' performances; and (v) span a range of real-world contexts for likely deployment.

Motivation

- Concern #2: Averages are not nuanced enough to report strengths and weaknesses
  - An algorithm that is best on average undoubtedly has weaknesses
  - No Free Lunch Theorems
  - How do the characteristics of different types of instances affect algorithm behaviour?
  - Under what conditions should we expect one algorithm to be better than another?
  - Which instances possess properties that give an algorithm competitive advantage or cause it to struggle?

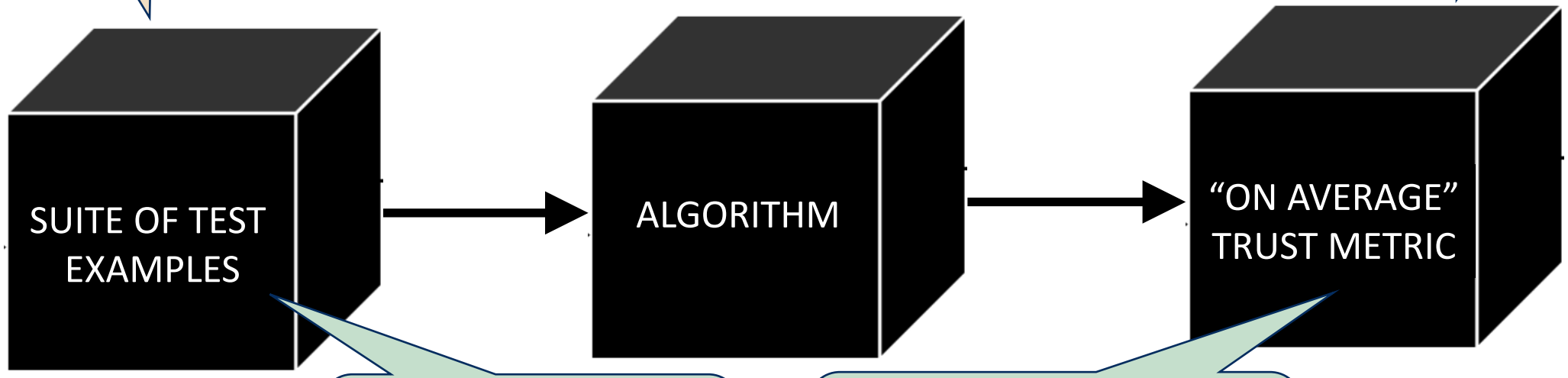
Algorithm performance should ideally be reported based on instance properties, rather than on average across all instance types.

Standard algorithm testing … in the darkness of black boxes

PROBLEM 1:
How can we scrutinise test examples to ensure they are sufficient and trustworthy?

J.N. Hooker, "Testing Heuristics: We Have It All Wrong", *Journal of Heuristics*, vol. 1, pp. 33-42, 1995.

PROBLEM 2:
Averages hide all sorts of sins … good on average doesn't mean good always!



MATILDA "cracks-open" suite to visualise bias, diversity, real-world like within likely bounds

MATILDA reveals strengths, weaknesses and suitability of algorithms on all possible examples within likely bounds

Long-standing criticism of standard practice

- J. Hooker (1994) , “Needed: An empirical science of algorithms”, Operations Research, vol. 42, no. 2, 201-212.
- J. Hooker (1995), “Testing heuristics: We have it all wrong”, Journal of Heuristics vol. 1, no. 1, pp. 33-42.
- C. C. McGeoch (2002), “Experimental analysis of algorithms”, in: Handbook of Global Optimization, Springer, pp. 489-513.
- N. G. Hall, M. E. Posner (2010), “The generation of experimental data for computational testing in optimization”, in: Experimental Methods for the Analysis of Optimization Algorithms, Springer, pp. 73-101.

It is no-one’s fault that standard practice of reporting *average* performance across *unscrutinised* test suites continues … the methodologies and tools to overcome these concerns have not been available

→ MOTIVATION FOR INSTANCE SPACE ANALYSIS & MATILDA

Instance Space Analysis: Goals

- To understand and visualize strengths and weaknesses of algorithms
  - → which algorithm should be used when and why
- To facilitate objective assessment of algorithmic power across an instance space
  - → improve research practice
  - → establish algorithmic trust
  - → avoid deployment disasters
- To guide the generation of comprehensive test instances with controllable characteristics
  - → expand existing benchmarks to be fit for purpose
  - → drive further insights and algorithm advances

matilda.unimelb.edu.au/matilda/

THE UNIVERSITY OF MELBOURNE

The University of Melbourne

SEARCH LOGIN MENU

MATILDA

Melbourne Algorithm Test Instance Library with Data Analytics

ABOUT MATILDA | OUR TEAM | OUR METHODOLOGY | GETTING STARTED

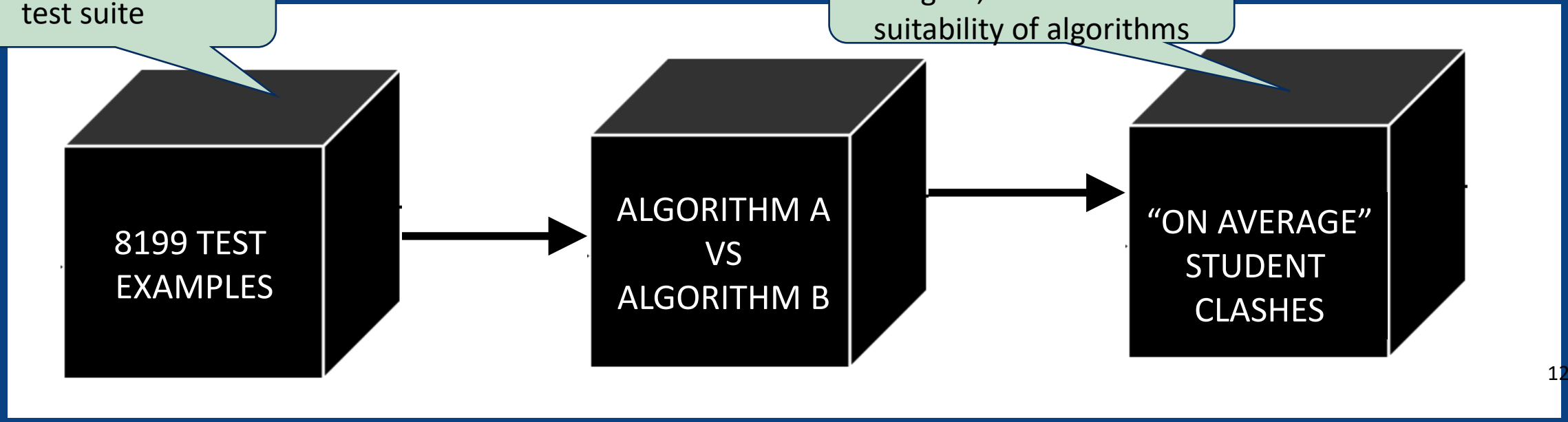
Case Study: University Timetabling

Algorithm A wins	Algorithm B wins	Algorithms tied	TOTAL
2096	2409	3694	8199
25.56%	**29.38%**	**45.05%**	**100%**

Which algorithm do you trust more?

MATILDA "cracks-open" test suite

MATILDA reveals strengths, weaknesses and suitability of algorithms



Illustrative Case Study

- University Course Timetabling
  - Top two algorithms: Tabu Search and Simulated Annealing from International Timetabling Competition (2007)
  - 21 real-world instances from University of Udine (Italy)
  - 4492 randomly generated instances from Burke et al. (2010)
  - 3686 “real-world-like” instances from Lopes and Smith-Miles (2013)

INSTANCES	SACP best	TSCS best	tied	total
Random	475	957	3060	4492
Real-world-like	1613	1442	631	3686
Udine	8	10	3	21
total	2096	2409	3694	8199
%	25.56%	29.38%	45.05%	

Which algorithm is better?

Spoiler alert: They both have weaknesses as Instance Space Analysis will reveal!

Burke, E. K., Marecek, J., Parkes, A. J. and Rudova, H., “A supernodal formulation of vertex colouring with applications in course timetabling”, Annals of Operations Research, vol. 179, no. 1, pp. 105-130, 2010.
Lopes, L. and Smith-Miles, K., “Generating Applicable Synthetic Instances for Branch Problems”, Operations Research, vol. 61, no. 3, pp. 563-577, 2013.

Open Questions

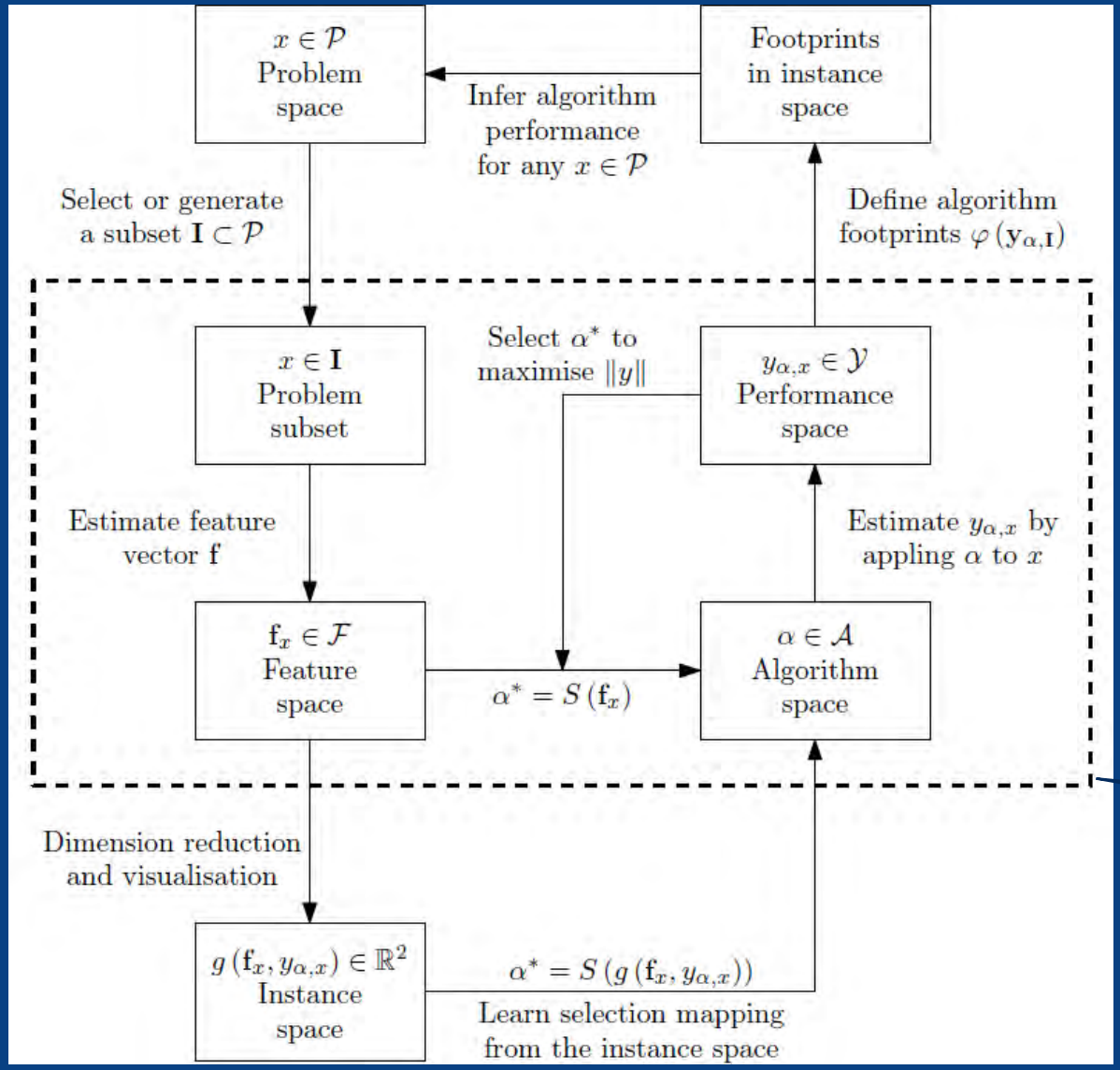
- How do instance features help us understand the strengths and weaknesses of algorithms?
- How can we infer and visualise algorithm performance across a huge instance space?
- How can we measure objectively the relative performance of algorithms?
- How easy or hard are the benchmark instances in the literature? How diverse are existing instances?
- How can we generate new test instances to gain insights into algorithmic power?

INSTANCES	SACP best	TSCS best	tied	total
Random	475	957	3060	4492
Real-world-like	1613	1442	631	3686
Udine	8	10	3	21
total	2096	2409	3694	8199
%	25.56%	29.38%	45.05%	

Instance Space Analysis: Challenges

- Visualise the space of all possible test instances of a problem
  - Identify the features of instances that affect algorithm performance
  - Mathematically define the boundaries of the feature space
  - Project to 2D to visualize existing test instances within the theoretical boundary
- Scrutinise the adequacy of existing benchmarks in the instance space
  - Identify regions to evolve new instances to enhance benchmarks
- Gain insights into strengths and weaknesses of algorithms across the instance space
  - Quantitatively: measure algorithm “footprints” as areas with good performance
  - Qualitatively: explore how instance features affect algorithm footprints
- Automate algorithm selection based on instance features
- Design new algorithms based on insights to ensure footprints cover the instance space

Instance Space Analysis: Framework



{I, F, Y, A} → meta-data

J. R. Rice (1976). "The algorithm selection problem", In Advances in Computers. Vol. 15. Elsevier, 65–118.

Meta-data requirements

- Features
  - What makes the problem hard?
  - What features capture instance difficulty?
- Instances
  - Which instances show sufficient feature diversity
  - Which instances represent real-world problems?
  - Which instances elicit differences in algorithm performance?
- Algorithms
  - Which algorithms show sufficient diversity of performance that we can learn something about the effectiveness of their underlying mechanism?
- Performance metric
  - What performance metric(s) is most relevant? How do we define good performance?

	Timetabling Meta-data (I, F, Y, A)
I	8199 instances from 3 sources
F	32 features based on underlying graph colouring problems (teachers and students), and properties such as slack, number of one room events, etc.
Y	minimise total student clashes Good = Best (minimal clashes)
A	2 algorithms (SACP and TSCS)

Instance Space Analysis: Methodology

Pre-ISA: Collect meta-data		Instances, features, algorithm performance metrics
Step 1:	Create instance space	Feature selection and dimensionality reduction
Step 2:	Visualise benchmarks	Diversity within theoretical boundary
Step 3:	Visualise algorithm footprints	Strengths, weaknesses, uniqueness
Step 4:	Explain strengths/weaknesses	Feature distribution over footprints
Step 5:	Automate algorithm selection	Machine learning predictions
Post-ISA: Insights into new features		Devise new features to improve separation
Post-ISA: Insights into new instances		Generate new instances to fill gaps
Post-ISA: Insights into new algorithms		Devise new algorithms to improve coverage
REPEAT		An iterative process until convergence of instance space

Step 1: Create the Instance Space

- Which features should be selected as a minimal set that explains algorithm performance?
  - Correlation analysis: predictive
  - Clustering: avoid redundancies

  → Optimal subset to predict good/bad
- Which dimension reduction method will create a 2D visualisation that separates easy (good) and hard (not-good) instances and explains why?

$$\min_{\{A,B,C\}} \left|\left|F - \hat{F}\right|\right|^2 + \left|\left|Y - \hat{Y}\right|\right|^2$$

$$\text{subject to } Z = AF, \hat{F} = BZ, \hat{Y} = CZ$$

Timetabling Example

32 initial features
5 features selected to best separate good/bad

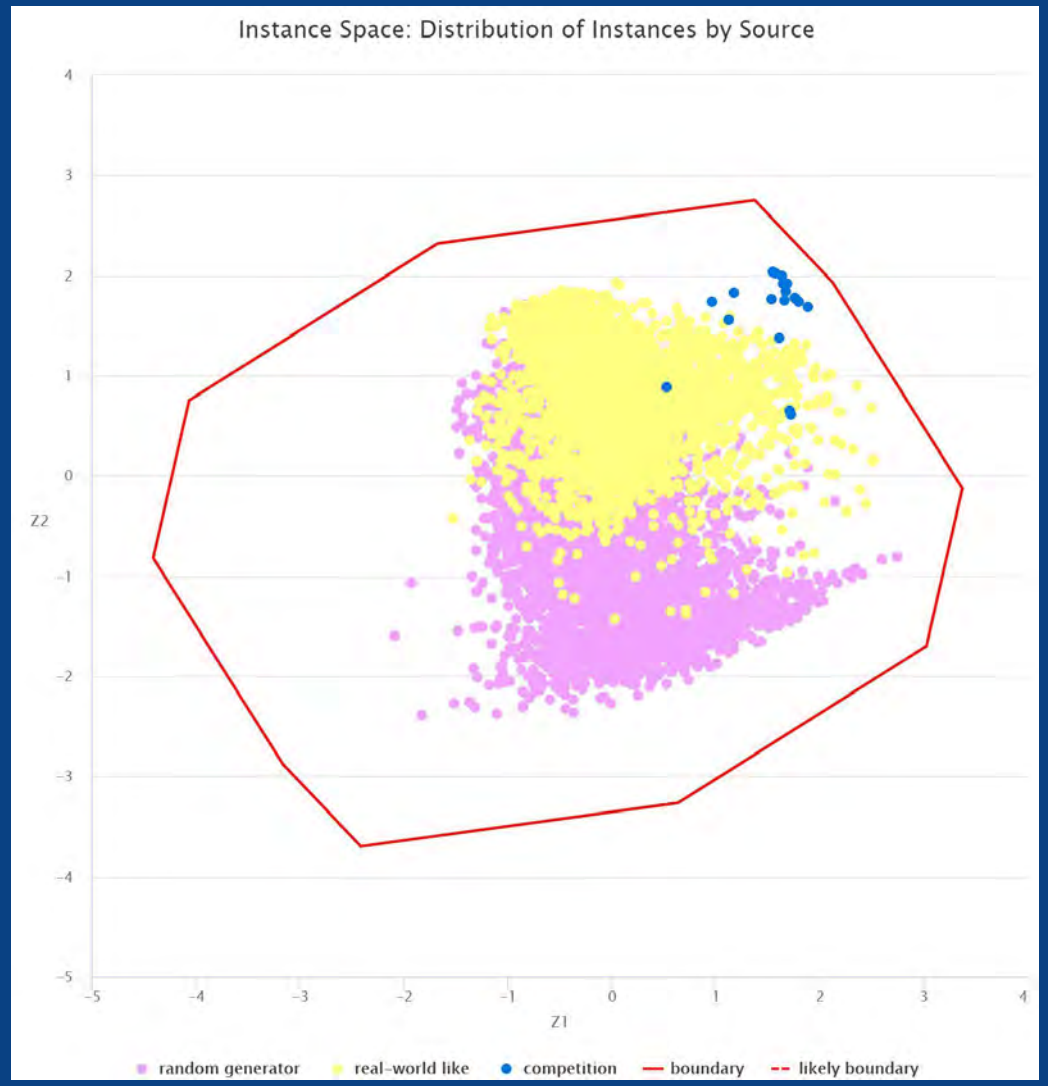
Good defined as best (minimal clashes)

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} -0.3662 & -0.0519 \\ -0.2814 & 0.4637 \\ 0.1931 & -0.2114 \\ -0.0992 & -0.4511 \\ -0.2499 & -0.1637 \end{bmatrix}^T \begin{bmatrix} sdEventSize \\ slack \\ eventDegreeTeacherConnectivity \\ oneRoomEvents \\ eventDegreeTeacherMeanWeighted \end{bmatrix}$$

M. A. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles, "Instance spaces for machine learning classification," *Mach. Learn.*, vol. 107, no. 1, pp. 109-147, 2018.

Step 2: Visualise Benchmark Instances

- Where do different instance classes lie?
- Do they fill within the theoretical boundary?
- Are they
- (i) demonstrably diverse;
- (ii) lacking bias;
- (iii) sufficiently challenging;
- (iv) discriminating of algorithms' performances;
- (v) spanning a range of real-world contexts for likely deployment.

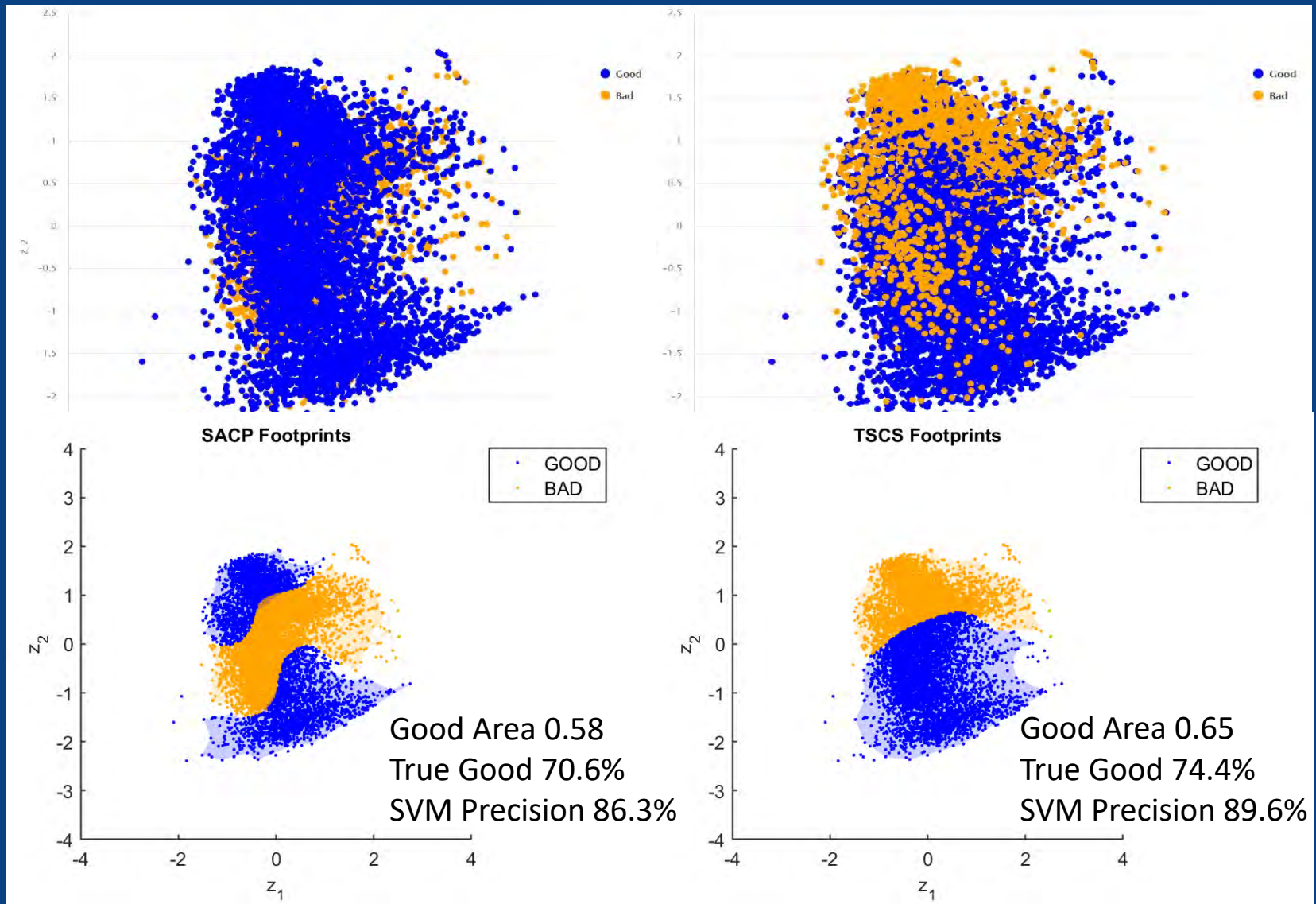


Step 3: Visualise Algorithm Footprints

- In which regions is an algorithm expected to perform well or poorly?

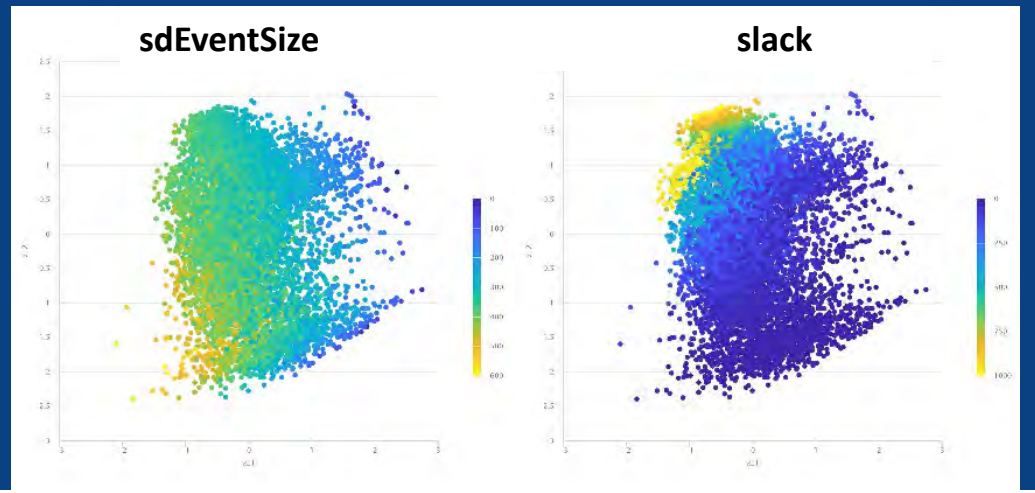
  → machine learning

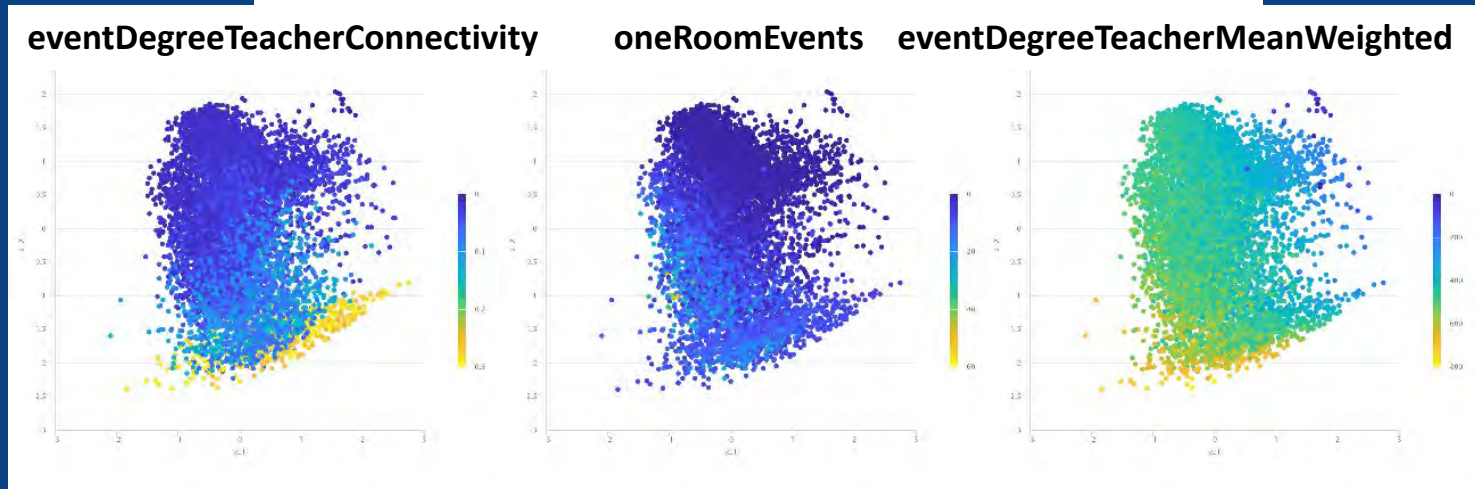
- How large is its footprint, relative to other algorithms?
- Does its footprint overlap real-world instances?
- Is it unique anywhere?



Step 4: Explain Strengths and Weaknesses

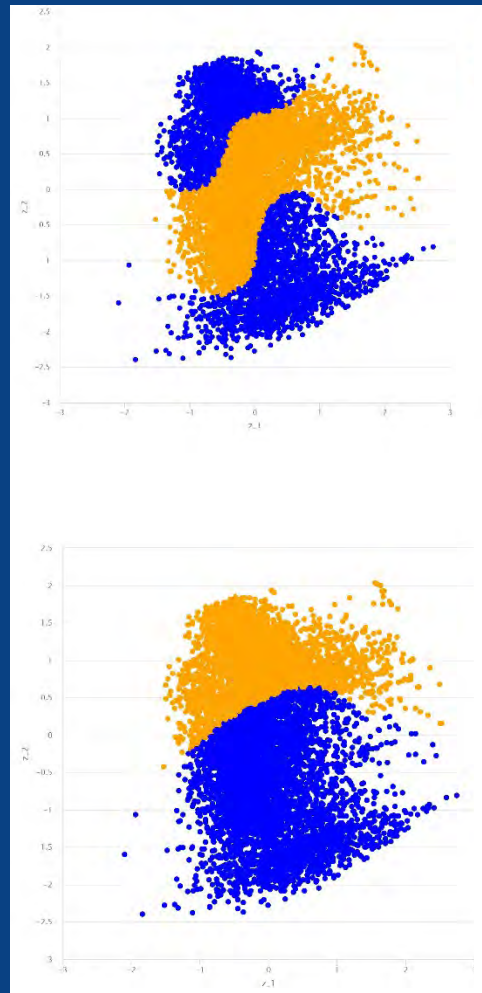
- Examine the distribution of features across the instance space
  - Correlations with directions of hardness?
- How can we describe instances in each class?
- How can we describe instances falling within (strengths) and outside (weaknesses) of algorithm footprints?
- Are selected features adequate?

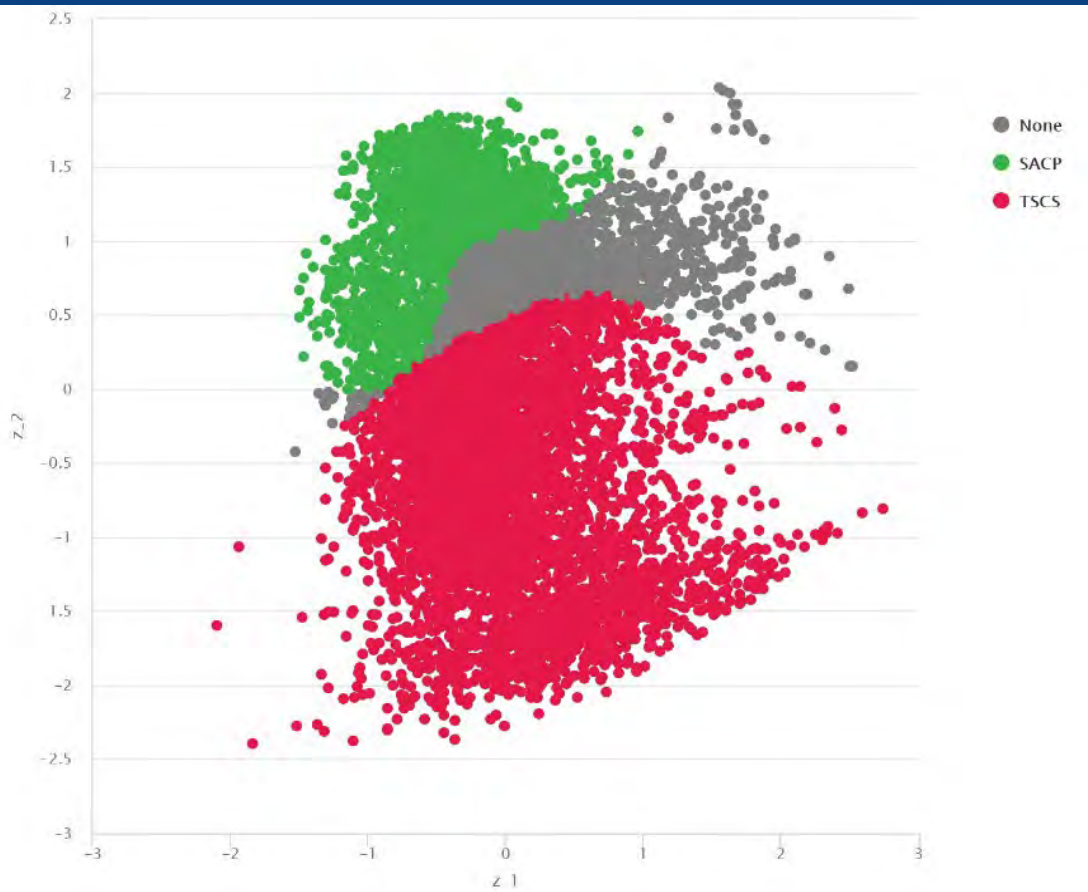




Step 5: Automated Algorithm Selection

- Combining each algorithm's SVM predictions provides an algorithm recommendation for each instance
  - Which algorithm is predicted "good"?
  - SVM with highest precision breaks ties
- No algorithm recommended if no SVM is confident
  - Suggests need for more instances or better features to separate mixed results





Pr(Good, always SACP)=0.706
Pr(Good, always TSCS)=0.744
Pr(Good, Algorithm Selector)=0.839

Post-ISA Insights: Instances

- Udine instances have very different properties to the random generator
  - much lower number of oneRoomEvents
  - lower values of node degree for the two graphs TeacherConnectivity and TeacherMeanWeighted
- Udine instances have lower slack values compared to the real-world-like generator
- Harder instances (more constraint violations) created by larger sdEventSize, but not discriminating nor Udine-like
- Insufficient instances around Udine instances for SVMs to have confidence
  - both methods could find good solutions but hard to predict best
- Udine instances do not allow us to identify unique strengths and weaknesses of TSCS and SACP
  - easy for both, and if one doesn't beat the other, it is very close
  - Real-world-like instances are more discriminating

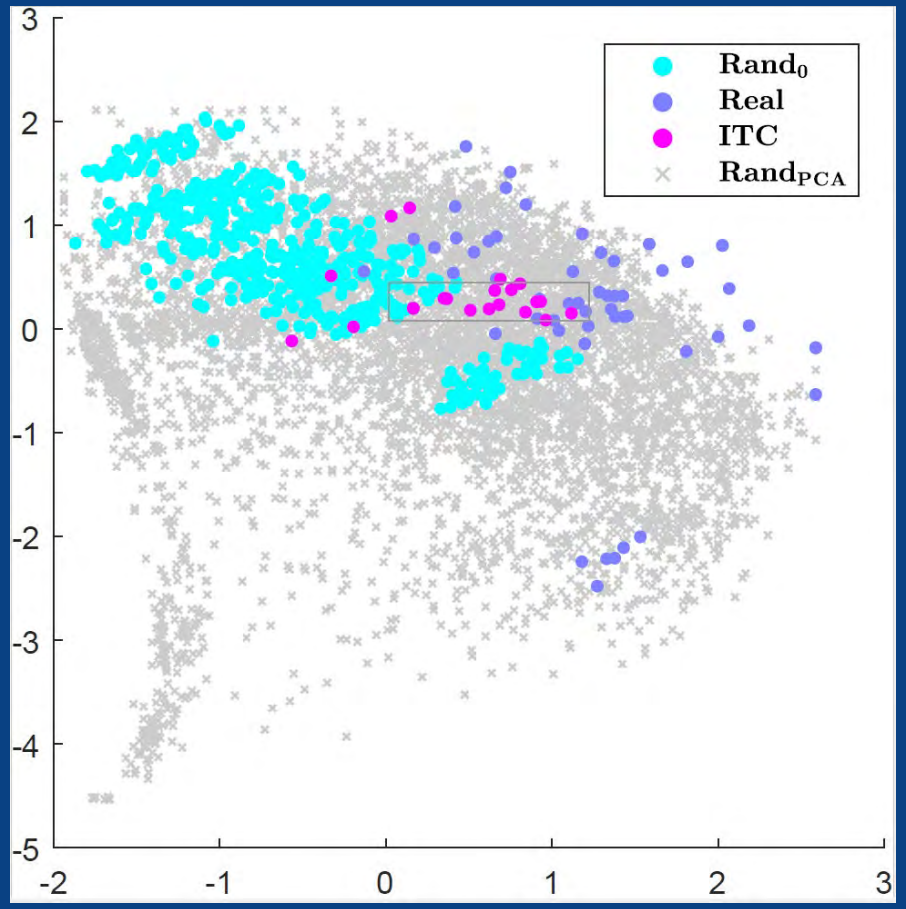
Post-ISA Insights: Algorithms

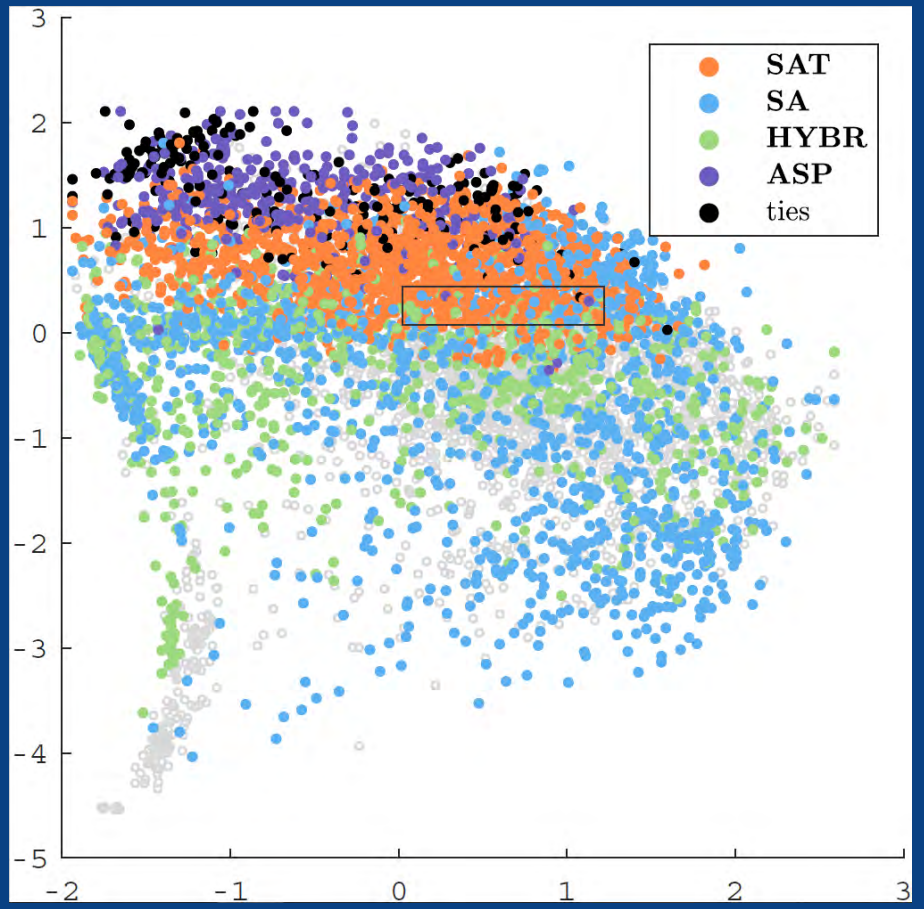
- Both algorithms perform similarly on instances defined by
  - larger values of node degree for Teacher Connectivity graph (easy for both)
  - smaller node degree for Teacher Weighted graph (hard for both)
  - lower slack values (easy for both)
  - lower sdEventSize (easy for both)
- SACP performs better when Teacher Connectivity node degree is low, and slack is high
- TSCS performs better for mid-range slack values and more one room events

INSTANCES	SACP best	TSCS best	tied	total
Random	475	957	3060	4492
Real-world-like	1613	1442	631	3686
Udine	8	10	3	21
total	2096	2409	3694	8199
%	25.56%	29.38%	45.05%	

An Iterative Process …

- More features
- More instances classes
- More algorithms
- More collaborators and insights!





Journal of Scheduling
https://doi.org/10.1007/s10951-021-00701-x

Algorithm selection and instance space analysis for curriculum-based course timetabling

Arnaud De Coster[1] · Nysret Musliu[2] · Andrea Schaerf[3] · Johannes Schoisswohl[1] · Kate Smith-Miles[4]

Accepted: 1 July 2021

Abstract

We propose an algorithm selection approach and an instance space analysis for the well-known curriculum-based course timetabling problem (CB-CTT), which is an important problem for its application in higher education. Several state of the art algorithms exist, including both exact and metaheuristic methods. Results of these algorithms on existing instances in the literature show that there is no single algorithm outperforming the others. Therefore, a deep analysis of the strengths and weaknesses of these algorithms, depending on the instance, is an important research question. In this work, a detailed analysis of the instance space for CB-CTT is performed, charting the regions where these algorithms perform best. We further investigate the application of machine learning methods to automated algorithm selection for CB-CTT, strengthening the insights gained through the instance space analysis. For our research, we contribute new real-life instances and extend the generation of synthetic instances to better correspond to these new instances. Finally, this work shows how instance space analysis and the application of algorithm selection complement each other, underlining the value of both approaches in understanding algorithm performance.

Keywords Timetabling · Scheduling · Algorithm selection · Classification · Instance space · Instance generation

matilda.unimelb.edu.au/matilda/

THE UNIVERSITY OF MELBOURNE

The University of Melbourne

SEARCH LOGIN MENU

MATILDA

Melbourne Algorithm Test Instance Library with Data Analytics

ABOUT MATILDA | OUR TEAM | OUR METHODOLOGY | GETTING STARTED

MATILDA: Motivation

- An online tool to enable researchers to access Instance Space Analysis
  - Generate instance spaces, and include visualisations in their papers
  - Provide opportunity for more insights
- An online library of successful case studies for Instance Space Analysis, spanning a range of applications in “algorithmic science”
  - Optimisation
  - Machine Learning
  - Time Series Forecasting
  - Software Testing
- An online repository of benchmark instances and feature calculation code

MATILDA: The Acronym

- MATILDA = Melbourne Algorithm Test Instance Library with Data Analytics
- Besides being an acronym, it is also a symbolic name for Australians
  - The iconic song “Waltzing Matilda” is widely regarded as Australia’s unofficial national anthem
  - The Australian Women’s Soccer Team are known as “The Matildas”
- MATILDA also pays tribute to the many women who have contributed enormously to the advancement of knowledge and innovation, but not received the recognition they deserve due to societal biases known as “The Matilda Effect”
  - We acknowledge the contributions of the invisible and unapplauded, on whose shoulders each of us stands today.

MATILDA: The Logo

- The MATILDA logo is also symbolic
  - the constellations have the appearance of an instance space, with the boundary joining the dominant stars (labelled as M-A-T-I-L-D-A) reminiscent of an algorithm footprint
  - the path joining the M-A-T-I-L-D-A stars is a Hamiltonian cycle, relating to one of our library problems (TSP)
  - the M-A-T-I-L-D-A stars contain the Southern Cross, the best known and most easily recognised star formation in the Southern Hemisphere, as viewed from Melbourne where the MATILDA team is based

MATILDA: Library Problems

Final methodology:
Smith-Miles, K. and Muñoz, M. A., "Instance Space Analysis for Algorithm Testing: Methodology and Software Tools", *ACM Computing Surveys*, vol. 55, no. 12, 2023.

- Optimisation
  - Travelling Salesman Problem
  - Graph Colouring Problem
  - Knapsack Problem
  - Job Shop Scheduling
  - Timetabling
  - Bin-packing
  - Maximum Flow
  - Rotating Workforce Scheduling
  - MaxCut Graph Problem
  - Black-Box Optimisation
  - Multiobjective Black-Box Optimisation
  - Multifidelity Black-Box Optimisation
  - Mixed Integer Programming
  - Vehicle Routing
  - Container Loading
- Learning and Model Fitting
  - Machine Learning Classification
  - Time Series Forecasting
  - Anomaly Detection
  - Facial Age Estimation
  - Regression
  - Clustering
  - Reinforcement Learning
  - Consumer Choice Modelling
  - Multi-fidelity Surrogate Modelling
- Software Testing
  - Automated test suite generation
  - Mutation testing
  - Quantum Computing …

Graph Colouring

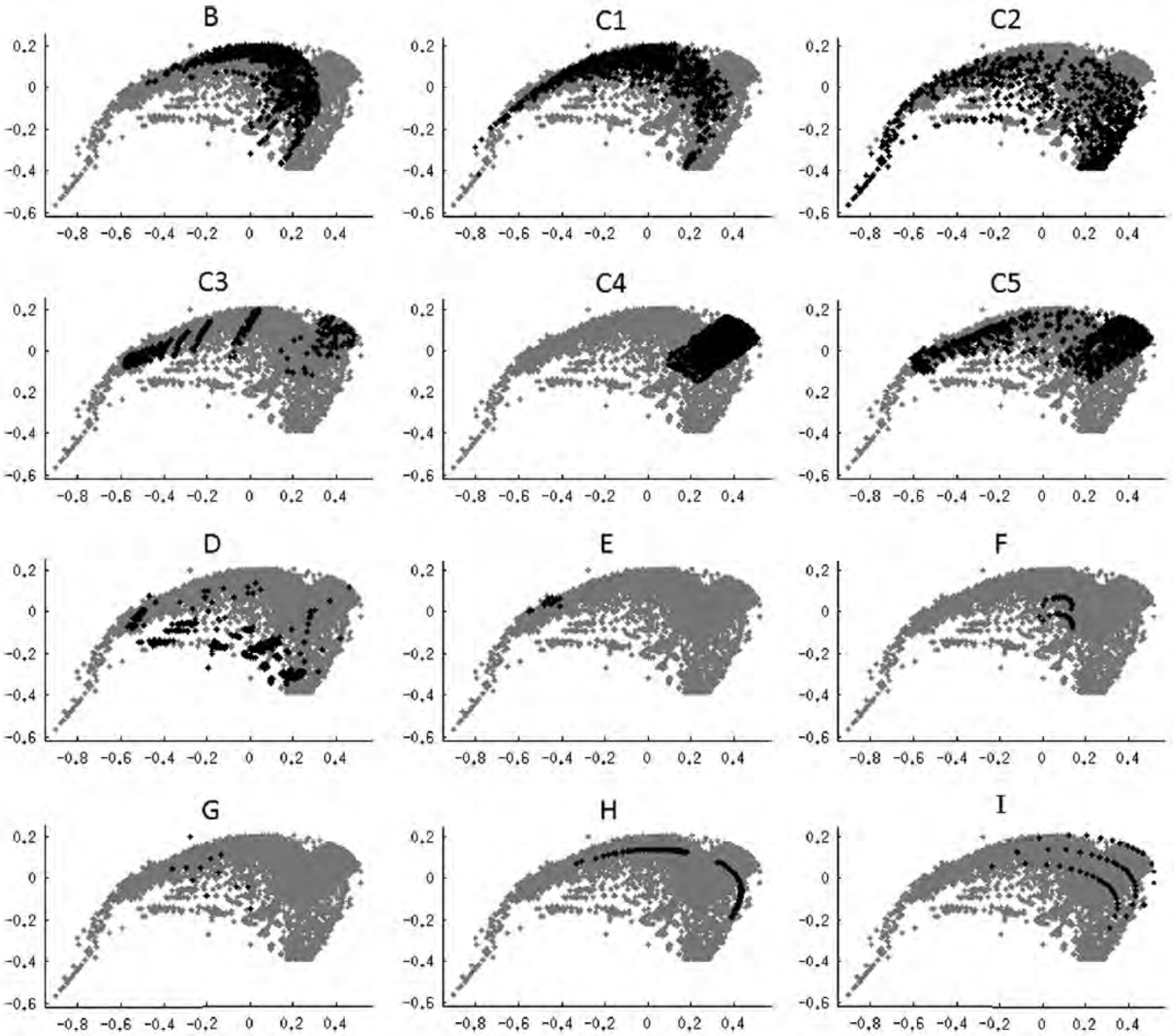


Fig. 2. Each instance sets shown as black points in the instance space, ordered alphabetically from set B (shown at top left) to set I (shown at bottom right). The grey points define the entire instance space.

Computers & Operations Research 45 (2014) 12–24

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Towards objective measures of algorithm performance across instance space

Kate Smith-Miles [a,*], Davaatseren Baatar [a], Brendan Wreford [a], Rhyd Lewis [b]

[a] School of Mathematical Sciences, Monash University, Victoria 3800, Australia
[b] School of Mathematics, Cardiff University, Wales, United Kingdom

ARTICLE INFO

Available online 6 December 2013

Keywords:
Comparative analysis
Heuristics
Graph coloring
Algorithm selection
Performance prediction

ABSTRACT

This paper tackles the difficult but important task of objective algorithm performance assessment for optimization. Rather than reporting average performance of algorithms across a set of chosen instances, which may bias conclusions, we propose a methodology to enable the strengths and weaknesses of different optimization algorithms to be compared across a broader instance space. The results reported in a recent *Computers and Operations Research* paper comparing the performance of graph coloring heuristics are revisited with this new methodology to demonstrate (i) how pockets of the instance space can be found where algorithm performance varies significantly from the average performance of an algorithm; (ii) how the properties of the instances can be used to predict algorithm performance on previously unseen instances with high accuracy; and (iii) how the relative strengths and weaknesses of each algorithm can be visualized and measured objectively.

Computers & Operations Research 63 (2015) 102–113

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Generating new test instances by evolving in instance space

Kate Smith-Miles *, Simon Bowly

School of Mathematical Sciences, Monash University, Victoria 3800, Australia

ARTICLE INFO

Available online 9 May 2015

Keywords:
Test instances
Benchmarking
Graph colouring
Instance space
Evolving instances

ABSTRACT

Our confidence in the future performance of any algorithm, including optimization algorithms, depends on how carefully we select test instances so that the generalization of algorithm performance on future instances can be inferred. In recent work, we have established a methodology to generate a 2-d representation of the instance space, comprising a set of known test instances. This instance space shows the similarities and differences between the instances using measurable features or properties, and enables the performance of algorithms to be viewed across the instance space, where generalizations can be inferred. The power of this methodology is the insights that can be generated into algorithm strengths and weaknesses by examining the regions in instance space where strong performance can be expected. The representation of the instance space is dependent on the choice of test instances however. In this paper we present a methodology for generating new test instances with controllable properties, by filling observed gaps in the instance space. This enables the generation of rich new sets of test instances to support better the understanding of algorithm strengths and weaknesses. The methodology is demonstrated on graph colouring as a case study.

Evolving new graphs

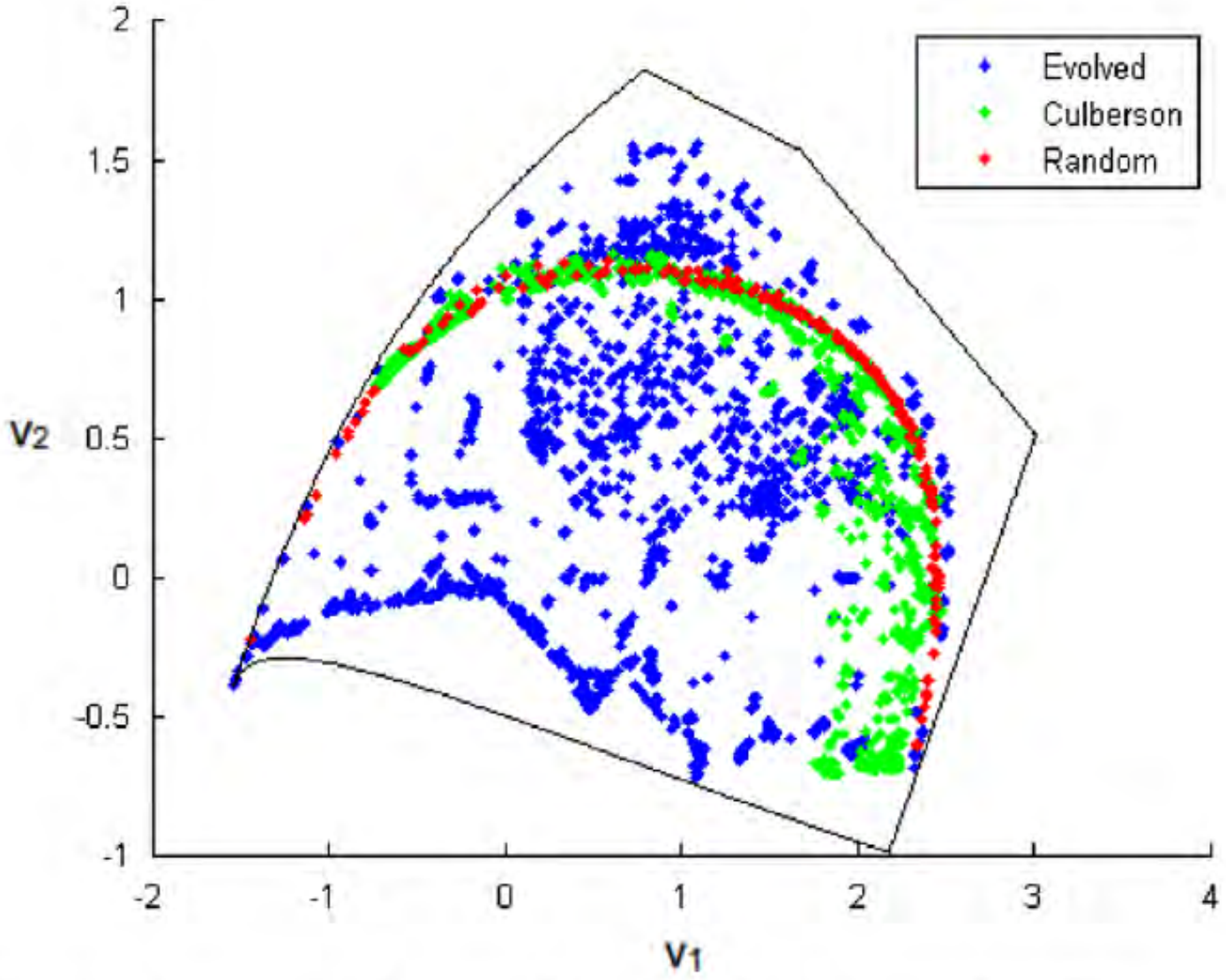


Fig. 8. Graph instances and expected boundary for the normalized PCA space, for 100 node graphs, showing the location of random and Culberson graphs and our newly evolved graphs. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

Mach Learn (2018) 107:109–147
https://doi.org/10.1007/s10994-017-5629-5

Instance spaces for machine learning classification

Mario A. Muñoz[1] · **Laura Villanova**[1] · **Davaatseren Baatar**[1] · **Kate Smith-Miles**[1]

Received: 10 May 2016 / Accepted: 20 January 2017 / Published online: 28 December 2017

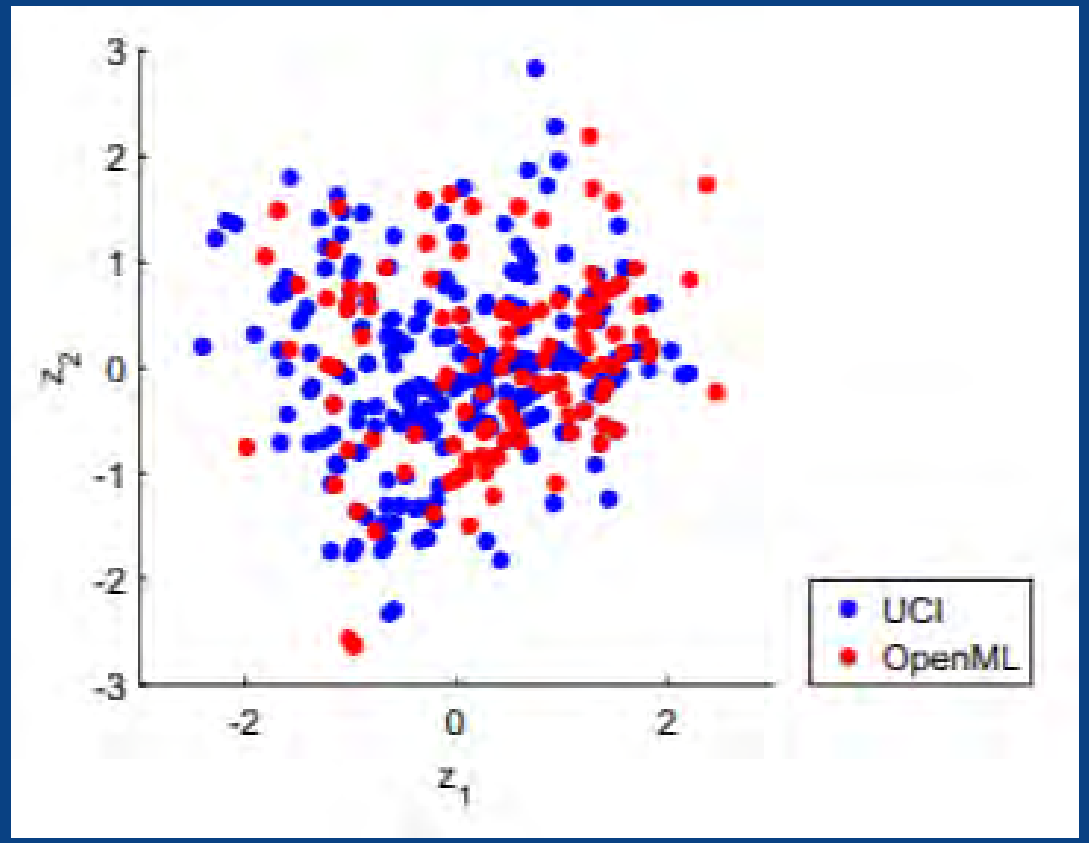
Abstract This paper tackles the issue of objective performance evaluation of machine learning classifiers, and the impact of the choice of test instances. Given that statistical properties or features of a dataset affect the difficulty of an instance for particular classification algorithms, we examine the diversity and quality of the UCI repository of test instances used by most machine learning researchers. We show how an instance space can be visualized, with each classification dataset represented as a point in the space. The instance space is constructed to reveal pockets of hard and easy instances, and enables the strengths and weaknesses of individual classifiers to be identified. Finally, we propose a methodology to generate new test instances with the aim of enriching the diversity of the instance space, enabling potentially greater insights than can be afforded by the current UCI repository.

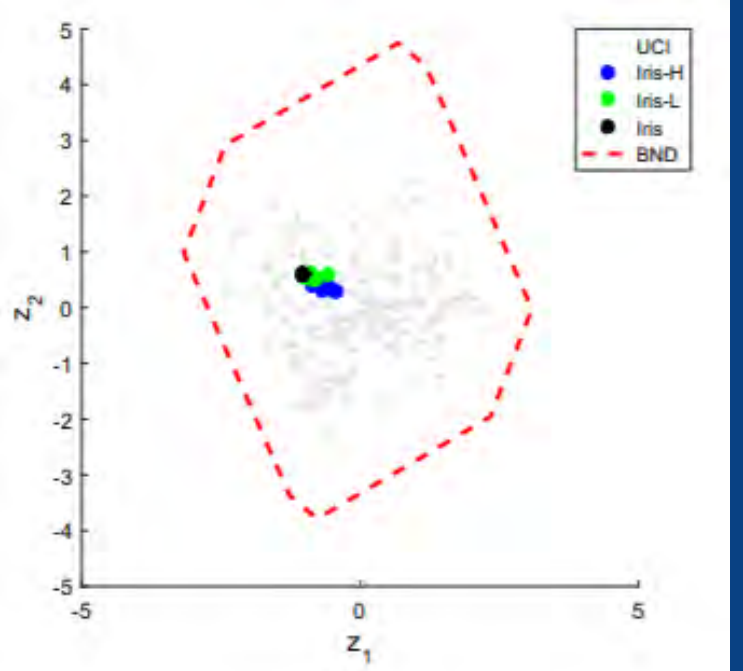
Keywords Classification · Meta-learning · Test data · Instance space · Performance evaluation · Algorithm footprints · Test instance generation · Instance difficulty

1 Introduction

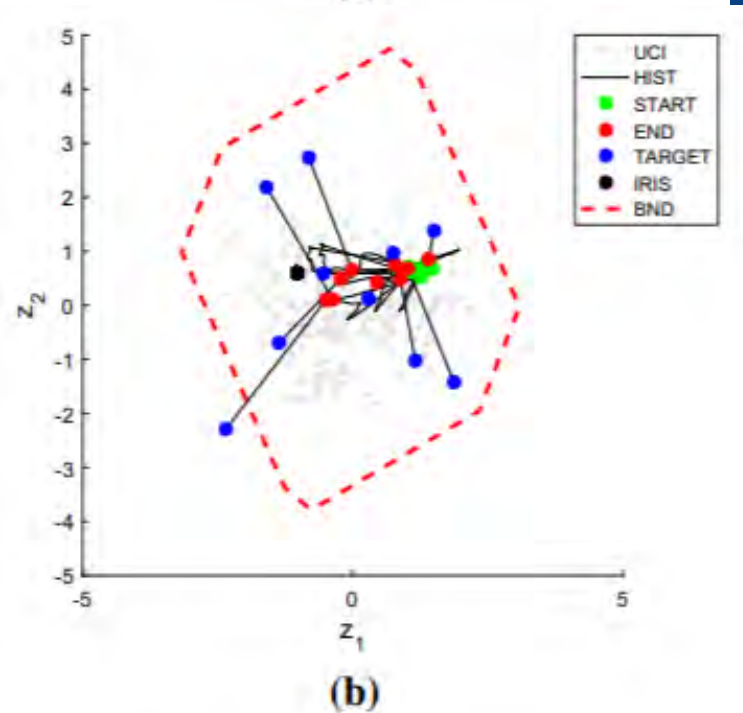
The practical importance of machine learning (ML) has resulted in a plethora of algorithms in recent decades (Carbonell et al. 1983; Flach 2012; Jordan and Mitchell 2015). Are new

Machine Learning (UCI Repository and OpenML)





(a)



(b)

Computers and Operations Research 128 (2021) 105184

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/caor

Revisiting *where are the hard knapsack problems?* via Instance Space Analysis

Kate Smith-Miles *, Jeffrey Christiansen, Mario Andrés Muñoz

School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia

ARTICLE INFO

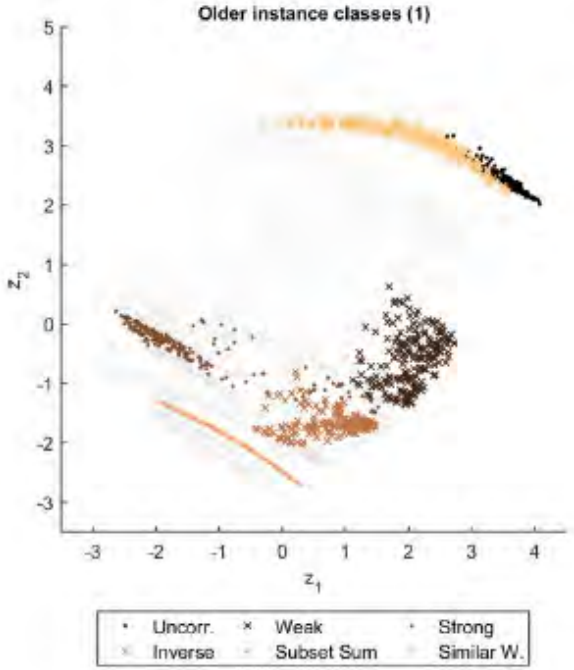
Article history:
Received 17 June 2020
Revised 7 December 2020
Accepted 10 December 2020
Available online 18 December 2020

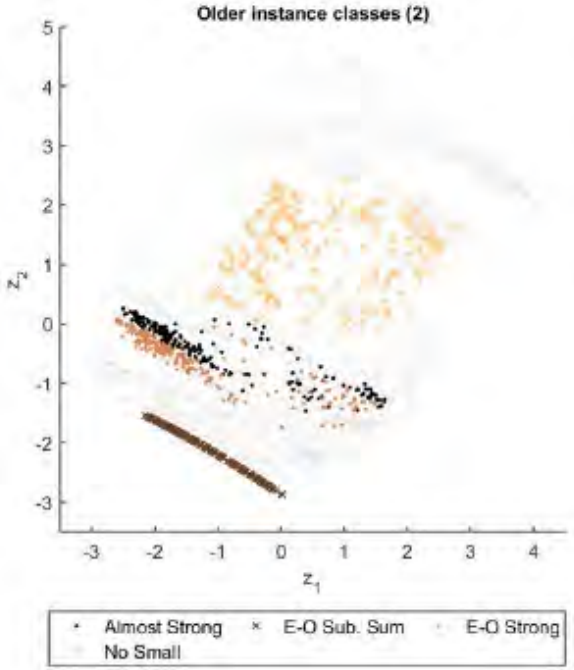
Keywords:
0–1 Knapsack problem
Instance Space Analysis
Instance generation
Instance difficulty
Performance evaluation
Algorithm portfolios
Algorithm selection

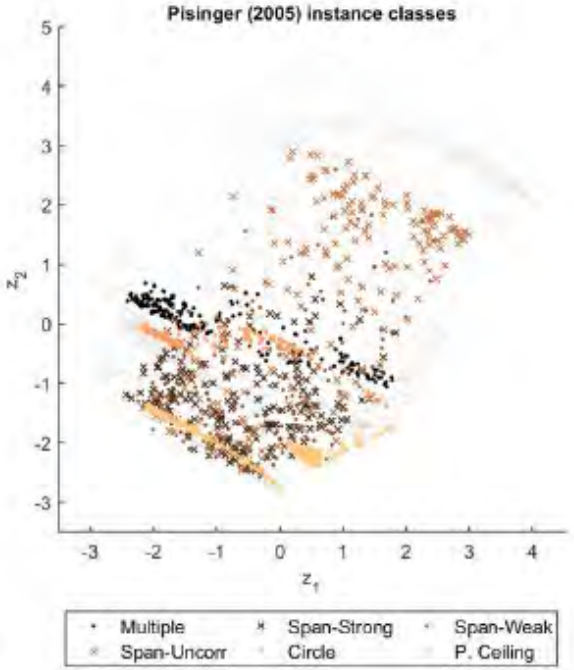
ABSTRACT

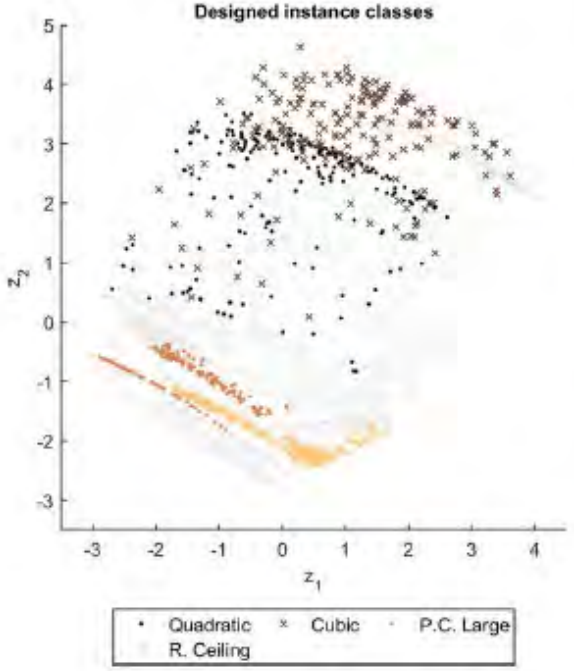
In 2005, David Pisinger asked the question "*where are the hard knapsack problems?*". Noting that the classical benchmark test instances were limited in difficulty due to their selected structure, he proposed a set of new test instances for the 0–1 knapsack problem with characteristics that made them more challenging for dynamic programming and branch-and-bound algorithms. This important work highlighted the influence of diversity in test instances to draw reliable conclusions about algorithm performance. In this paper, we revisit the question in light of recent methodological advances – in the form of Instance Space Analysis – enabling the strengths and weaknesses of algorithms to be visualised and assessed across the broadest possible space of test instances. We show where the hard instances lie, and objectively assess algorithm performance across the instance space to articulate the strengths and weaknesses of algorithms. Furthermore, we propose a method to fill the instance space with diverse and challenging new test instances with controllable properties to support greater insights into algorithm selection, and drive future algorithmic innovations.

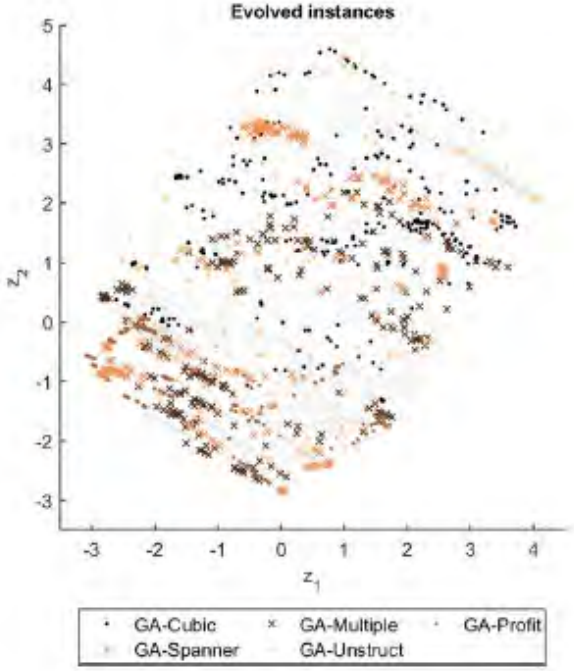
Knapsack

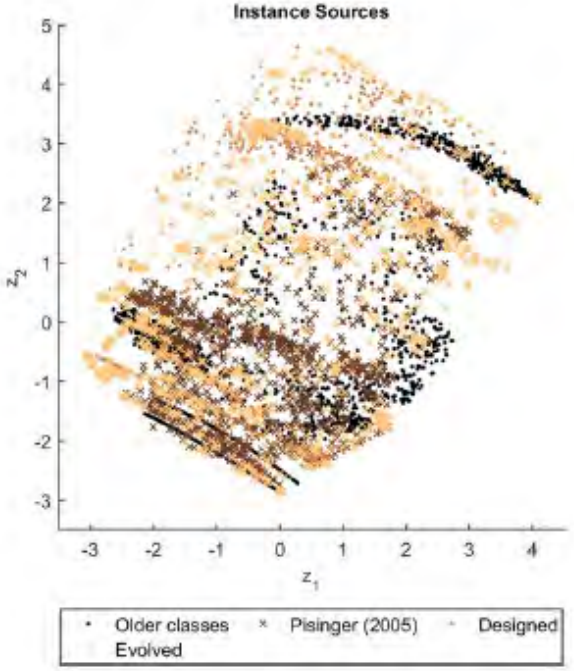

Older instance classes (1)
Uncorr.
Weak
Strong
Inverse
Subset Sum
Similar W.


Older instance classes (2)
Almost Strong
E-O Sub. Sum
E-O Strong
No Small


Pisinger (2005) instance classes
Multiple
Span-Strong
Span-Weak
Span-Uncorr
Circle
P. Ceiling


Designed instance classes
Quadratic
Cubic
P.C. Large
R. Ceiling


Evolved instances
GA-Cubic
GA-Multiple
GA-Profit
GA-Spanner
GA-Unstruct


Instance Sources
Older classes
Pisinger (2005)
Designed
Evolved

European Journal of Operational Research 304 (2023) 411–428

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Enhanced instance space analysis for the maximum flow problem

Hossein Alipour*, Mario Andrés Muñoz, Kate Smith-Miles

School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia

ARTICLE INFO

Article history:
Received 14 June 2021
Accepted 9 April 2022
Available online 14 April 2022

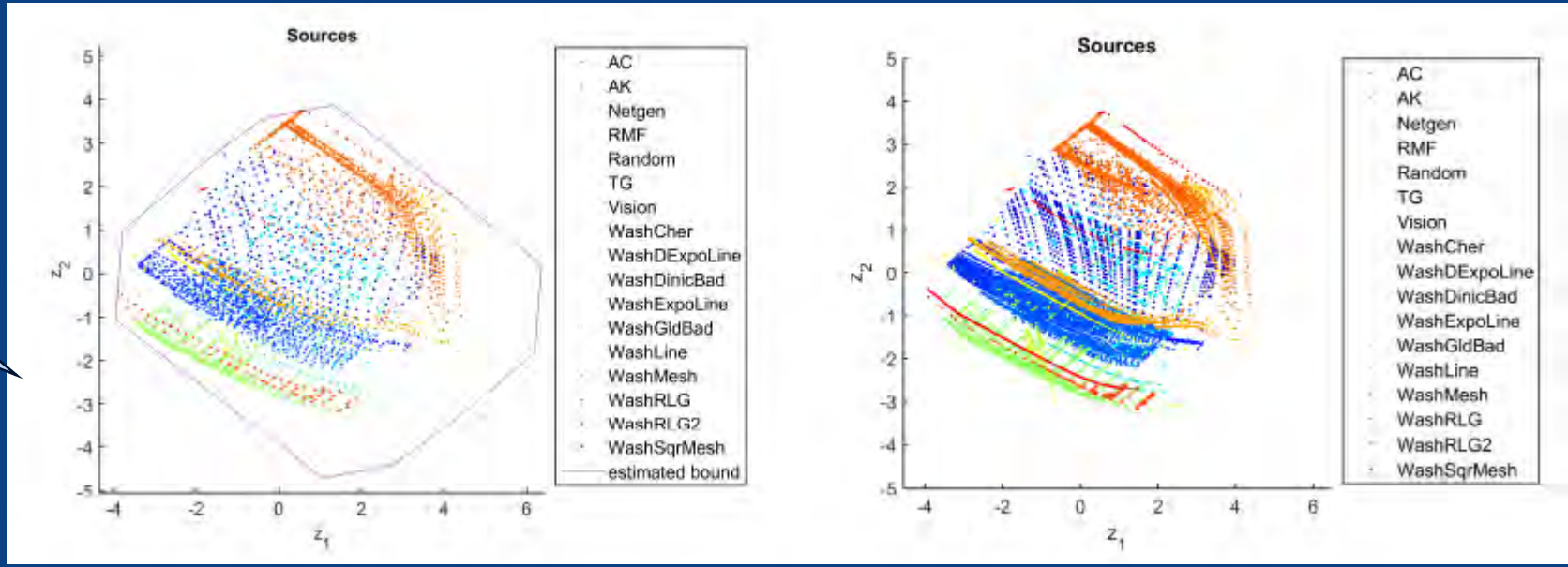
Keywords:
Validation of OR computations
Maximum flow
Instance space analysis
Instance selection
Feature selection

ABSTRACT

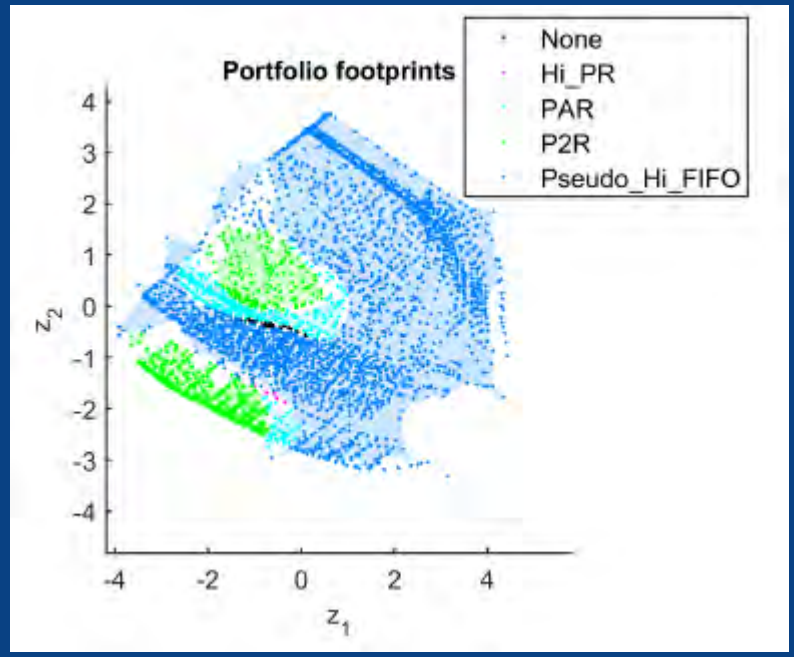
The Maximum Flow Problem (MFP) is a fundamental network flow theory problem, for which many algorithms, supported by strong theoretical worst-case analyses, have been proposed. However, their practical efficiency depends on the network structure, making it unclear which algorithm is best for a particular instance or a class of MFP. Instance Space Analysis (ISA) is a methodology that provides insights into such per-instance analysis. In this paper, the instance space of MFP is constructed and analysed for the first time. Novel features from the networks are extracted, capturing the performance of MFP algorithms. Additionally, this paper expands the ISA methodology by addressing the issue of how benchmark instances should be selected to reduce bias in the analysis. Using the enhanced ISA methodology with MFP as the case study, this paper demonstrates that the most important features can be detected, and machine learning methods can identify their impact on algorithm performance, whilst reducing the bias caused by over-representation within the selected sample of test instances. The enhanced methodology enables new insights into the performance of state-of-the-art general purpose MFP algorithms, as well as recommendations for the construction of comprehensive and unbiased benchmark test suites for MFP algorithm testing.

New instance filter method to reduce bias, defining critical and redundant instances

MaxFlow

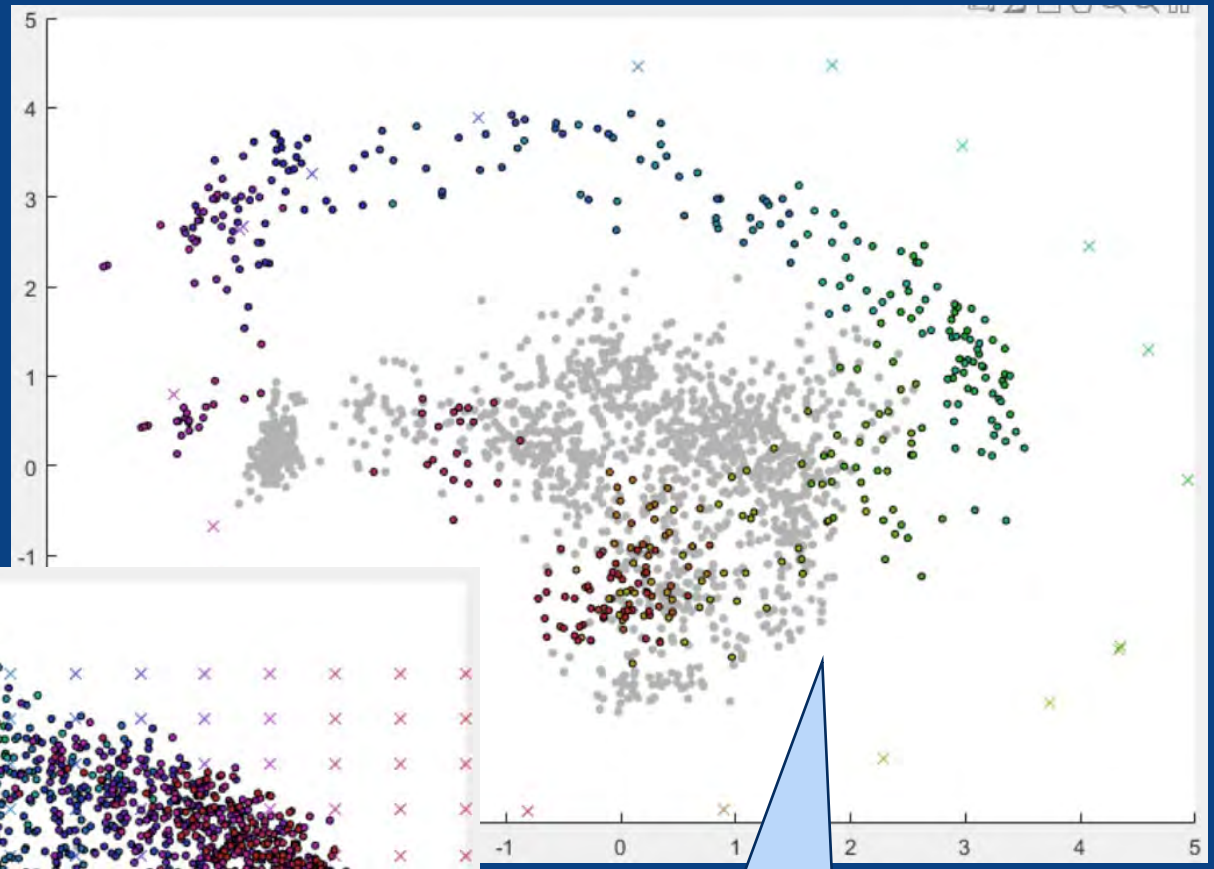


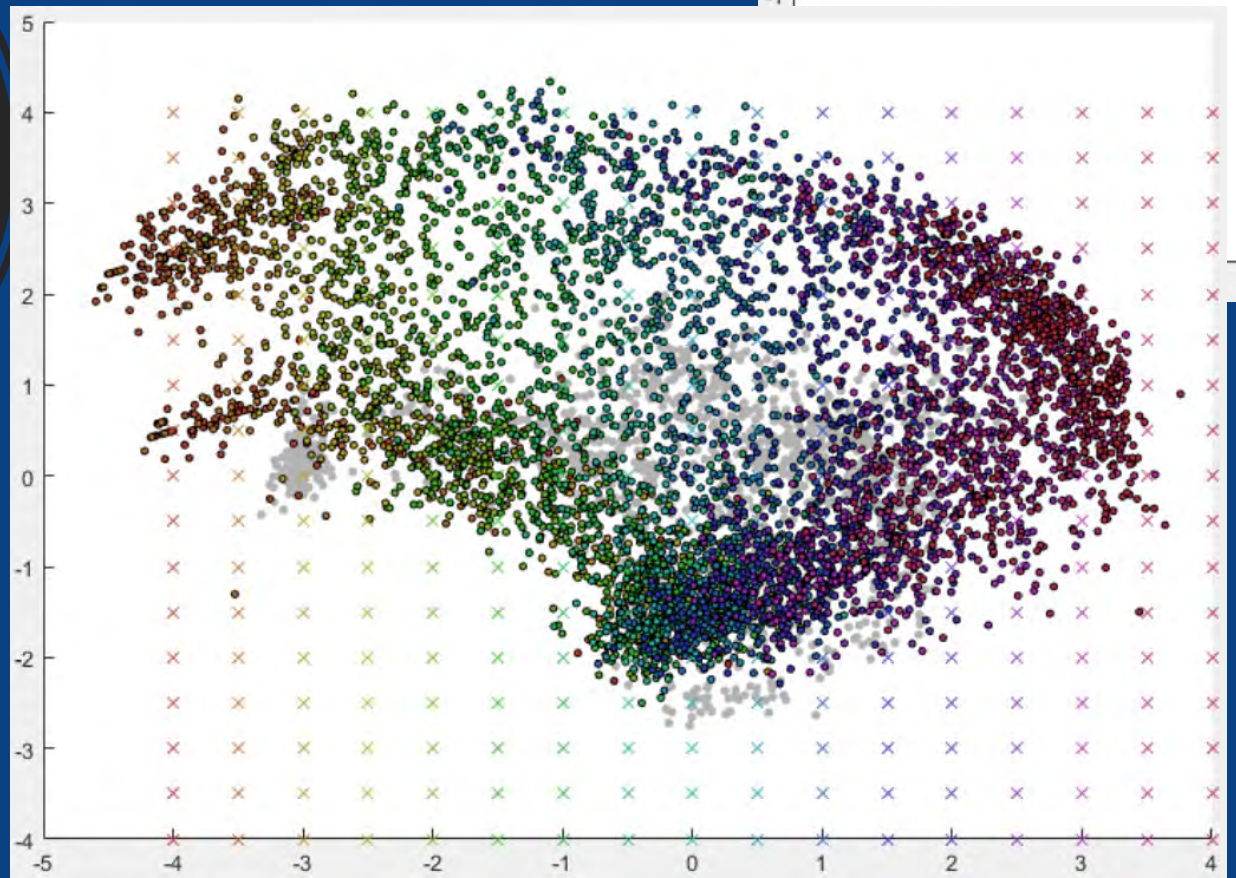
$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} -0.4208 & -0.0738 \\ -1.0599 & 0.7700 \\ 0.0458 & 0.5908 \\ 0.4662 & 0.3763 \\ -0.3384 & -0.2591 \\ 0.4274 & 0.2908 \end{bmatrix}^{T} \begin{bmatrix} Order \\ Size \\ AvNdDg \\ AVScPotNetExcess \\ cvNdDg \\ ScCapDens \end{bmatrix}$$



Silva, E., Oliveira, J. F., and Wäscher, G. (2014). 2dcpackgen: A problem generator for two-dimensional rectangular cutting and packing problems. European Journal of Operational Research, 237(3):846–856.

Bin-packing





Martello, S. and Vigo, D., Mgmt Sci, 1998
and
Berkey, J. O. and Wang, P. Y., JORS, 1987

International Journal of Forecasting 33 (2017) 345–358

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Visualising forecasting algorithm performance using time series instance spaces

Yanfei Kang [a,*], Rob J. Hyndman [b], Kate Smith-Miles [c]

[a] School of Economics and Management, Beihang University, Beijing, 100191, China
[b] Department of Econometrics and Business Statistics, Monash University, Clayton VIC 3800, Australia
[c] School of Mathematical Sciences, Monash University, Clayton VIC 3800, Australia

ARTICLE INFO

Keywords:
M3-Competition
Time series visualisation
Time series generation
Forecasting algorithm comparison

ABSTRACT

It is common practice to evaluate the strength of forecasting methods using collections of well-studied time series datasets, such as the M3 data. The question is, though, how diverse and challenging are these time series, and do they enable us to study the unique strengths and weaknesses of different forecasting methods? This paper proposes a visualisation method for collections of time series that enables a time series to be represented as a point in a two-dimensional instance space. The effectiveness of different forecasting methods across this space is easy to visualise, and the diversity of the time series in an existing collection can be assessed. Noting that the diversity of the M3 dataset has been questioned, this paper also proposes a method for generating new time series with controllable characteristics in order to fill in and spread out the instance space, making our generalisations of forecasting method performances as robust as possible.

Time Series Forecasting (M3 Competition)

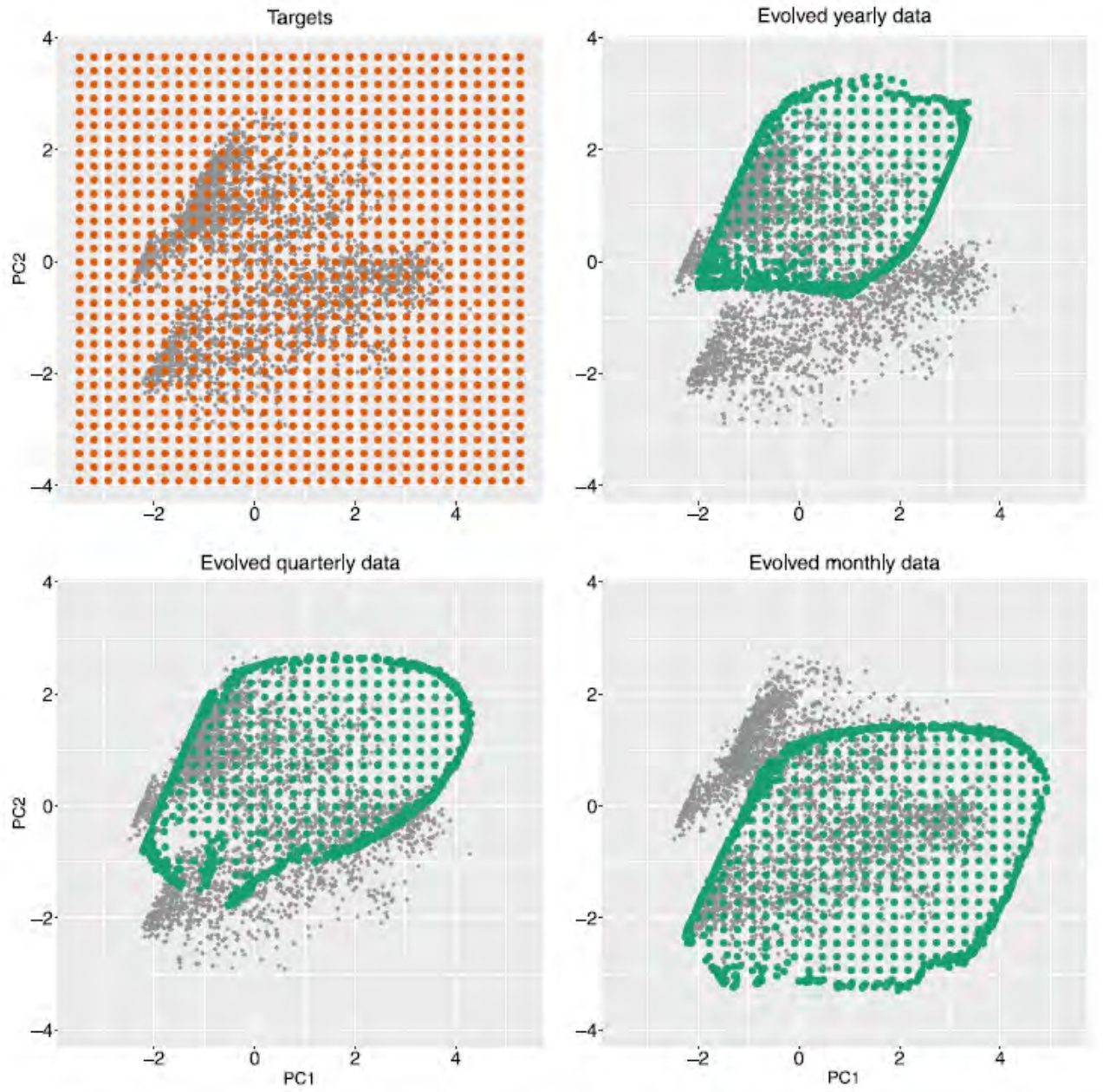


Fig. 4. Evolved new instances using the genetic algorithm. The red points in the top-left panel are the 1024 target points we set; the green points in the top-right panel represent the 1024 yearly time series that are evolved newly from the target points using the generic algorithm. Similarly, the two bottom panels show the 1024 evolved quarterly and monthly time series, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Using MATILDA

- There are two ways of performing an Instance Space Analysis:
  - Download the MATLAB source code from GitHub
    - ➢ Run the automated pipeline
    - ➢ Use the live script
  - Use the tools available on MATILDA website

- Online tutorial available:

https://matilda.unimelb.edu.au/matilda/matildadata/tutorials/matilda-technical-details.mp4

MATILDA user accounts created from over 80 research groups worldwide since launch mid 2019

Open-source access for academic purposes

Contract research for proof-of-concept studies for commercial purposes

BOEING

Australian Government
Department of Defence
Science and Technology

MATILDA: Library Problems

Final methodology:
Smith-Miles, K. and Muñoz, M. A., "Instance Space Analysis for Algorithm Testing: Methodology and Software Tools", *ACM Computing Surveys*, vol. 55, no. 12, 2023.

- Optimisation
  - Travelling Salesman Problem
  - Graph Colouring Problem
  - Knapsack Problem
  - Job Shop Scheduling
  - Timetabling
  - Bin-packing
  - Maximum Flow
  - Rotating Workforce Scheduling
  - MaxCut Graph Problem
  - Black-Box Optimisation — Lecture 3
  - Multiobjective Black-Box Optimisation
  - Multifidelity Black-Box Optimisation
  - Mixed Integer Programming
  - Vehicle Routing
  - Container Loading
- Learning and Model Fitting
  - Machine Learning Classification
  - Time Series Forecasting
  - Anomaly Detection
  - Facial Age Estimation — Lecture 2
  - Regression
  - Clustering
  - Reinforcement Learning
  - Consumer Choice Modelling
  - Multi-fidelity Surrogate Modelling
- Software Testing
  - Automated test suite generation
  - Mutation testing
  - Quantum Computing …

THE UNIVERSITY OF MELBOURNE

QUESTIONS?

MATILDA:

https://matilda.unimelb.edu.au

MATLAB code:

https://github.com/andremun/InstanceSpace

Please contact us for any enquiries or support:

matilda-team@unimelb.edu.au

smith-miles@unimelb.edu.au

THE UNIVERSITY OF MELBOURNE

Lecture 2

In search of trustworthy algorithms ... show us the stress-testing!

Lecture 1 (Part 1)

- Introduction to Instance Space Analysis
- Introduction to MATILDA
- Case Study: Optimisation (University Timetabling)

Lecture 2

- Case Study: Computer Vision (Facial Age Estimation)

Lecture 3

- Evolving new instances to fill an instance space (Black-box optimisation and new artwork!)

THE UNIVERSITY OF MELBOURNE

Part 2

How to perform an Instance Space Analysis

Mario Andrés Muñoz : munoz.m@unimelb.edu.au

https://matilda.unimelb.edu.au/matilda/matildadata/tutorials/matilda-technical-details.mp4

- Using the MATLAB code/live script
- Using MATILDA's web Interface
- Case Study: Machine Learning (Classification)

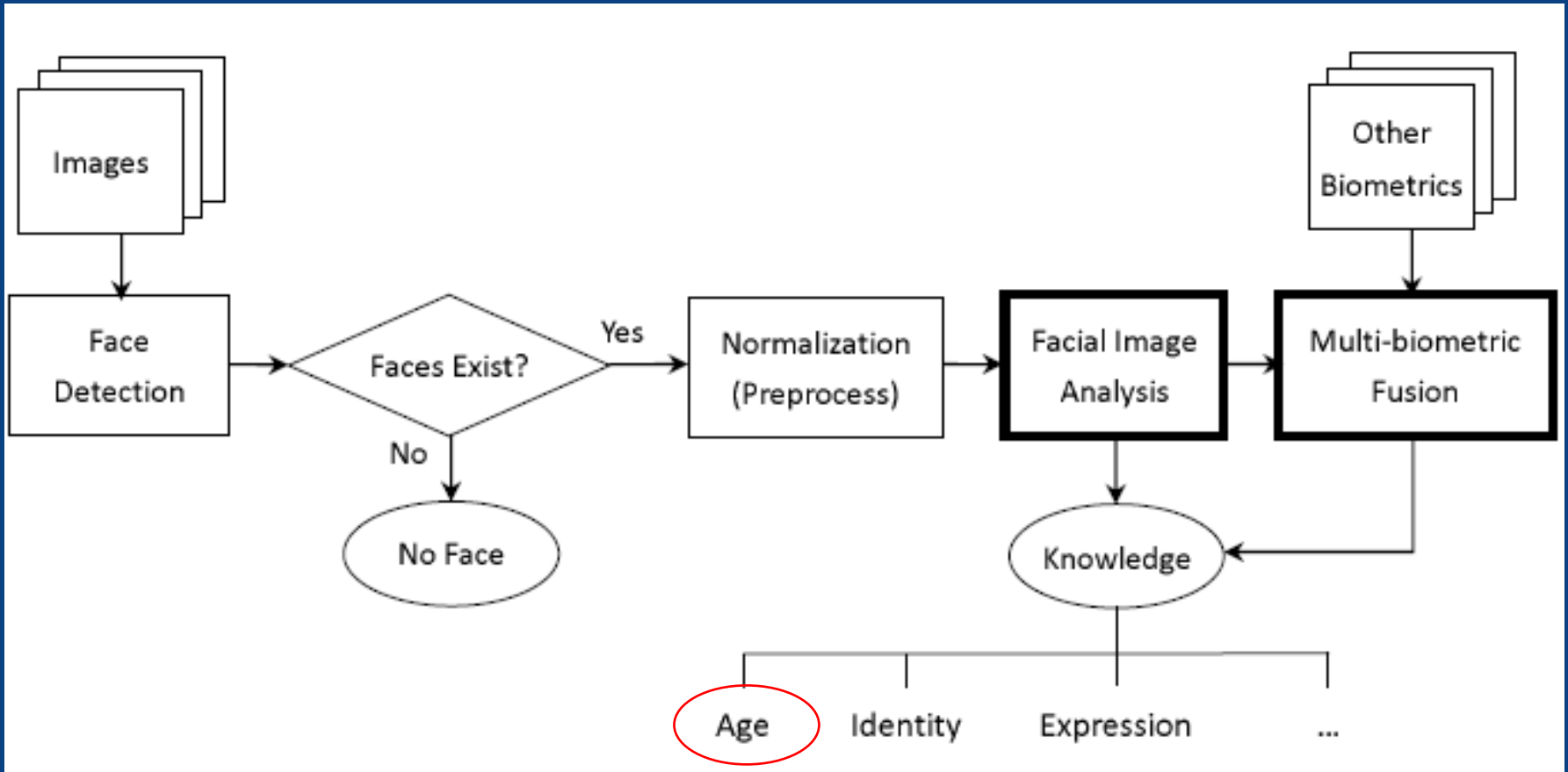
Lecture 2 Outline

- The importance of facial age estimation
- Our proposed algorithm (2007)
- Experimental results
- Conclusions

Published in Geng, X., Zhou, Z.-H., and Smith-Miles, K. A., "Automatic Age Estimation Based on Facial Aging Patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234-2240, 2007.

- Revisiting a decade later
- Stress-testing via Instance Space Analysis … do our original 2007 conclusions still hold?

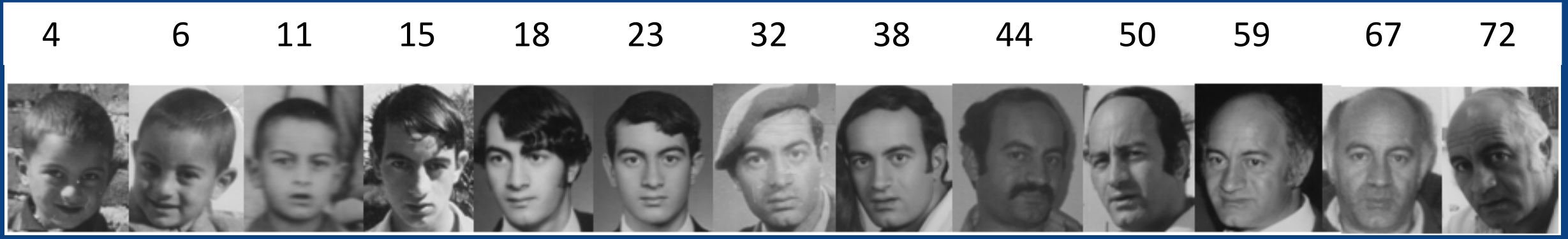

PERSONNEL
Ledger Nrgo
« WE NEED YOUR EXACT AGE... 'WOODSTOCK GENERATION'
ISN'T SPECIFIC ENOUGH. »



Automatic Facial Image Understanding System

Automatic Facial Age Estimation

- Why did we start researching this more than a decade ago?
  - ✓New research topic
  - ✓Existing methods needed improvement
  - ✓Important applications
  - ✓Interesting
  - ✓Challenging



Difficult?

19

49

5

Why is age estimation important?

- Age-specific Human-Computer Interaction (HCI)
  - automatically choose the vocabulary, interface, and services that are suitable for the user's age.
- Age-specific access control
  - develop devices or software that only gives access to the people at legal ages.
- Multi-cue identification/verification.
  - work together with other biometric traits, such as fingerprint, face, and iris, to improve the identification/verification accuracy.
- Law enforcement
  - provide age estimates for victims or suspects

Example Application

“Fujitaka's new cigarette vending machines employs an advanced facial recognition system that compares a buyer's bone structure, skin sag, brow wrinkles and crow's feet against a record of more than 100,000 people. If the buyer fails the visual scan, they will be required to insert their ID card into the machine in order to verify that they are of legal age to smoke.”

People have already tricked it by holding up photos from magazines!


...AND ALL THAT TIME WENDY WAS LYING TO ME ABOUT HER AGE...
offthemark.com

The start of research efforts …

FG-Net is the European working group on face and gesture recognition

First aging face database (*2002*)

- 1002 face images
- 82 subjects
- 6-18 images each subject
- Age: 0-69
- Variations: illumination, pose, expression, beards, moustaches, spectacles, hats

FG-NET Aging Database

How to convert image into numbers?

To build a model, we use a training set of annotated images where:

- landmark points have been marked on each image (Fig 1).
- Procrustes analysis is used to align the sets of points (each represented as a vector, x_s, Fig. 2) and capture the shape of the face.
- Each training image is warped so the points match those of the mean shape, obtaining a "shape-free patch" (Fig. 3). A texture vector x_t is then created via raster scan.

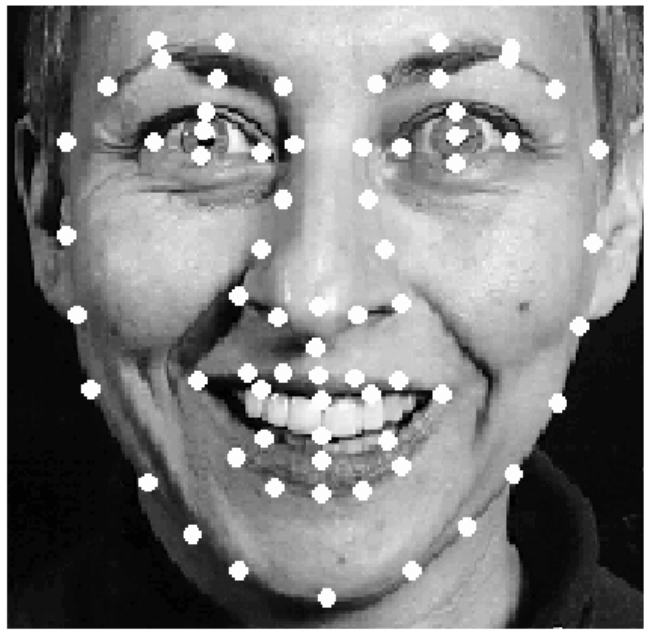


Figure 1; labelled image

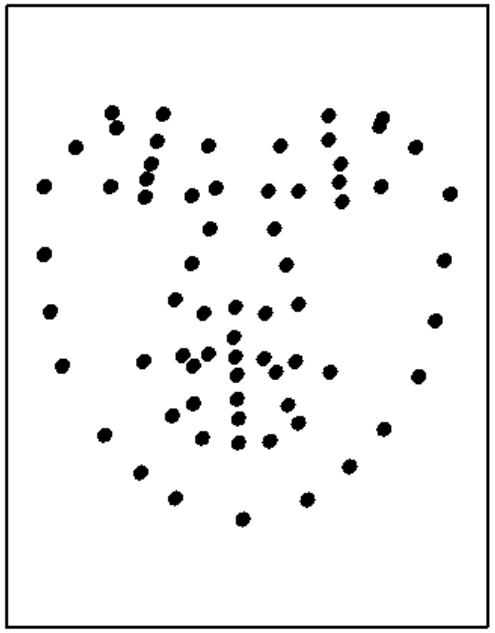


Figure 2: points

Figure 3: shape-free patch

- The combined shape and texture vector is used to build a matrix X of features for all images

- Principal Component Analysis (PCA) is used to summarise each face as a coordinate in a 200-d space
  - The data is mean-centred $\tilde{X} = X - \bar{X}$
  - The scaled covariance matrix is calculated $C = \tilde{X}\tilde{X}^T$
  - Eigendecomposition of the covariance matrix C gives

Real, symmetric C so can be diagonalised

$$C = P^T \Lambda P = \sum_{i=1}^{N} \lambda_i p_i p_i^T \approx \sum_{i=1}^{k<N} \lambda_i p_i p_i^T$$

  - Consider only the k=200 eigenvectors corresponding to the top 200 (sorted) eigenvalues stored as the columns of $P_{k:}$ principal components of a new coordinate system that explains most of the variation in the images

Projecting data onto PCs

- A facial feature vector x is transformed to a new point b in this 200d space by $b = P_k(x - \bar{x})$ and back again by $x = P_k{}^T b + \bar{x}$
- The vector b provides the coordinates of a face image in the 200d coordinate system and has compressed the data without losing too much
- The result is each facial image summarised as a vector b (of length k=200), based on PCA of the deviations from the mean shape and intensity image (b is the vector of face model parameters)

G. Edwards, A. Lanitis, C. Taylor, and T. Cootes, “Statistical Models of Face Images-Improving Specificity”, *Image and Vision Computing*, vol. 16, pp. 203-211, 1998.

Existing Method 1

- First age estimation algorithm **[Lanitis *et al*., 2002]**

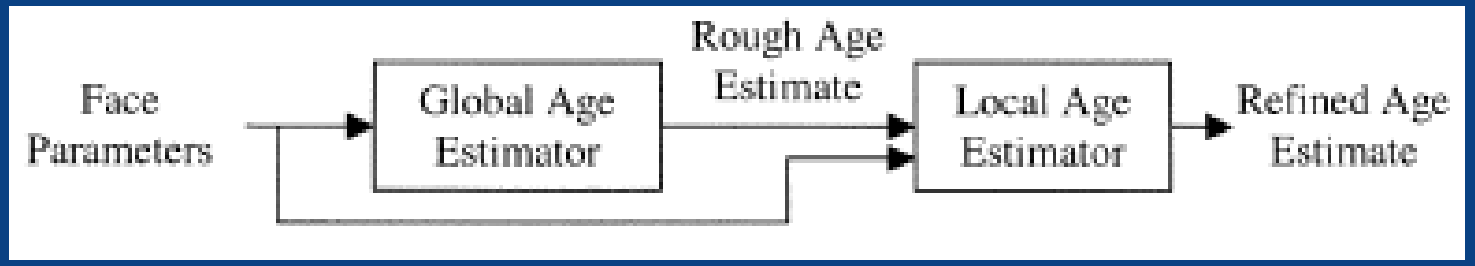
Aging function: ***age*** **= *f*(b)**,

- **b**: the vector of the face model parameters
- ***f***: a quadratic function.
- Training
  - ***f***() is fitted by Genetic Algorithm (GA) for each individual in the training set to determine his/her unique aging function
- Age estimation: Weighted Appearance Specific (WAS)
  - The aging function for an unseen face is calculated by the weighted sum of the known aging functions, where the weights are determined by the Mahalanobis distance from the test image to the training images

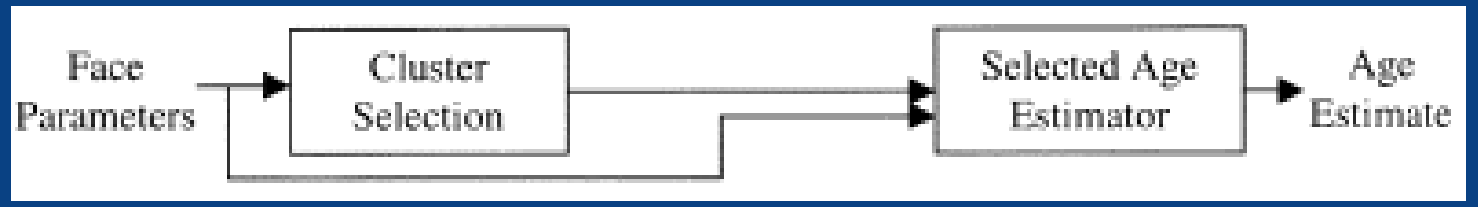
Considers correlations and is scale-invariant

Existing Method 2

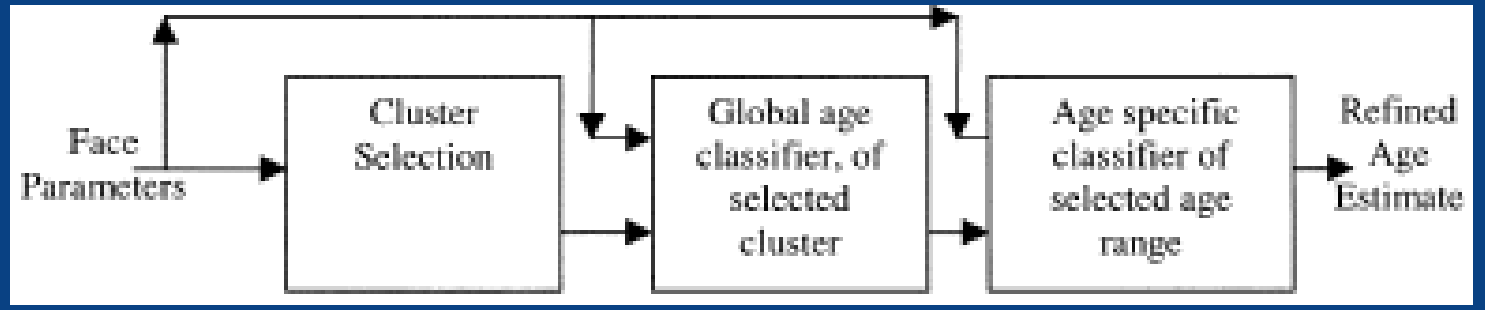
Hierarchical architectures **[Lanitis *et al*., 2004]**



Age Specific Age Estimation



Appearance Specific Age Estimation



Appearance and Age Specific (AAS) Age Estimation

Characteristics of facial aging

- The aging process is uncontrollable, slow and irreversible
- Collection of data is extremely laborious

Difficulty: *The available training data is a small set of highly incomplete aging patterns*

Characteristics of facial aging

- Different people age in different ways
- Determined by genetics as well as many external (lifestyle) factors

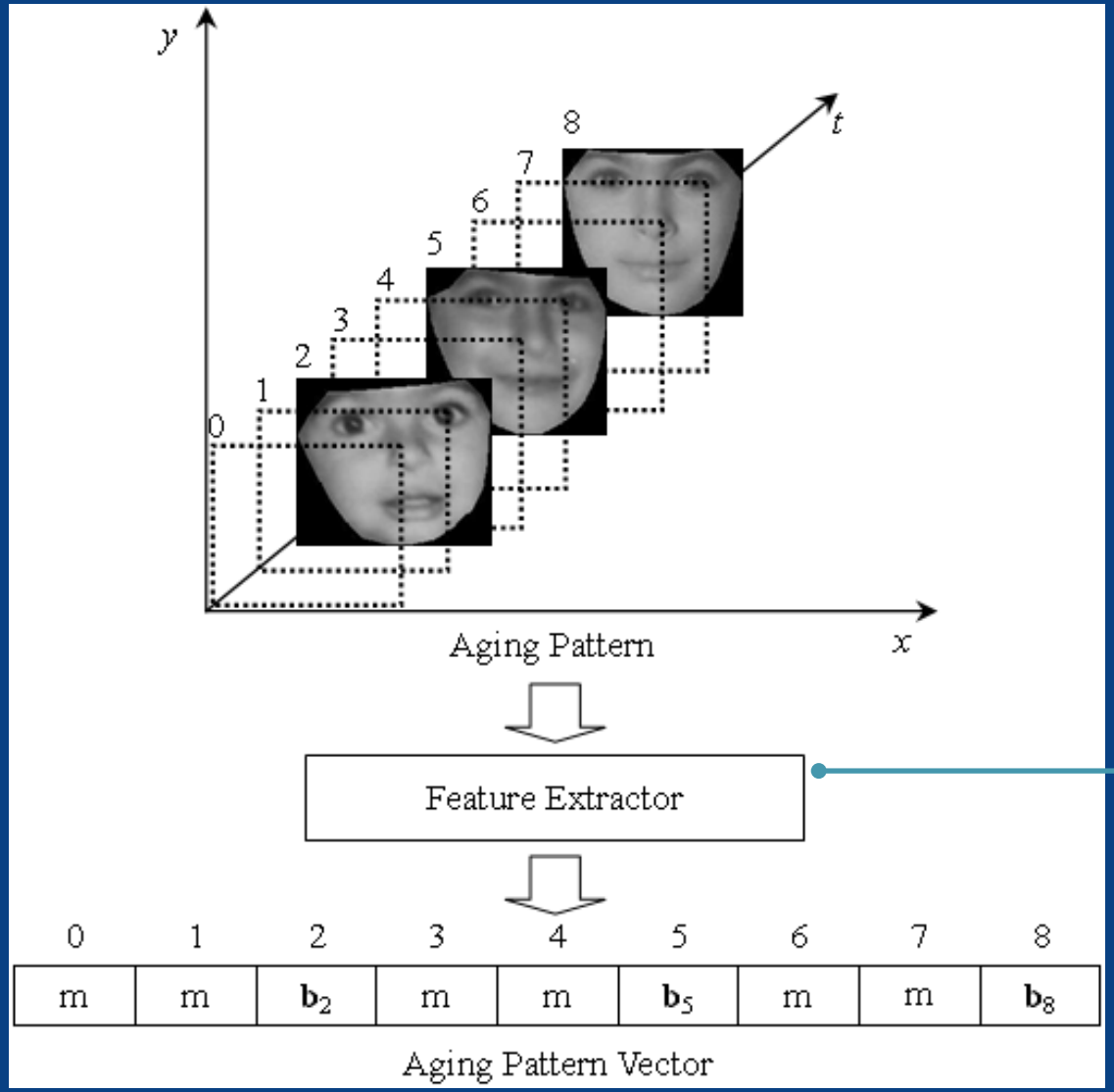
Difficulty: *The mapping from features (face images) to class labels (ages) is not unique*

Characteristics of facial aging

- The aging patterns are temporal data
- Each age has a unique rank in the time series.

Difficulty: *The set of class labels (ages) is a totally ordered set.*

Our method (2007)



We model the aging process, not just the age of each image

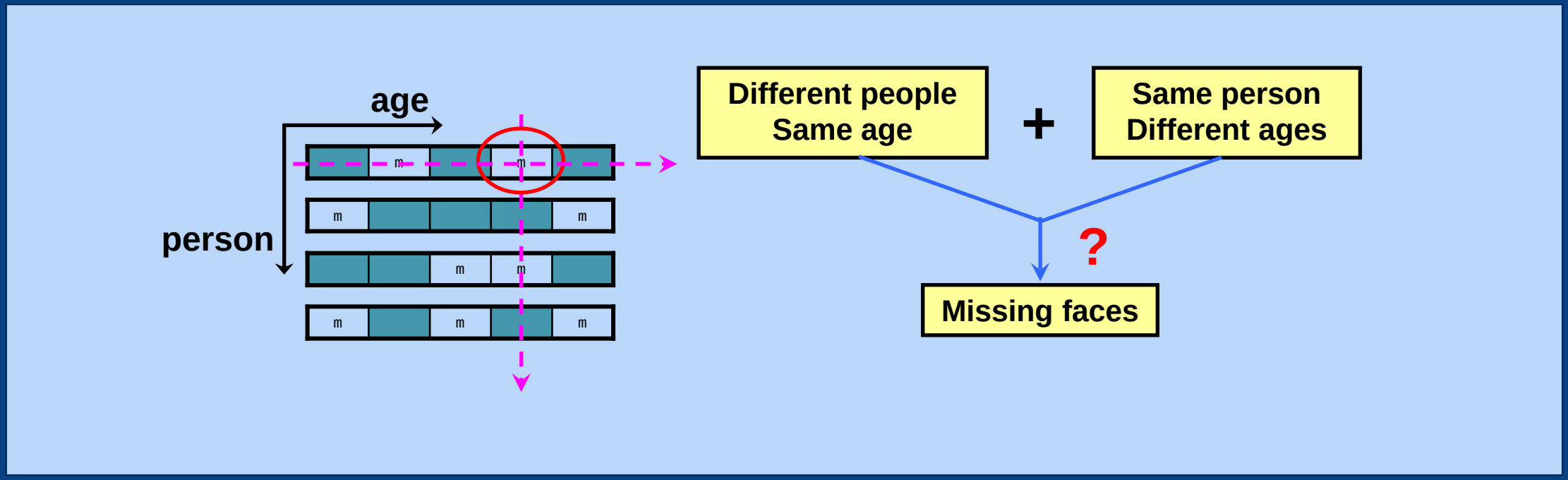
Appearance Model

[Edwards *et al*. IVC, 1998]

An aging pattern is a sequence of <u>personal</u> face images sorted in <u>time order</u>

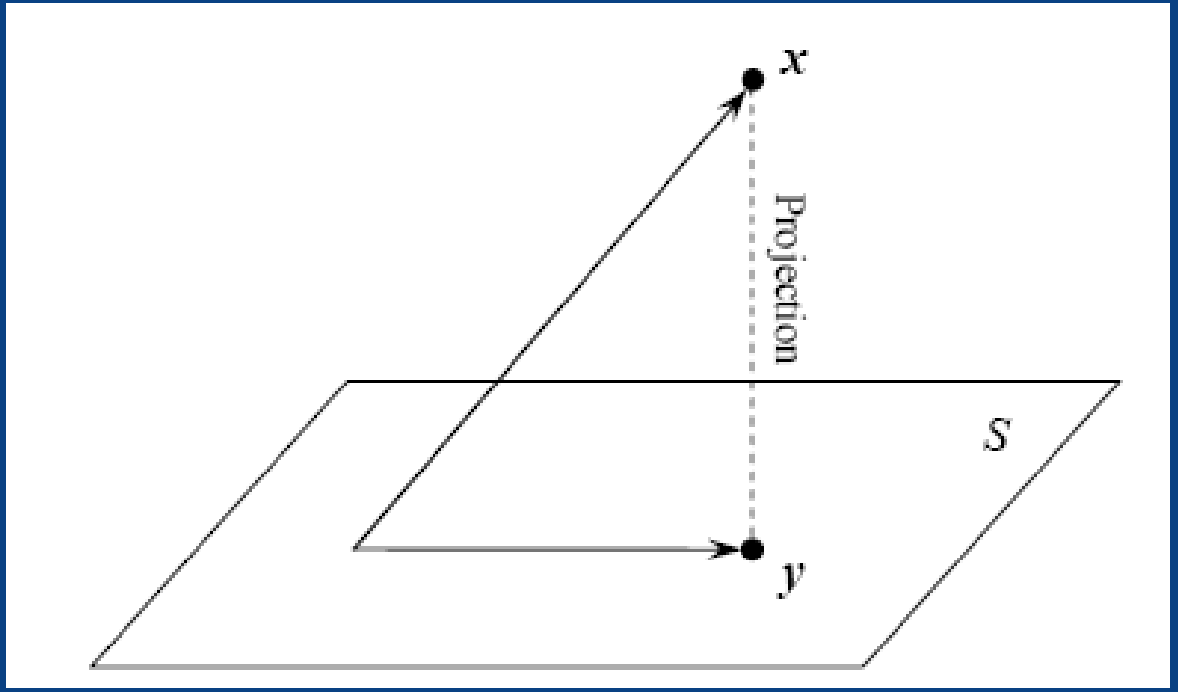
Handling missing data

- **The goal**
  - Find a representative subspace for the aging patterns that can handle the missing data problem
- **Basic idea of data imputation**



Our "AGES" Algorithm

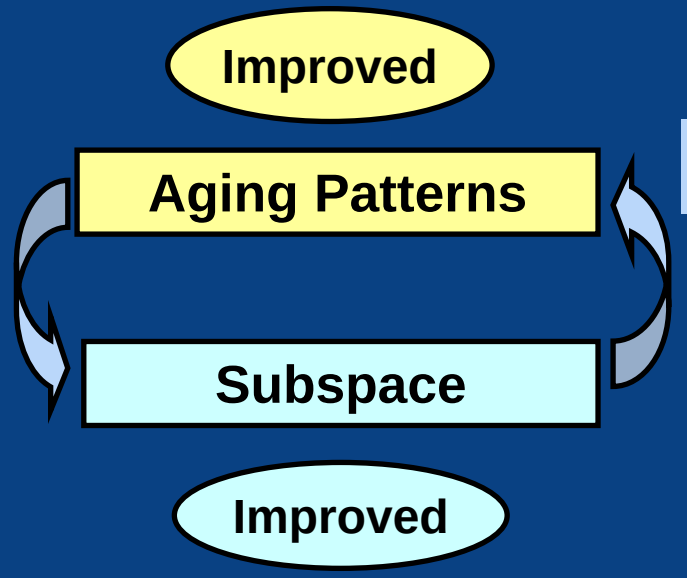
- AGES = AGing pattErn Subspace
- We learn a representative subspace for true aging patterns based on training examples



AGES – Learning

- Apply Principal Component Analysis (PCA) iteratively

Initialise missing values to mean (rough start)



$$y(t) = P_k(t)(x - \bar{x}(t))$$

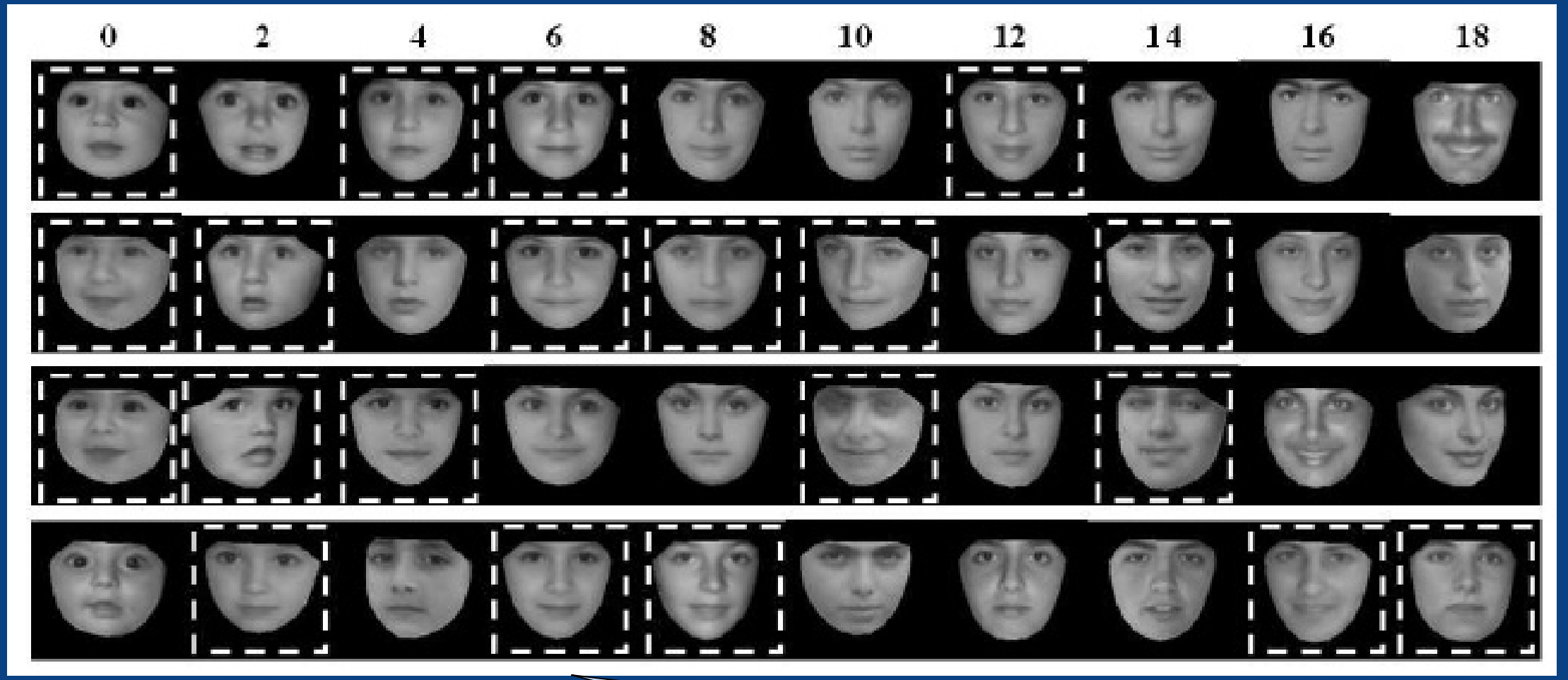
Project data onto subspace defined by PCs

$$\hat{x}(t) = P_k{}^T(t)y(t) + \bar{x}(t)$$

Project transformed data back.
Is it accurate?
Update missing values to mean of reconstructed values

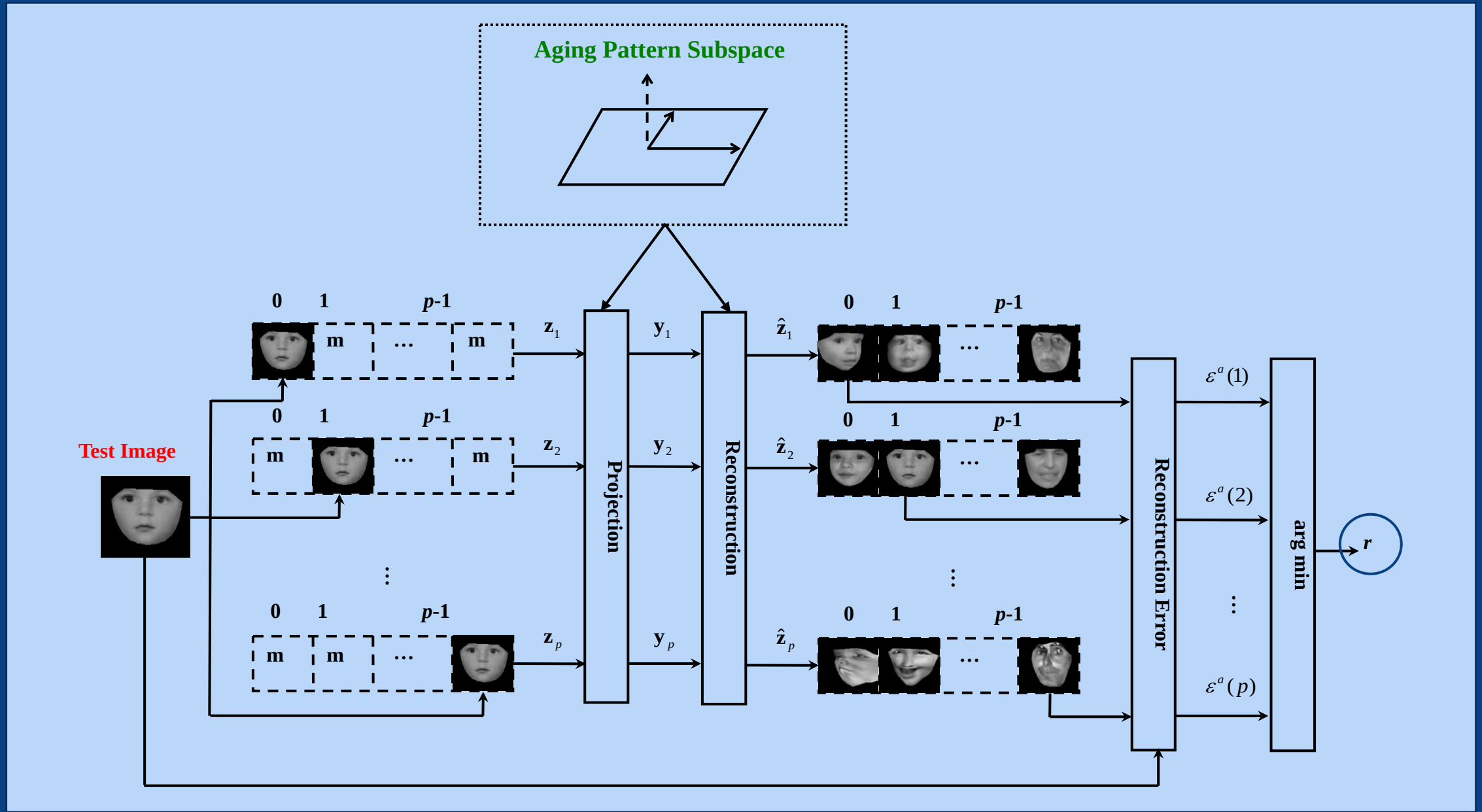
Repeat until the reconstruction error is lower than a certain threshold, which means we have a representative subspace of the aging patterns (see paper for proof of convergence)

AGES – Learning how the face ages



dotted outline images were missing and have been learned by our AGES algorithm

AGES – Age Estimation for a single image



Experiments - data

Data set: FG-NET Aging Database

- 1002 face images
- 82 subjects
- 6-18 images each subject
- Age: 0-69
- Variations: illumination, pose, expression, beards, moustaches, spectacles, hats

200 features (principal components) used to represent 95% of the variation in the data

Experiments - data

Data set: MORPH database

- 1,724 face images
- 515 subjects
- Around 3 images per subject
- Age: 15-68
- Variations: illumination, pose, expression, beards, moustaches, spectacles, hats
- Only 433 images from Caucasian descent are used as a test set

Experiments – Compared methods

- AGES / AGES$_{lda}$ [Geng, Zhou and Smith-Miles, 2007]
- WAS, [Lanitis *et al*., 2002]
- AAS, [Lanitis *et al*., 2004]
- Conventional machine learning classification methods
  *k*NN, BP, C4.5, SVM
- 29 human observers
  –HumanA test
  –HumanB test

Accuracy of Age Estimation

Our method has smallest average errors… yay!

$$MAE = \sum_{k=1}^{M} |\widehat{age}_k - age_k|/M.$$

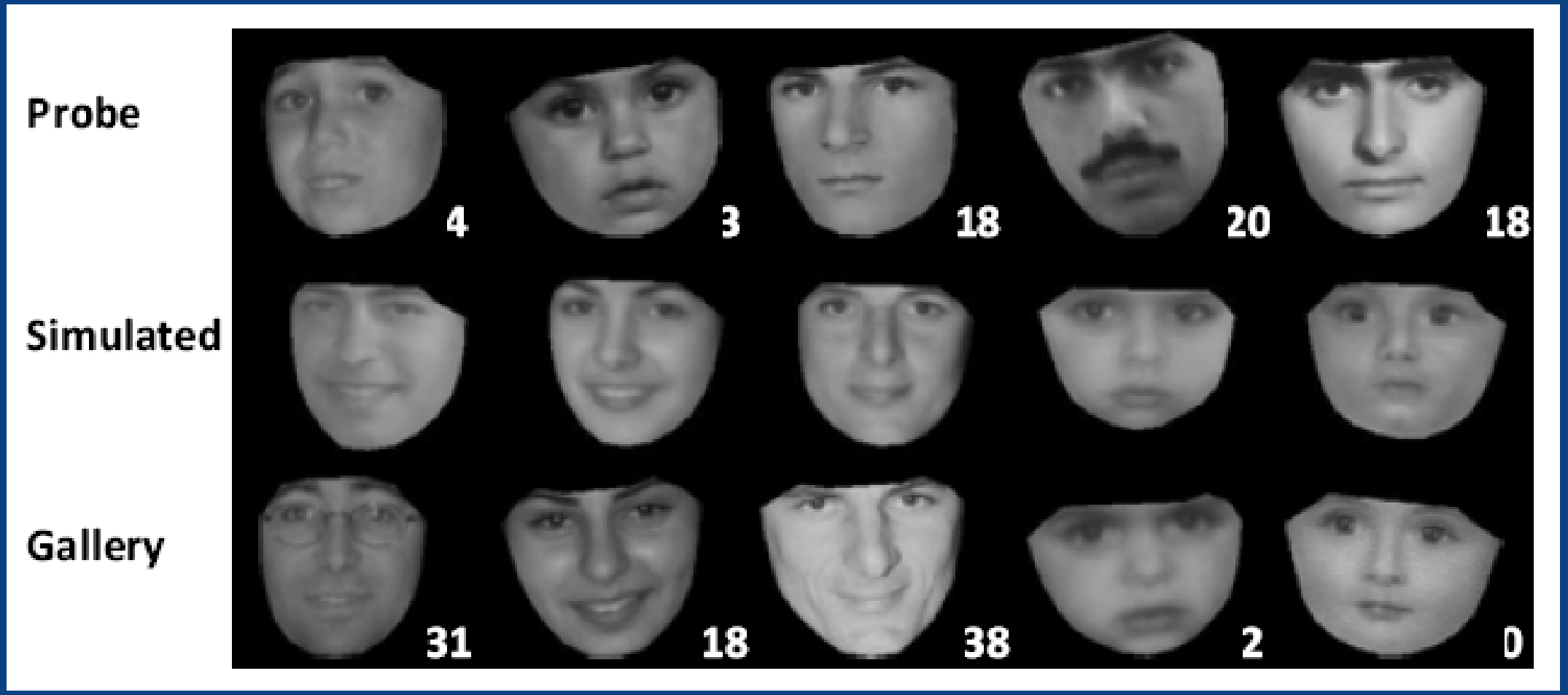
It's even more accurate than humans guessing!

MAE OF STANDARD AGE ESTIMATION ON FG-NET AND MORPH

Method	AGES	AGES$_{lda}$	WAS	AAS	kNN	BP	C4.5	SVM	HumanA	HumanB
FG-NET (LOPO)	**6.77**	**6.22**	**8.06** (1, 1)	14.83 (1, 1)	8.24 (1, 1)	11.85 (1, 1)	9.34 (1, 1)	**7.25** (1, 1)	8.13 (1, 1)	6.23 (-1, 0)
MORPH (Test Set)	8.83	**8.07**	9.32 (1, 1)	20.93 (1, 1)	11.30 (1, 1)	13.84 (1, 1)	12.69 (1, 1)	9.23 (1, 1)	— —	— —

See **Geng, X., Zhou, Z.-H., and Smith-Miles, K. A., "Automatic Age Estimation Based on Facial Aging Patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 29, no. 12, pp. 2234-2240, 2007.**

Results – Aging Effects Simulation



Conclusions

- AGES performs remarkably better than the state-of-the-art algorithms. It is even better than humans at guessing ages.
- Nice application of some fairly simple maths
- More sophisticated maths helps improve things further:

- Tensor analysis (multilinear subspace)
- Fusion with other biometrics
- Alternatives to PCA for imputing missing values

MAE down to 5.36 years

MAE down to 5.14 years by fusing face and voice data

Postscript … a decade later

- Lots of media coverage

  e.g. Cambridge University, The Naked Scientist radio show in 2008

https://www.thenakedscientists.com/articles/interviews/guess-your-age

- Lots of citations in the academic literature (>1000)

  so we knew research was advancing while we moved away …

Fast forward to late 2018 …

Biometric Mirror highlights flaws in artificial intelligence

Share

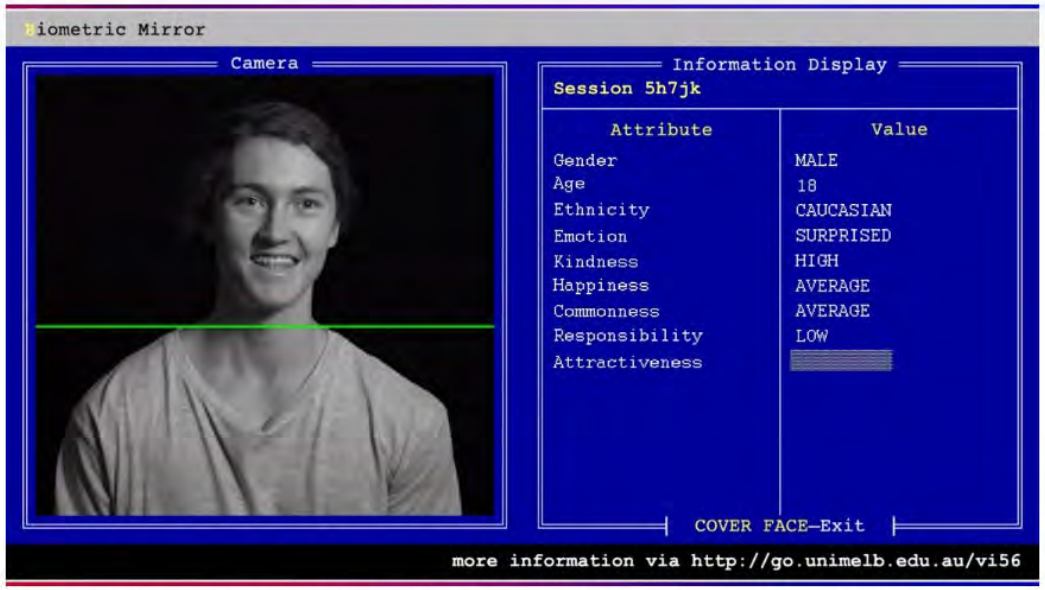
Like 7

Tweet

Share

0

G+



Biometric Mirror uses an open dataset of thousands of facial images and crowd-sourced evaluations. Picture: Sarah Fisher/University of Melbourne

In a world-first, University of Melbourne researchers have designed an artificial intelligence (AI) system to detect and display people's personality traits and physical attractiveness based solely on a photo of their face.

The system, called Biometric Mirror, investigates a person's understanding of AI and their response to the information about their unique traits.

More Information
Holly Bennett
+61 3 8344 7758
0466 514 367
holly.bennett@unimelb.edu.au

Department: Media

When someone stands in front of Biometric Mirror, the system detects a range of facial characteristics in seconds. It then compares the user's data to that of thousands of facial photos, which were evaluated for their psychometrics by a group of crowd-sourced responders.

"Our study aims to provoke challenging questions about the boundaries of AI. It shows users how easy it is to implement AI that discriminates in unethical or problematic ways which could have societal consequences. By encouraging debate on privacy and mass-surveillance, we hope to contribute to a better understanding of the ethics behind AI."

Curiosity killed the cat …

Before… au natural

45 year old
Female
Introvert
Kind
Happy
Irresponsible!

Marquardt Mask

After … mathematical perfection?

Same age
Less happy and less kind
Less aggressive
Extrovert
More responsible (40%)
Weird (100%)

In AI we trust …

- How do we establish trust in an algorithm?
  - Select some test instances,
  - Report statistics such as the mean and standard deviation of performance metric
  - If acceptable, and better (on average) than competing algorithms, then the conclusion is typically drawn that the algorithm is successful.
- Choice of test instances?
- Averages hide all sorts of sins …

Revisiting our 2007 study via Instance Space Analysis

- Instances: 1002 images sourced from the FG-NET database;
- Features: top 10 principal components (PCs) of the appearance model
- Algorithms: top 3 algorithms based on average absolute error in years

Our algorithm AGES-LDA (2007)	AAS (2004)	WAS (2002)	SVM	Neural Network	Decision Tree	kNN	Humans Observers
5.54	14.83	8.06	7.25	12.11	9.34	8.24	8.13

Measuring "good" algorithm performance

- Performance metric: scaled relative error in predicted age, compared to the known true age

$Y_{i,j} = \frac{|\hat{a}_{i,j} - a_i|}{\sum_{\alpha \in A} |\hat{a}_{i,\alpha} - a_i| / |A|}$ for instance $i \in I$ with true age a_i and algorithm $j \in A$, with estimated age $\hat{a}_{i,j}$.

  (compares the error of one algorithm relative to the average error of all algorithms)

- User-definition of good performance: we arbitrarily define "good" as $Y_{i,j} < 0.5$
- absolute error in age prediction is less than half the average error across all considered algorithms

Creating an instance space

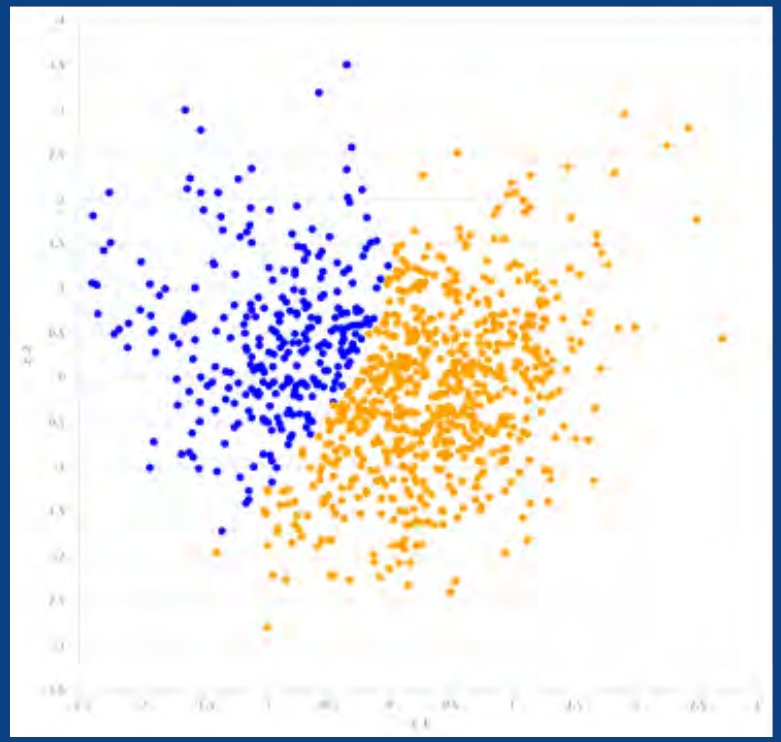
- Each instance is represented as a point in the 10D feature space
- Then projected to a 2D instance space using a novel dimensionality reduction method
  - includes a feature subset selection stage which chose 6 most informative features
- Optimal projection matrix to maximise linear trends

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.2525 & -0.3416 \\ 0.0544 & 0.347 \\ 0.7469 & 0.321 \\ -0.1919 & 0.5446 \\ -0.1147 & 0.2237 \\ -0.0656 & 0.4471 \end{bmatrix}^T \begin{bmatrix} PC3 \\ PC4 \\ PC5 \\ PC6 \\ PC7 \\ PC8 \end{bmatrix}$$

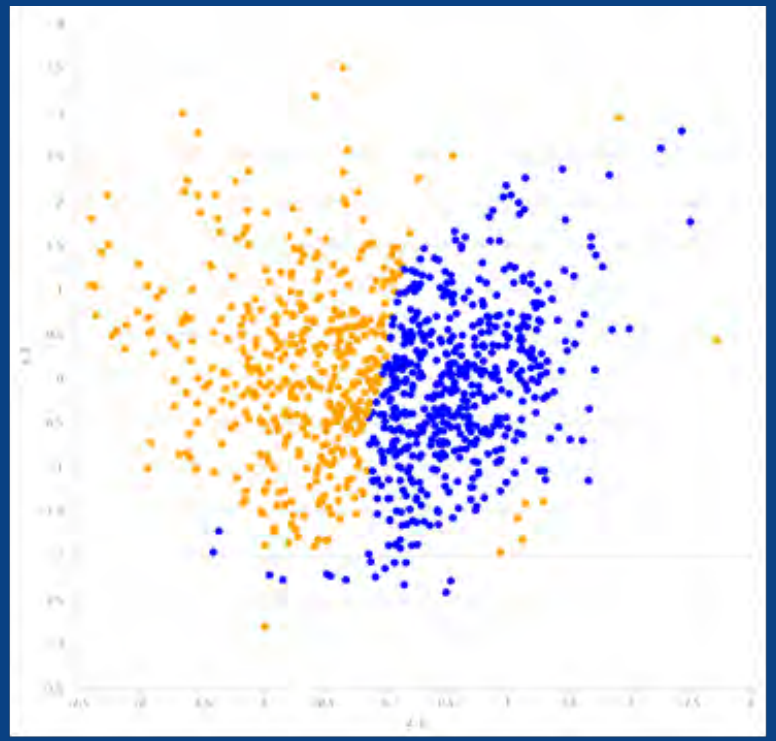
SVM Predictions of Good Performance

Blue = good; Orange = not good (bad?)

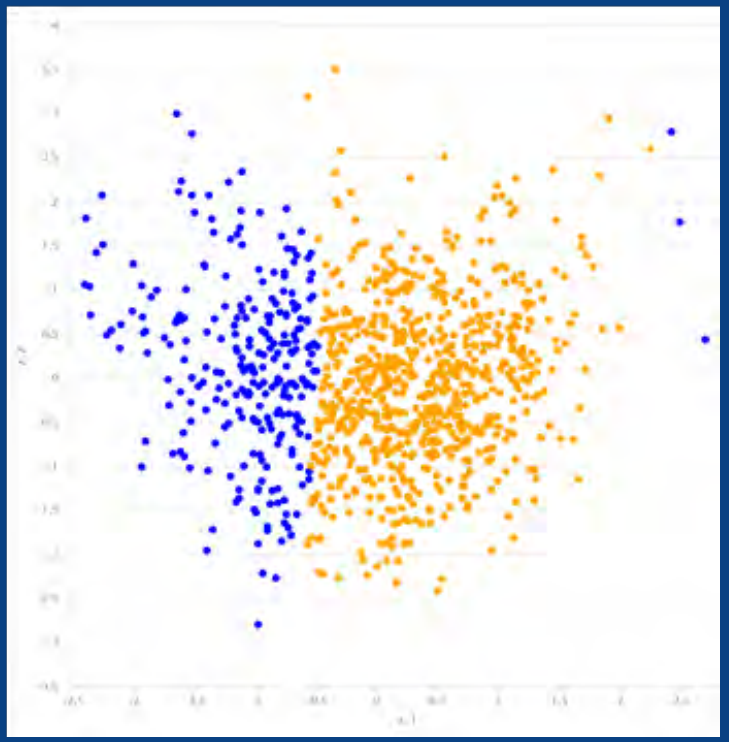
Each instance (facial image) is represented as a point in the 2D coordinate system defined by the optimal projection matrix.



our 2007 algorithm



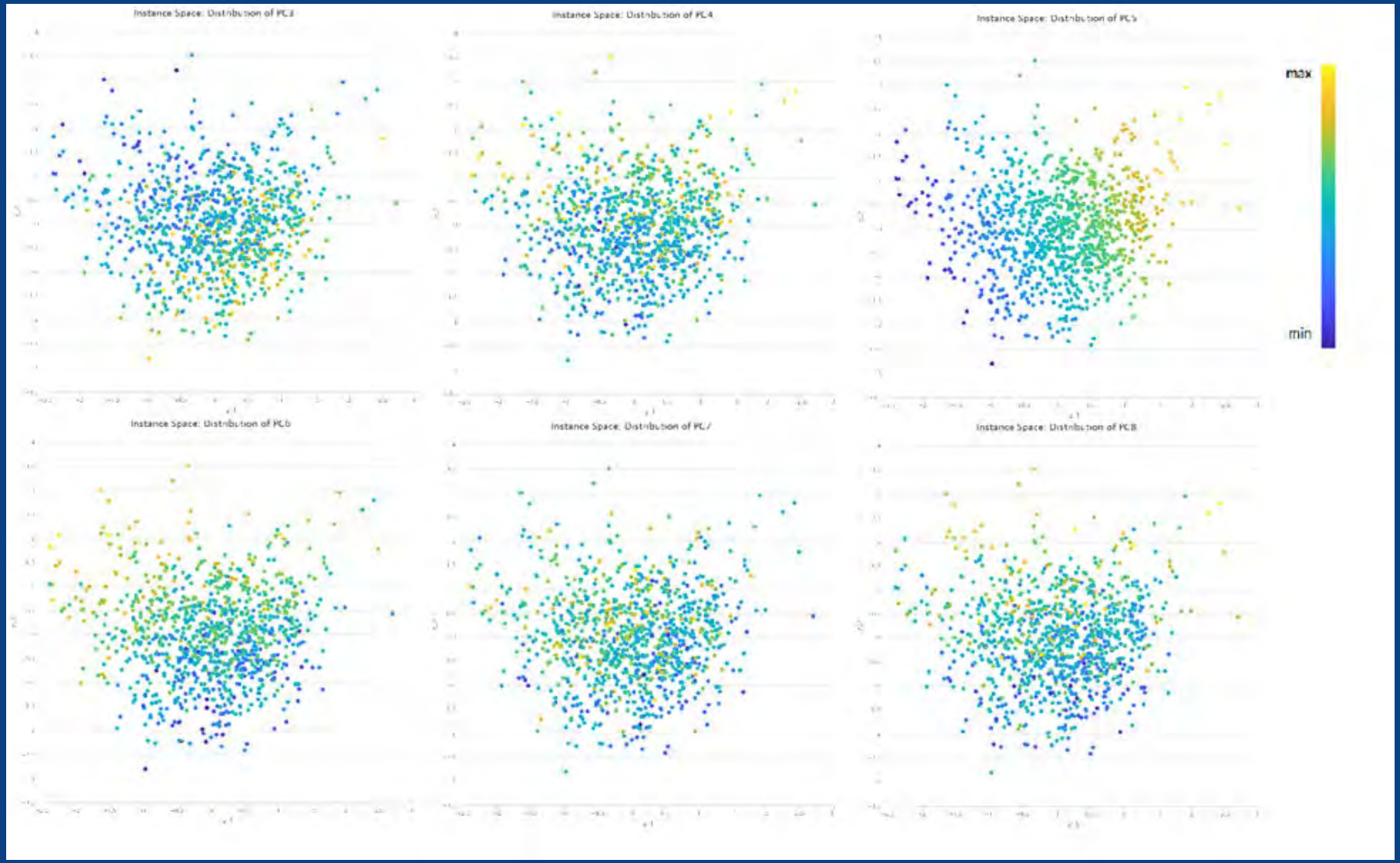
2002 WAS algorithm



SVM

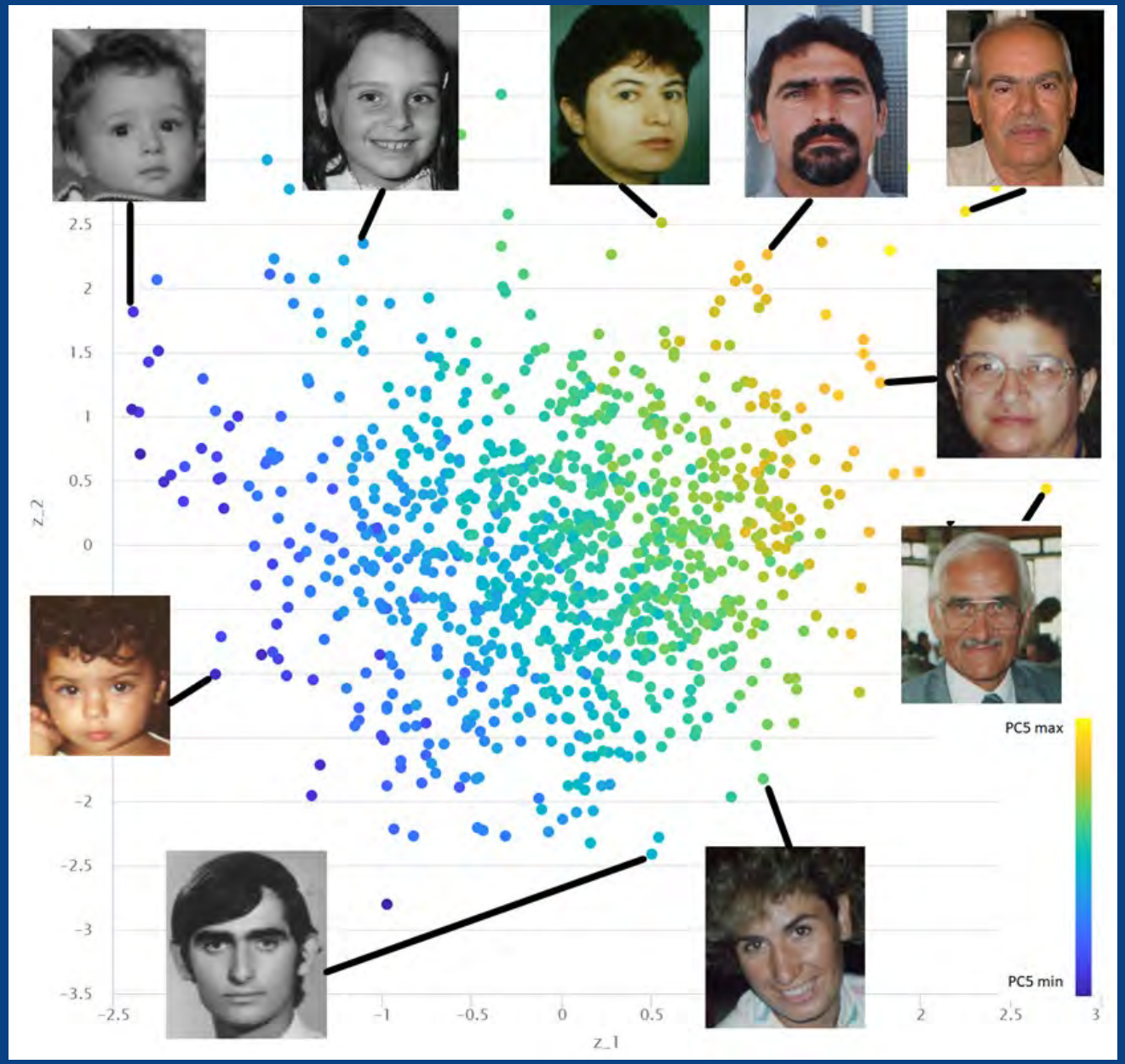
Distribution of Features

PC3 PC4 PC5



PC6 PC7 PC8

PC5 correlates with true age ...



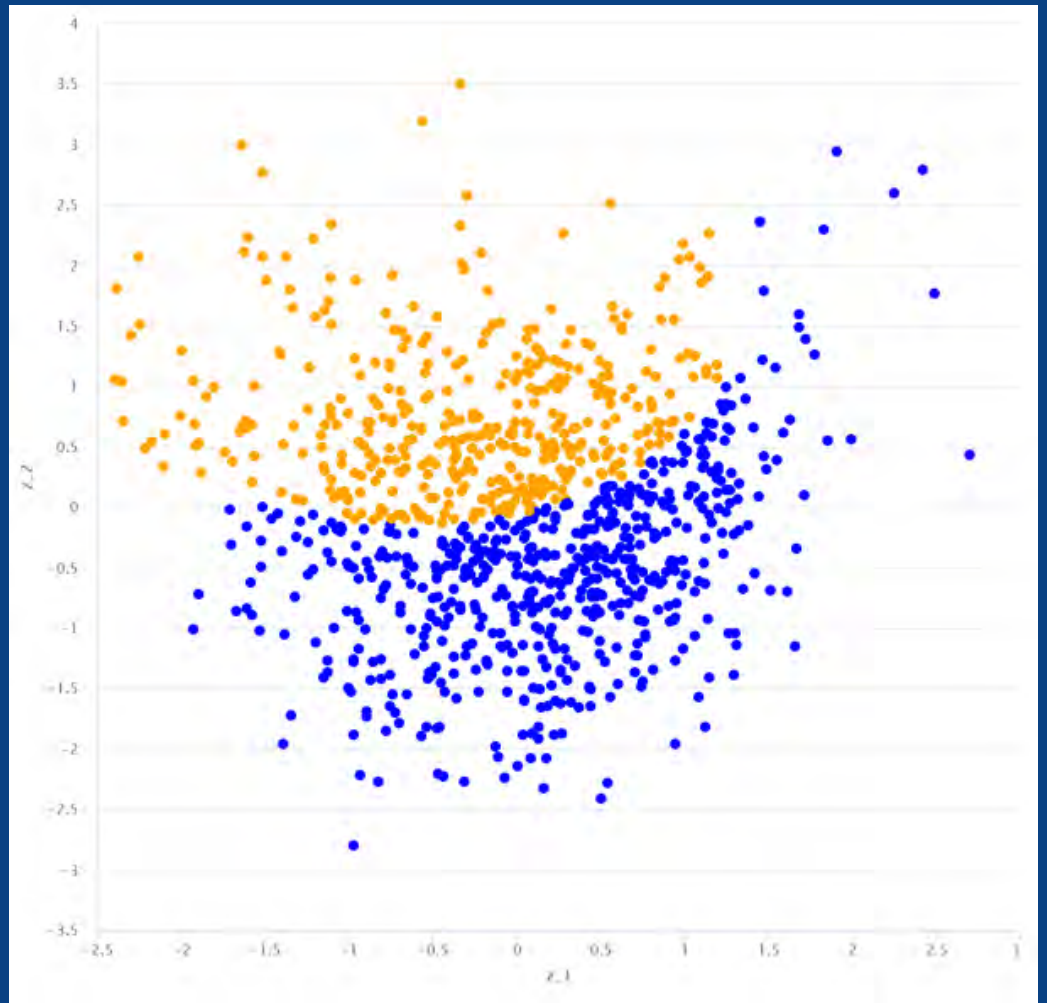
Insight → New Algorithm Idea

- We propose a new algorithm for age estimation based on an instance's PC5 value
- We fit a simple power law function to estimate age as:
  - $\hat{a}_{i,PC5} = 25.084(1 + PC5_i)^{1.2032} - 10$
  - where this new algorithm is called PC5, and estimates the age of facial image instance i using its 5th principal component.
- This PC5 algorithm is competitive with the others
- Absolute error on average across the FG-NET database of 7.36 years

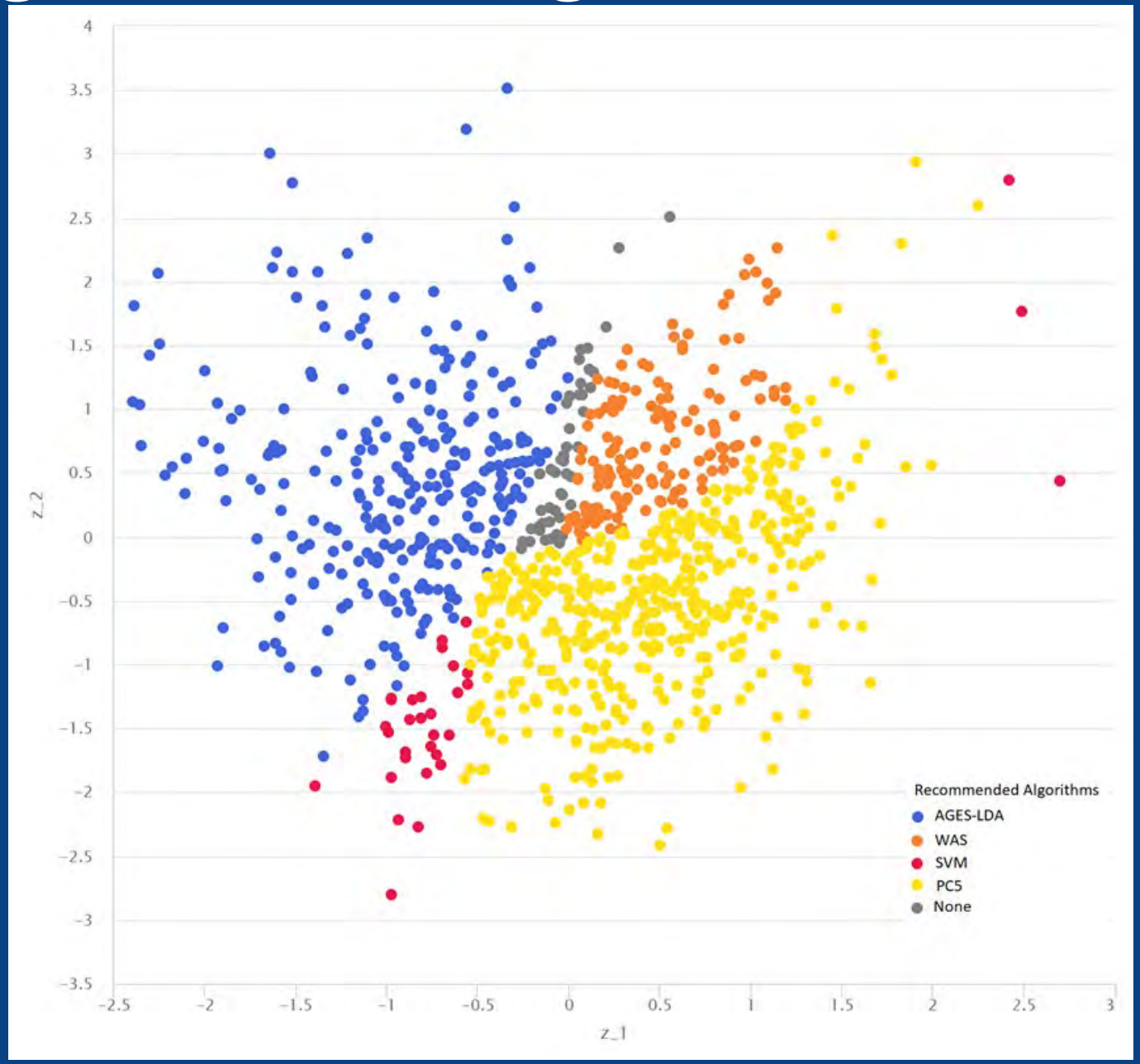
New PC5 algorithm's unique footprint

Not age-dependent like the others

Good across many ages, in lower half (↑ PC3 and ↓ PC6)



Combining SVMs for Algorithm Selection



Were our 2007 conclusions upheld?

- Yes, our AGES algorithm is best on average, but … only because FG-NET is biased towards young people
- Our AGES algorithm has strength on young faces, and weaknesses on older faces
- WAS is better for older faces
- SVM is similar to AGES, but also best for very old faces
- Insights offered by instance space analysis helped create a new algorithm with unique strengths
- The limitations of FG-NET, and the need for caution when interpreting “on average” results, have been exposed!

Lecture 1 (Part 1)

- Introduction to Instance Space Analysis
- Introduction to MATILDA
- Case Study: Optimisation (University Timetabling)

Lecture 2

- Case Study: Computer Vision (Facial Age Estimation)

Lecture 3

- Evolving new instances to fill an instance space (Black-box optimisation and new artwork!)

THE UNIVERSITY OF MELBOURNE

Part 2

How to perform an Instance Space Analysis

Mario Andrés Muñoz : munoz.m@unimelb.edu.au

https://matilda.unimelb.edu.au/matilda/matildadata/tutorials/matilda-technical-details.mp4

- Using the MATLAB code/live script
- Using MATILDA's web Interface
- Case Study: Machine Learning (Classification)

THE UNIVERSITY OF MELBOURNE

QUESTIONS?

MATILDA:

https://matilda.unimelb.edu.au

MATLAB code:

https://github.com/andremun/InstanceSpace

Please contact us for any enquiries or support:

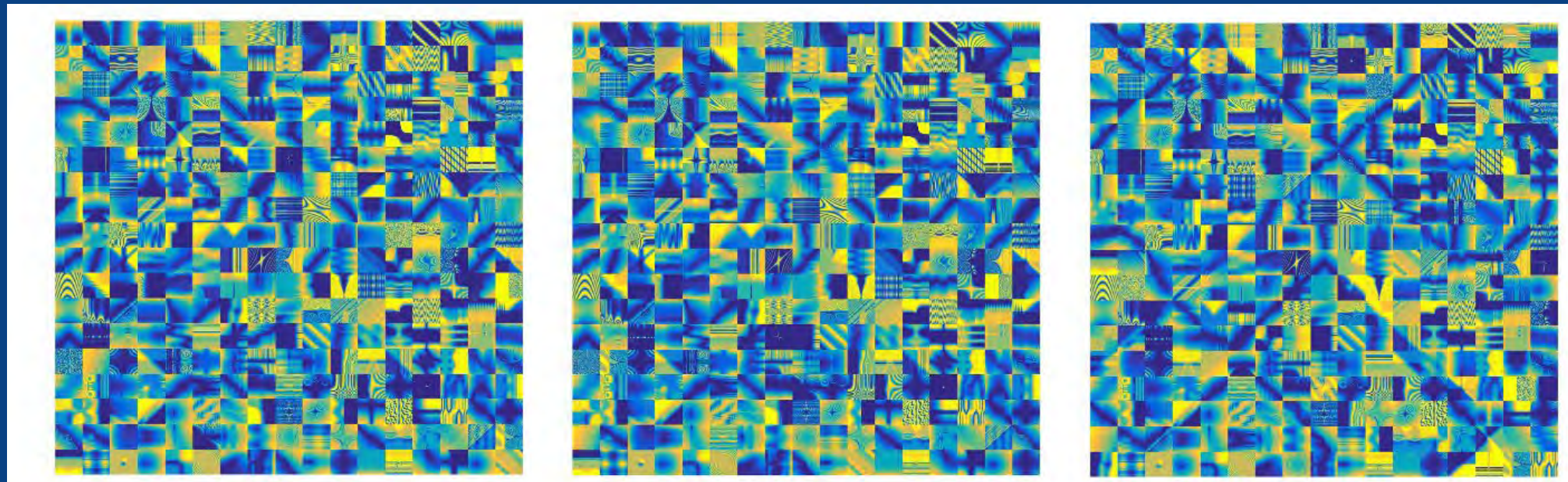
matilda-team@unimelb.edu.au

smith-miles@unimelb.edu.au

THE UNIVERSITY OF MELBOURNE

Lecture 3

When mathematics becomes art ... the unexpected beauty of self-evolving mathematical functions



Negentropy Triptych (2019) by Kate Smith-Miles and Mario Andres Munoz Acosta

Lecture 1 (Part 1)

- Introduction to Instance Space Analysis
- Introduction to MATILDA
- Case Study: Optimisation (University Timetabling)

Lecture 2

- Case Study: Computer Vision (Facial Age Estimation)

Lecture 3

- Evolving new instances to fill an instance space (Black-box optimisation and new artwork!)

THE UNIVERSITY OF MELBOURNE

Part 2

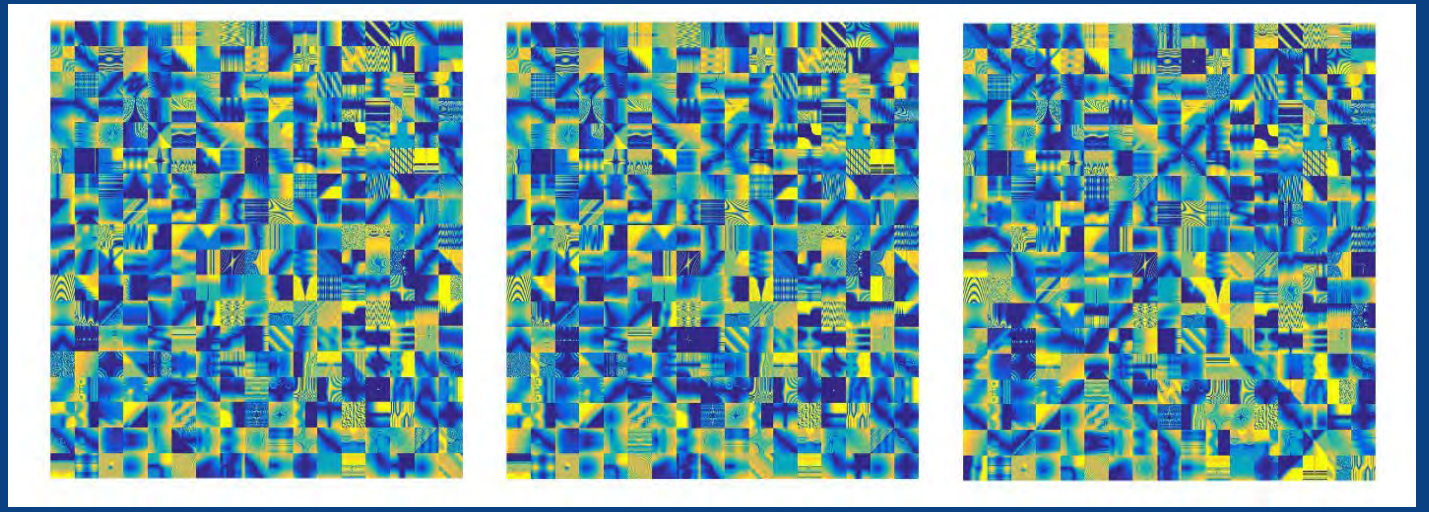
How to perform an Instance Space Analysis

Mario Andrés Muñoz : munoz.m@unimelb.edu.au

https://matilda.unimelb.edu.au/matilda/matildadata/tutorials/matilda-technical-details.mp4

- Using the MATLAB code/live script
- Using MATILDA's web Interface
- Case Study: Machine Learning (Classification)

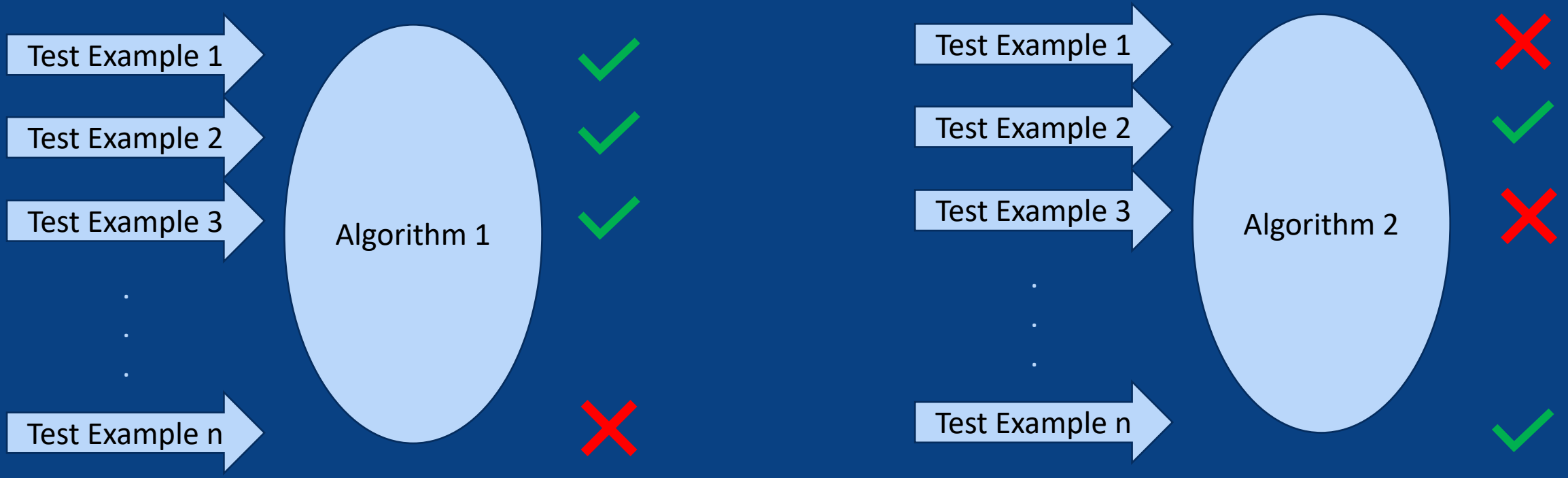
Lecture 3 Outline



- Scientific motivation
  - Aims
  - Method
  - Results
- Artistic motivation
  - A whirlwind tour of the history, philosophy, and science of aesthetics
  - Mathematical beauty
- The making of “Negentropy Triptych”
  - Artistic and Scientific (Algorithmic) Results

Scientific Motivation

- Algorithms guide machines to solve problems
- How do we know if an algorithm can be trusted?



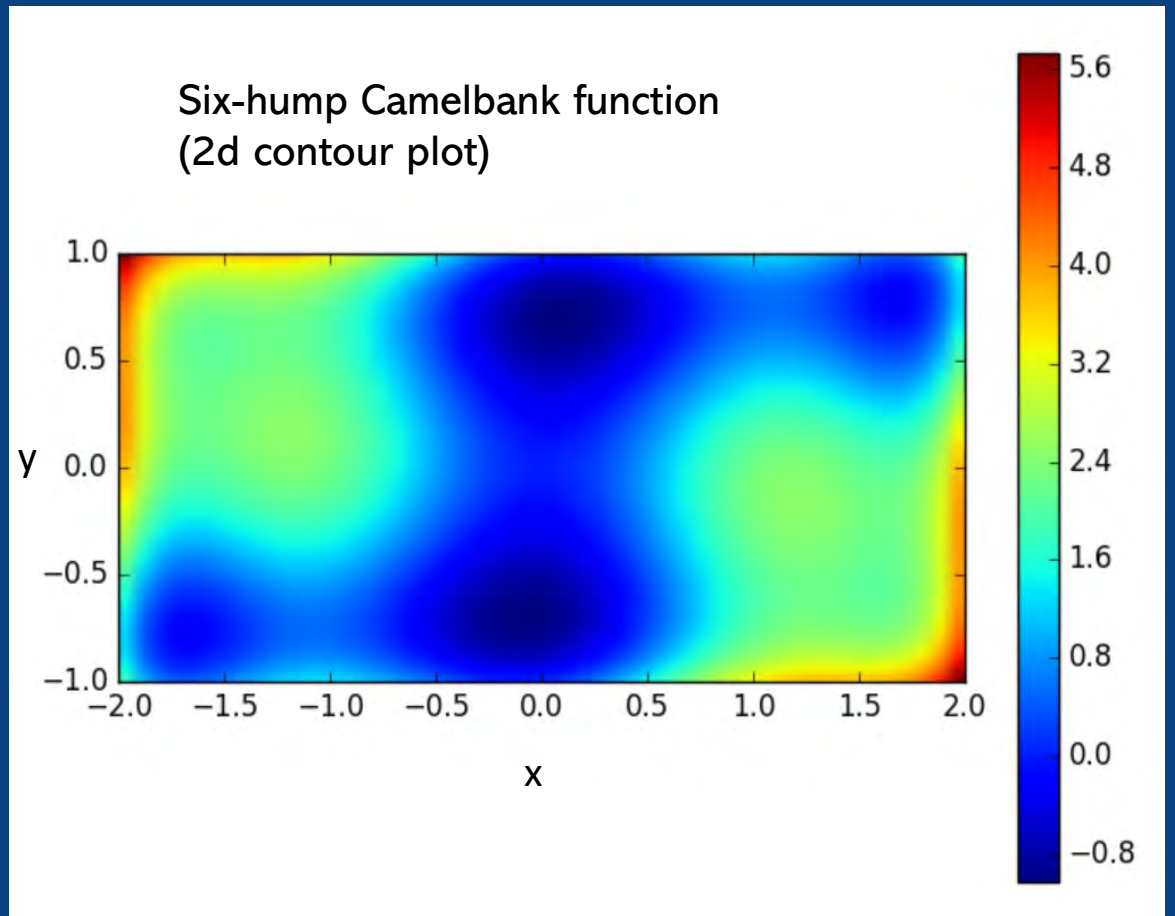
- Exposing algorithm weaknesses depends on test example diversity

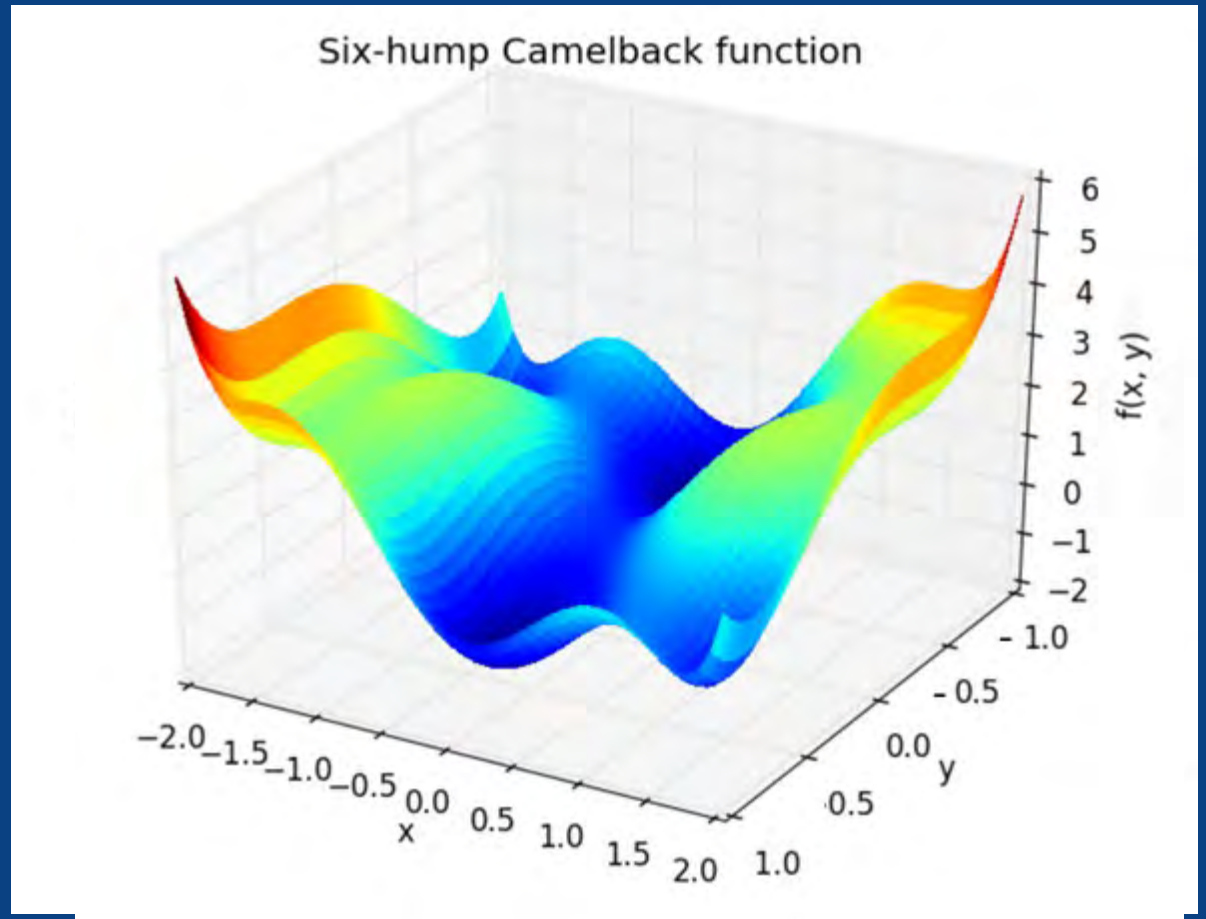
Which examples reveal an algorithm's weakness?

- Randomly chosen examples may not be difficult for an algorithm
- Can computers search for examples that an algorithm can't solve?

Algorithms for Solving Optimisation Problems

- What is an optimisation problem? Minimising a function (unconstrained)
  - Find the value of x to minimize f(x) … watch out for traps (local minima)
  - Find the values of (x,y) to minimize f(x,y)

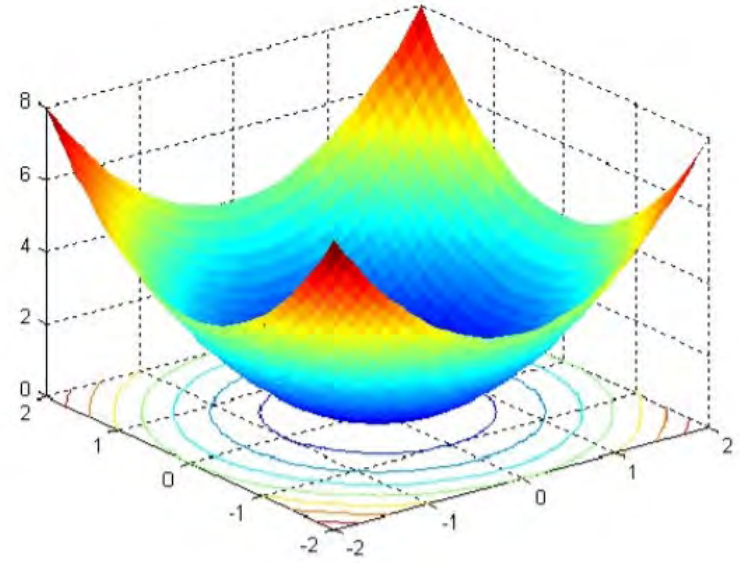




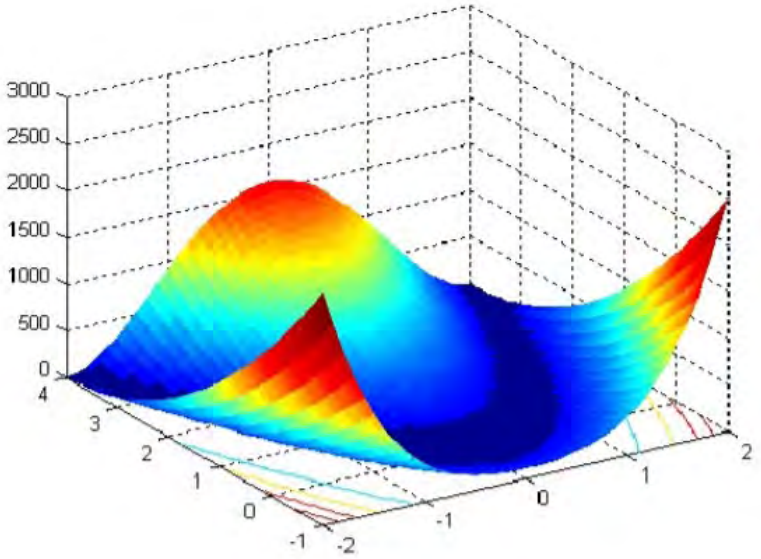
Easy for all algorithms … single optima

Easy for most algorithms … plateau trap

Well known 3d test examples

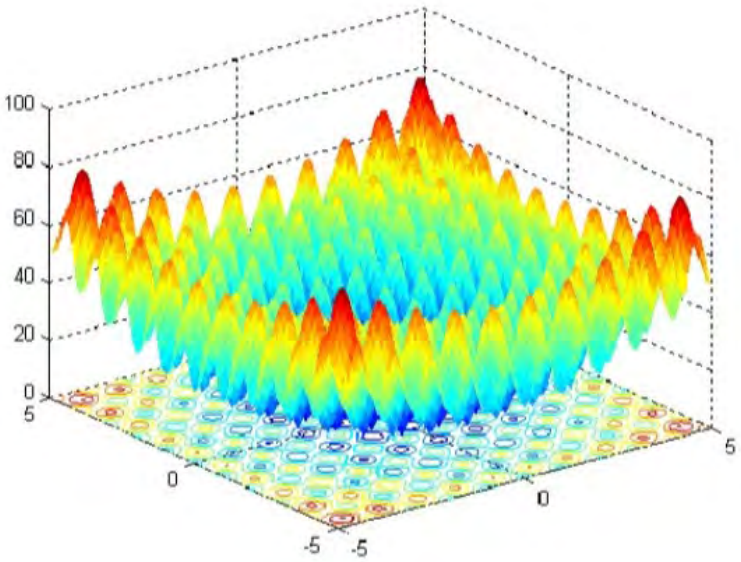


(a)

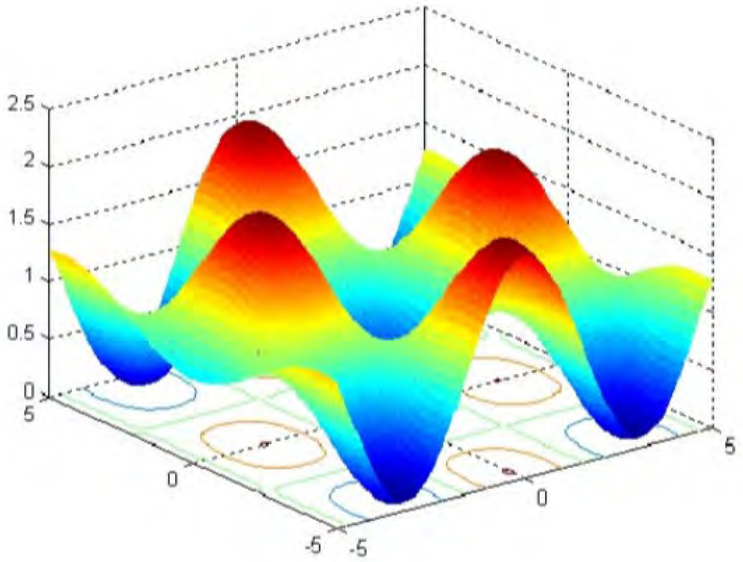


(b)

Hard for most algorithms … many minima



(c)



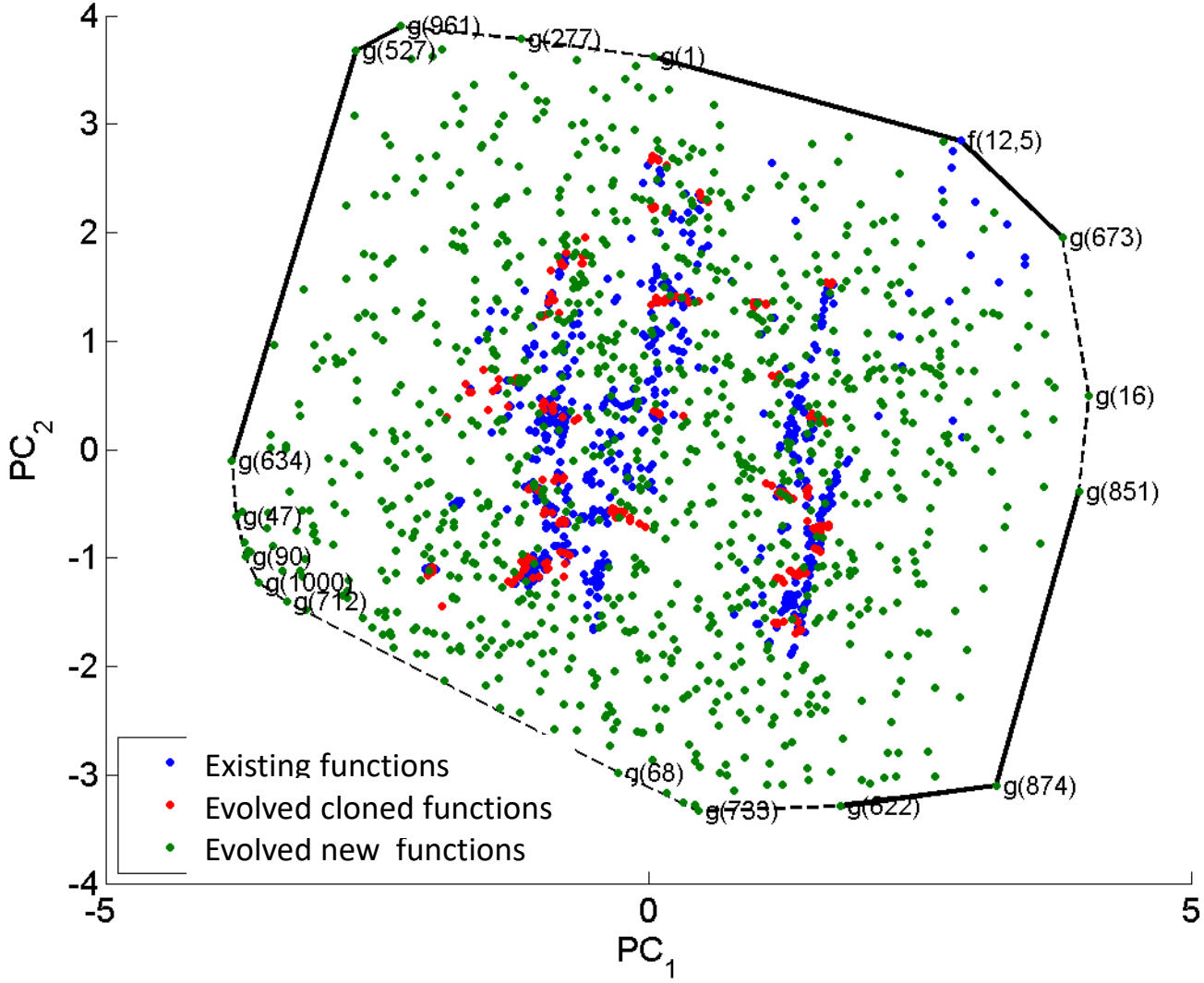
(d)

Hard for most algorithms … doesn't know what it doesn't know

What makes optimisation problems difficult?

- Characteristics such as “ruggedness”, flat plateaus, dependency between variables, etc.
- How can we see difficulty of a problem if more than 3d?
- An “instance space” projects all functions as points in a 2d plane, based on their similarities and differences in these characteristics
  - We estimate 8 characteristics based on a sample of points (statistics)
  - Each function is summarizing by unique 8d “feature vector”
  - Each feature vector is projected from 8d to 2d (linear algebra)
- Algorithm difficulty can be visualized across 2d instance space
  - Which algorithm is best for which types of functions, and why?

An instance space for optimisation problems



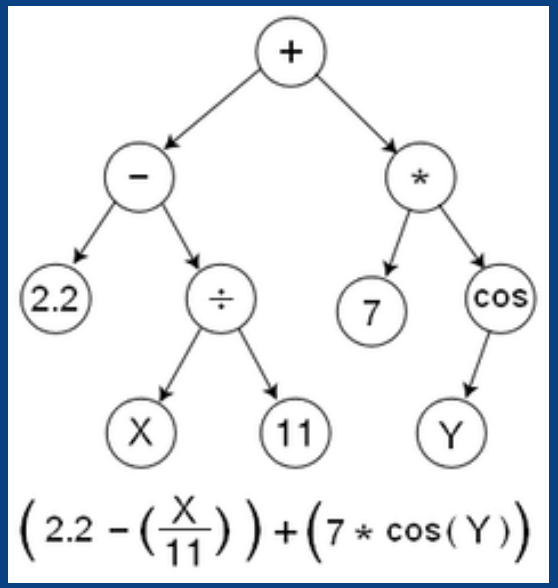
Evolving functions?

- Darwin's Natural Selection and Survival of the Fittest

  - Individuals are animal species (parents)
    → reproduce to create offspring, combining traits of both parents, with some random mutations
  - Only the fittest offspring survive
    → become parents
  - Average fitness of the population increases over time

Evolving functions by Genetic Programming

- Inspired by evolution
  - Individuals are tree structures defining a mathematical function f(x,y)
  - Parents create children by combining branches, with random mutations, using a vocabulary of:
    - Arithmetic operators
    - Trigonometric operators
    - Exponentials
    - Absolute values

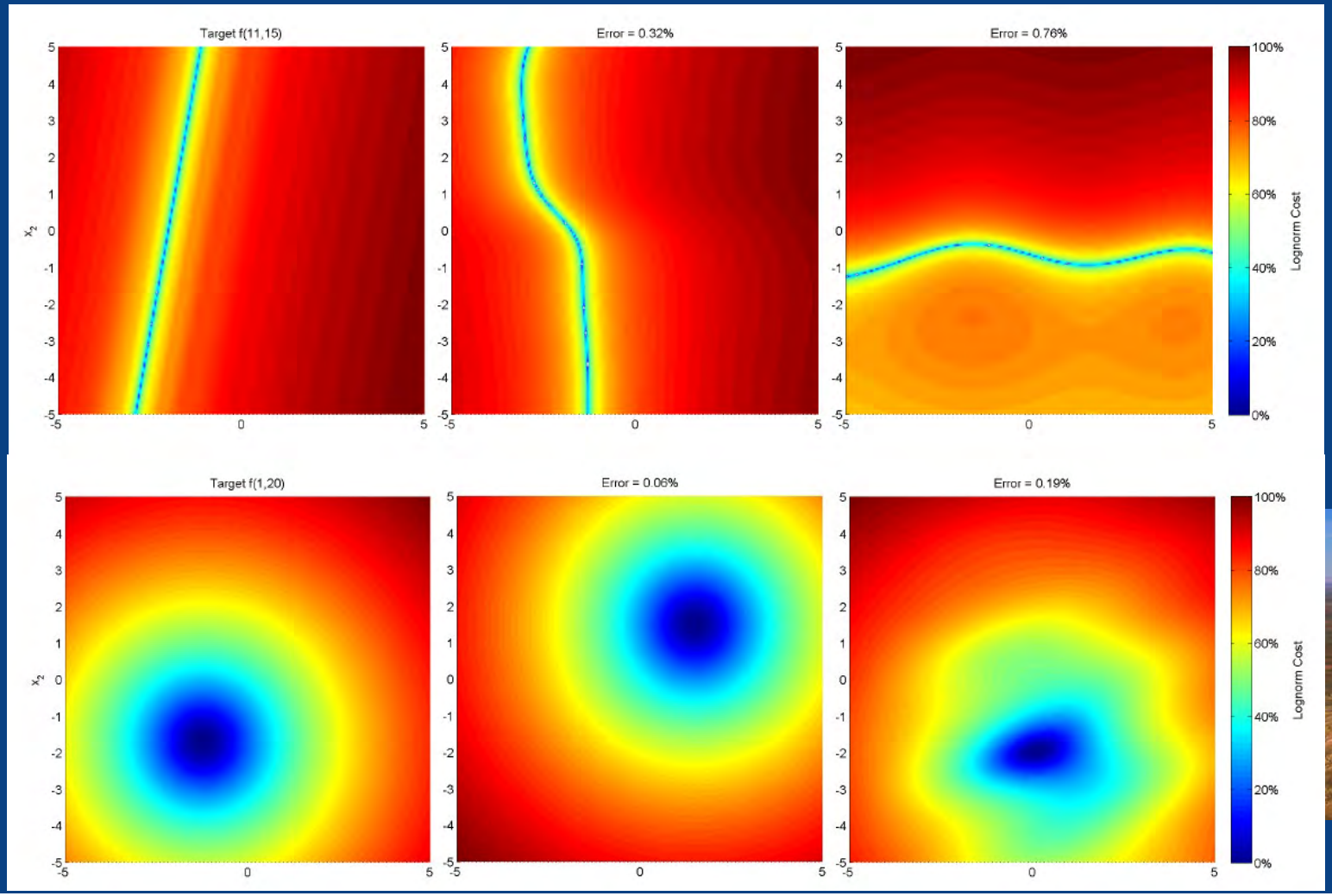


  - Fitness is evaluated based on some measure we seek to improve
  - Weak functions die → average fitness of functions improves
  - Simulation of evolutionary process (many generations) in seconds

What is a "fit" function? Evolutionary goal?

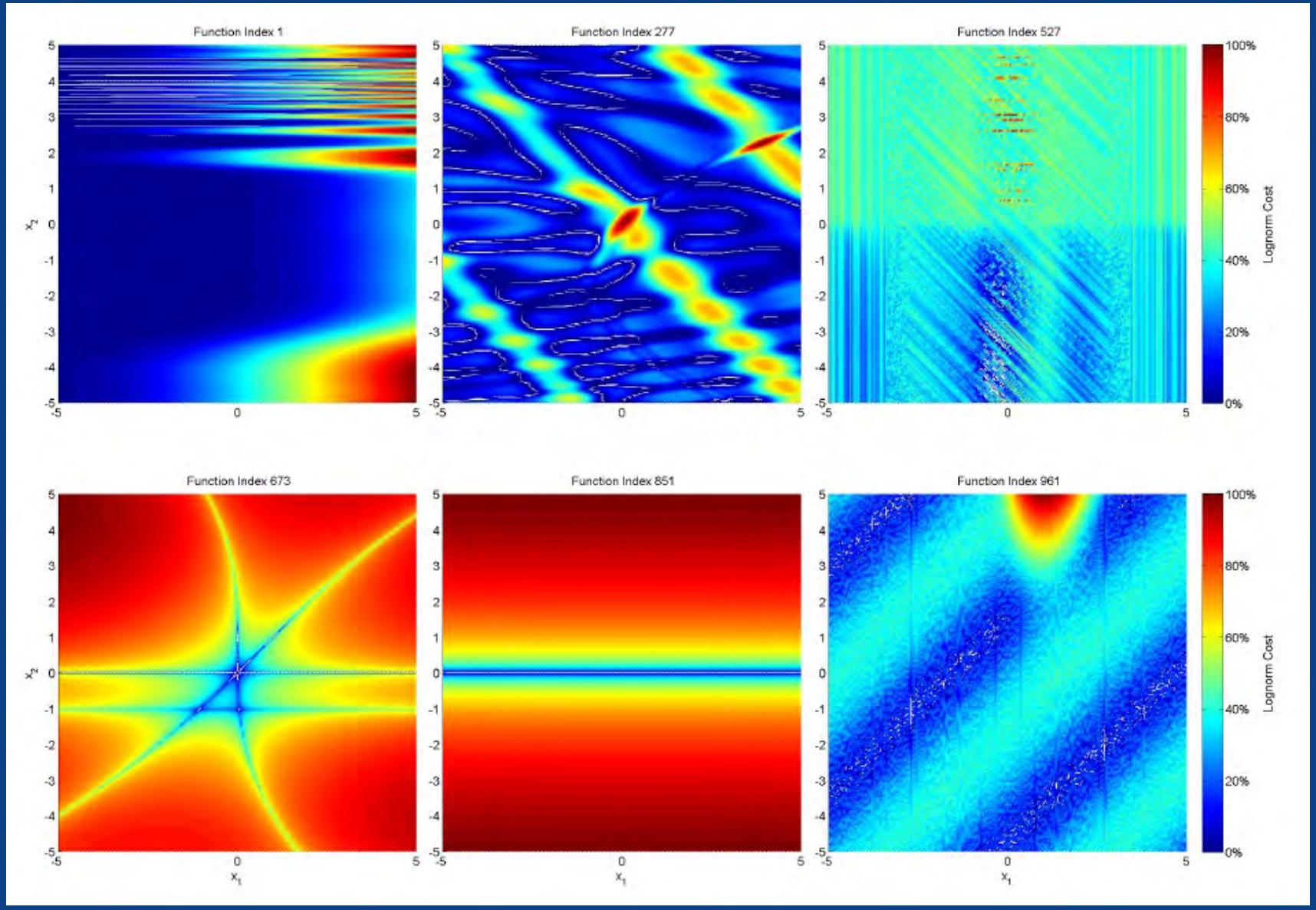
- We seek functions that live at target locations in the instance space
  - To ensure diversity and have a comprehensive set of test examples
  - To find examples that are hard for algorithms and expose weaknesses
- We measure fitness based on distance to a target point
  - Measure feature vector for an offspring (function)
  - Project to 2d instance space
  - Measure distance to target point
  - Fittest = closest to target point
- Over time, we get functions that live anywhere we ask them to be
  - Red points (clones): where we already have test examples (in blue)
  - Green points (new functions): unexplored territory

Cloned functions (confirmation of methodology)



$$f(x,y) = x^2 + y^2 - 2x - 3y - 183.2 - \cos\left(\sin\left(e^{y^2} - \mathrm{e}^{-\mathrm{e}^{-y^4}}\right)\right) - \mathrm{e}^{-\cos^2(x)} - \mathrm{e}^{-e^{\sin(\sin(y))}}$$

New functions (creating diversity)



Quarterly (spring, summer, fall, winter)
176pp. per issue
7 x 10
Founded: 1993

Generating New Space-Filling Test Instances for Continuous Black-Box Optimization

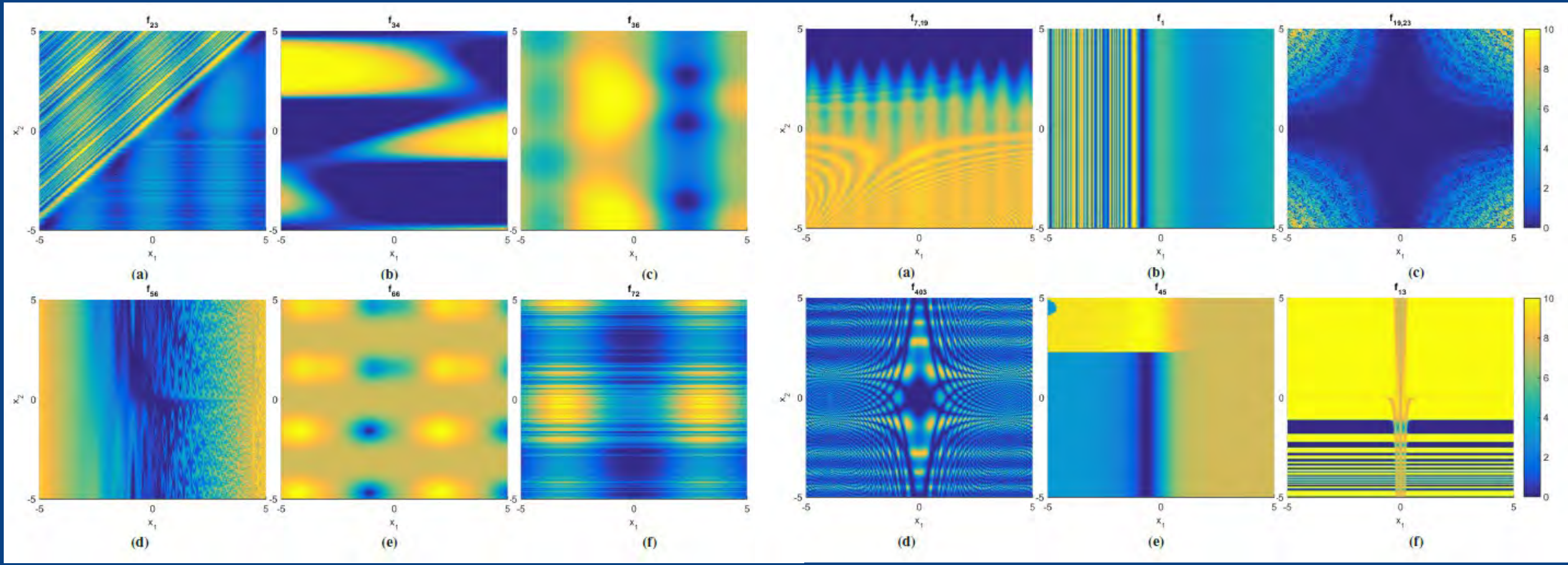
Mario A. Muñoz and Kate Smith-Miles

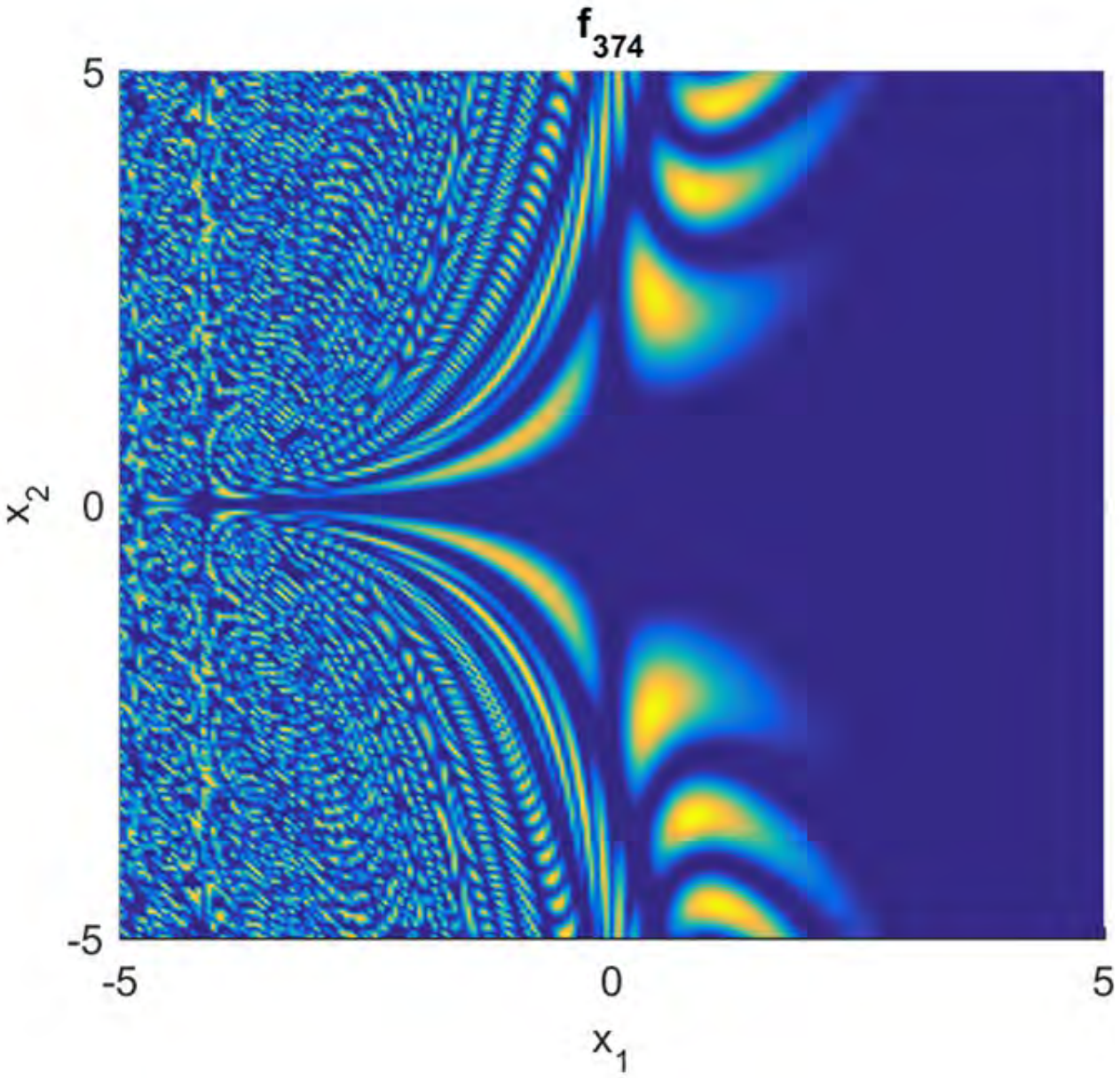
Posted Online September 01, 2020
https://doi.org/10.1162/evco_a_00262

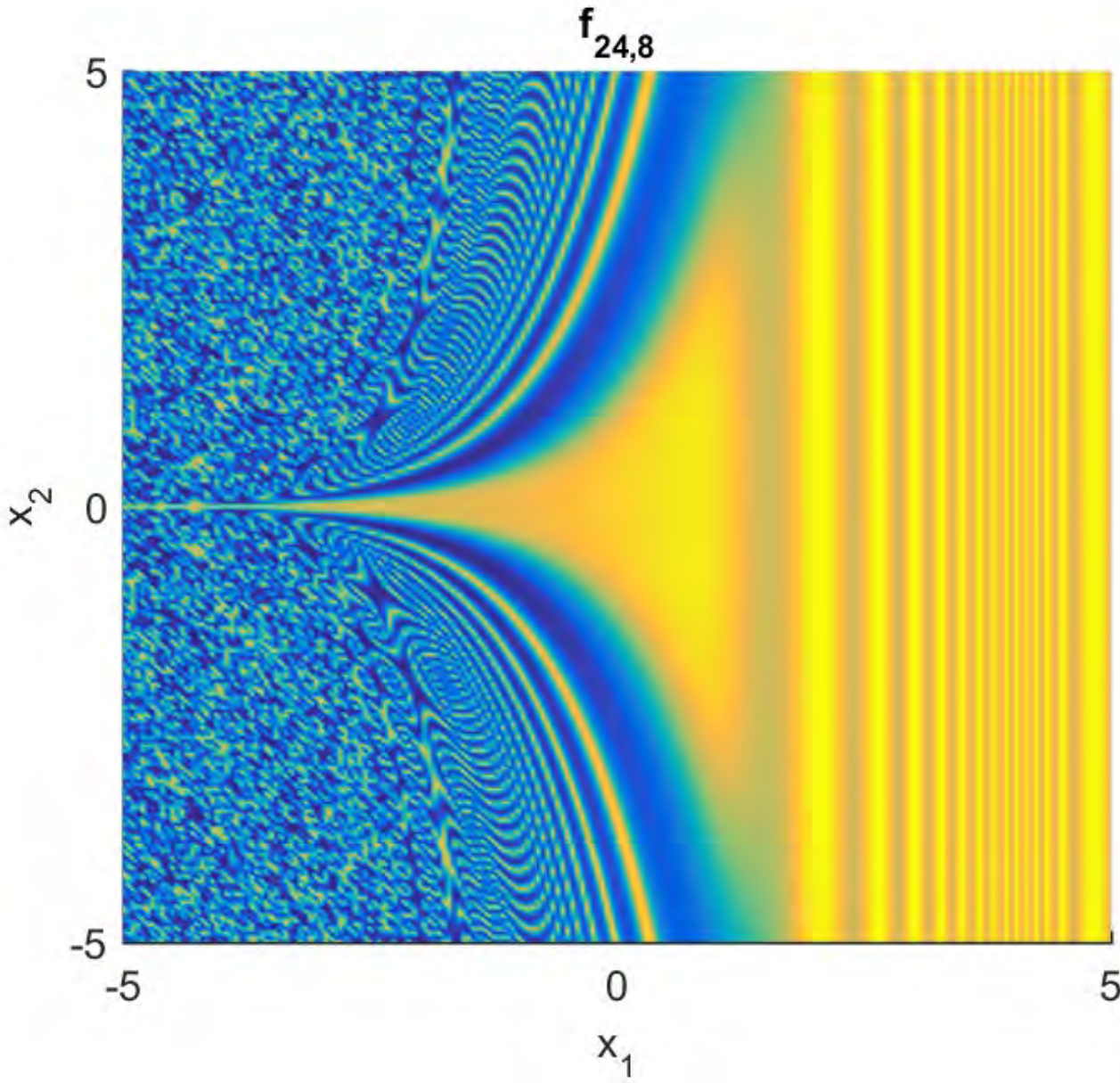
Evolutionary Computation
Volume 28 | Issue 3 | Fall 2020
p.379-404

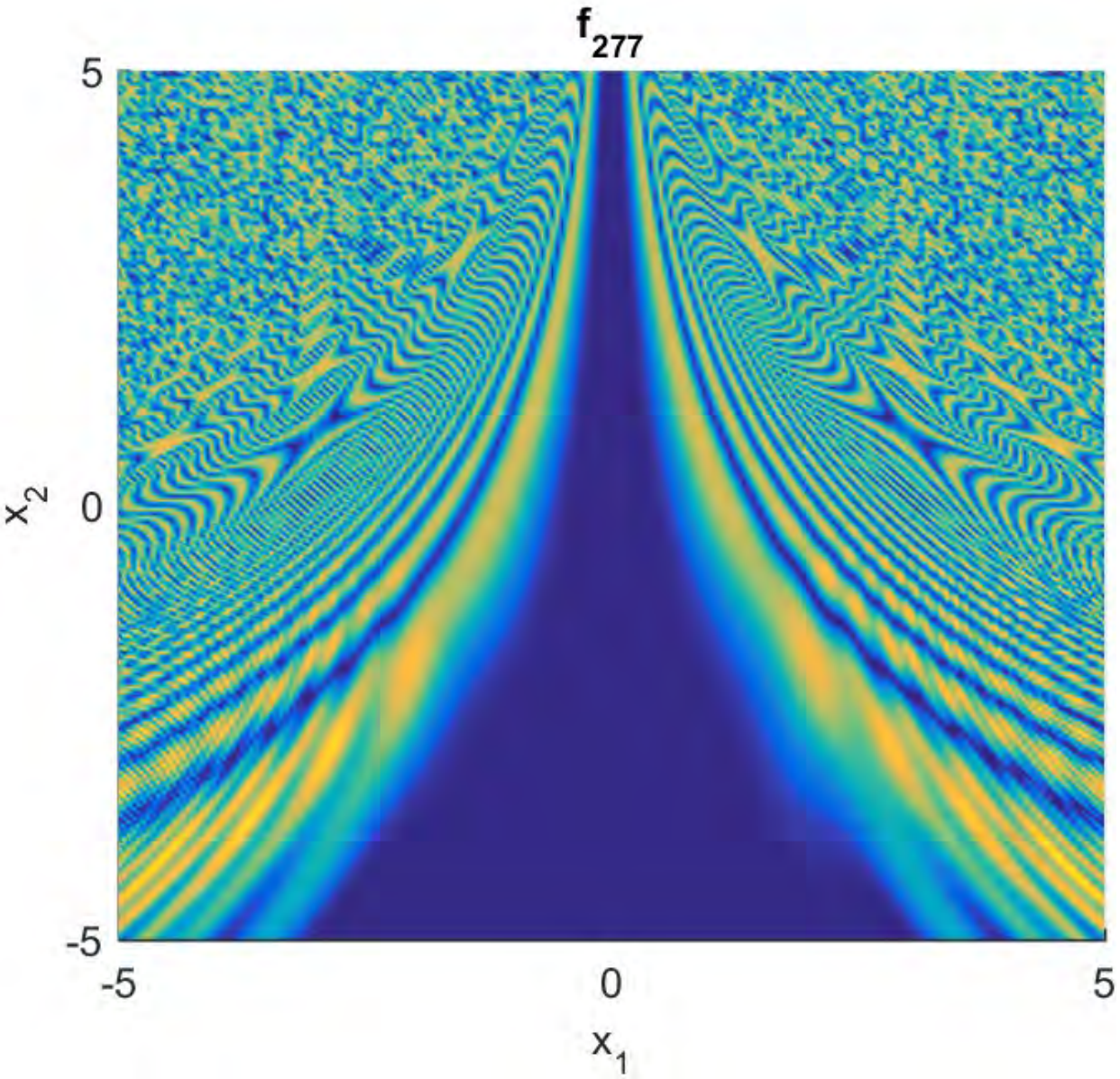
Keywords: Algorithm selection, benchmarking, black-box continuous optimization, exploratory landscape analysis, instance generator.

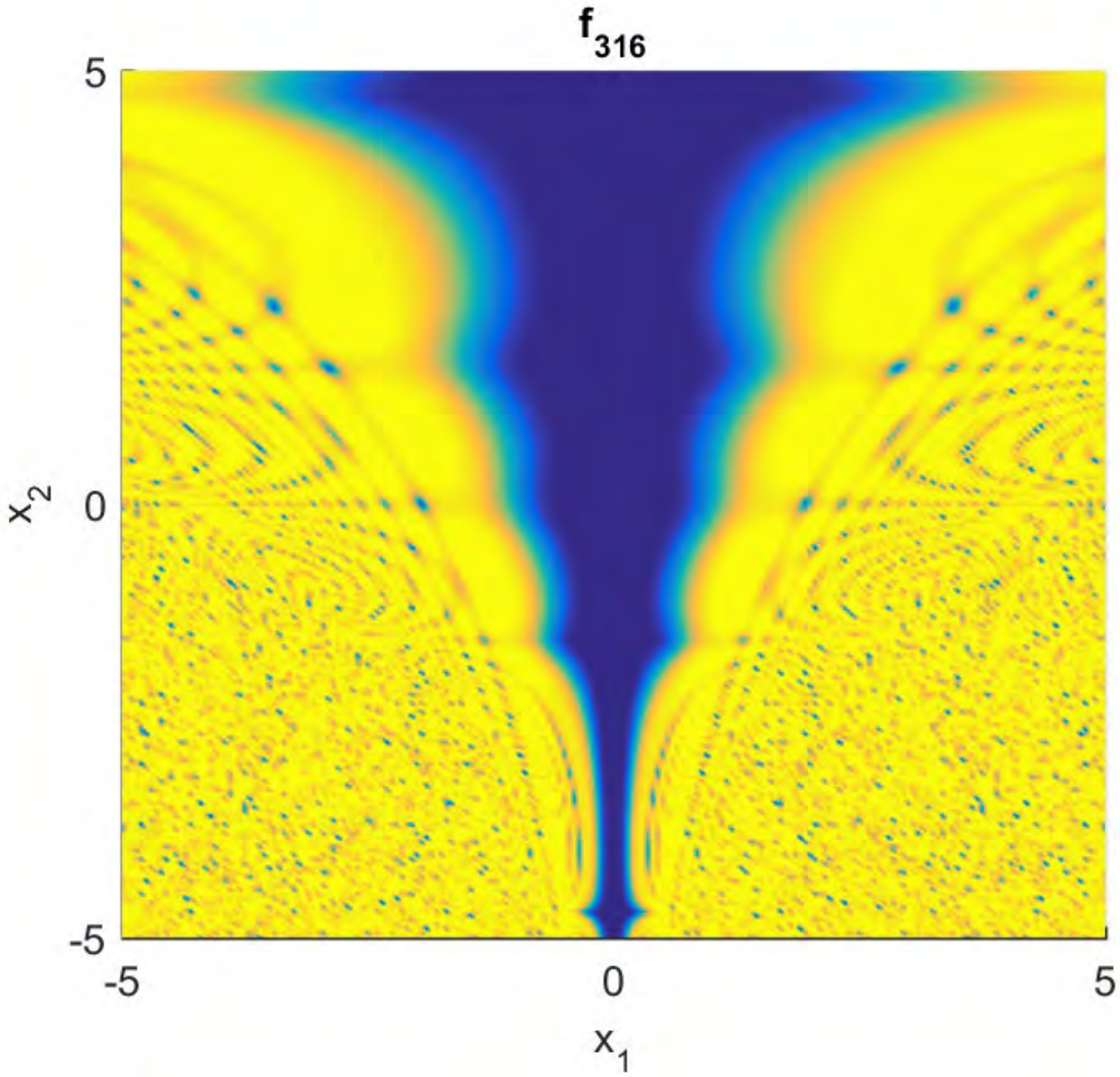
So many beautifully intricate functions!

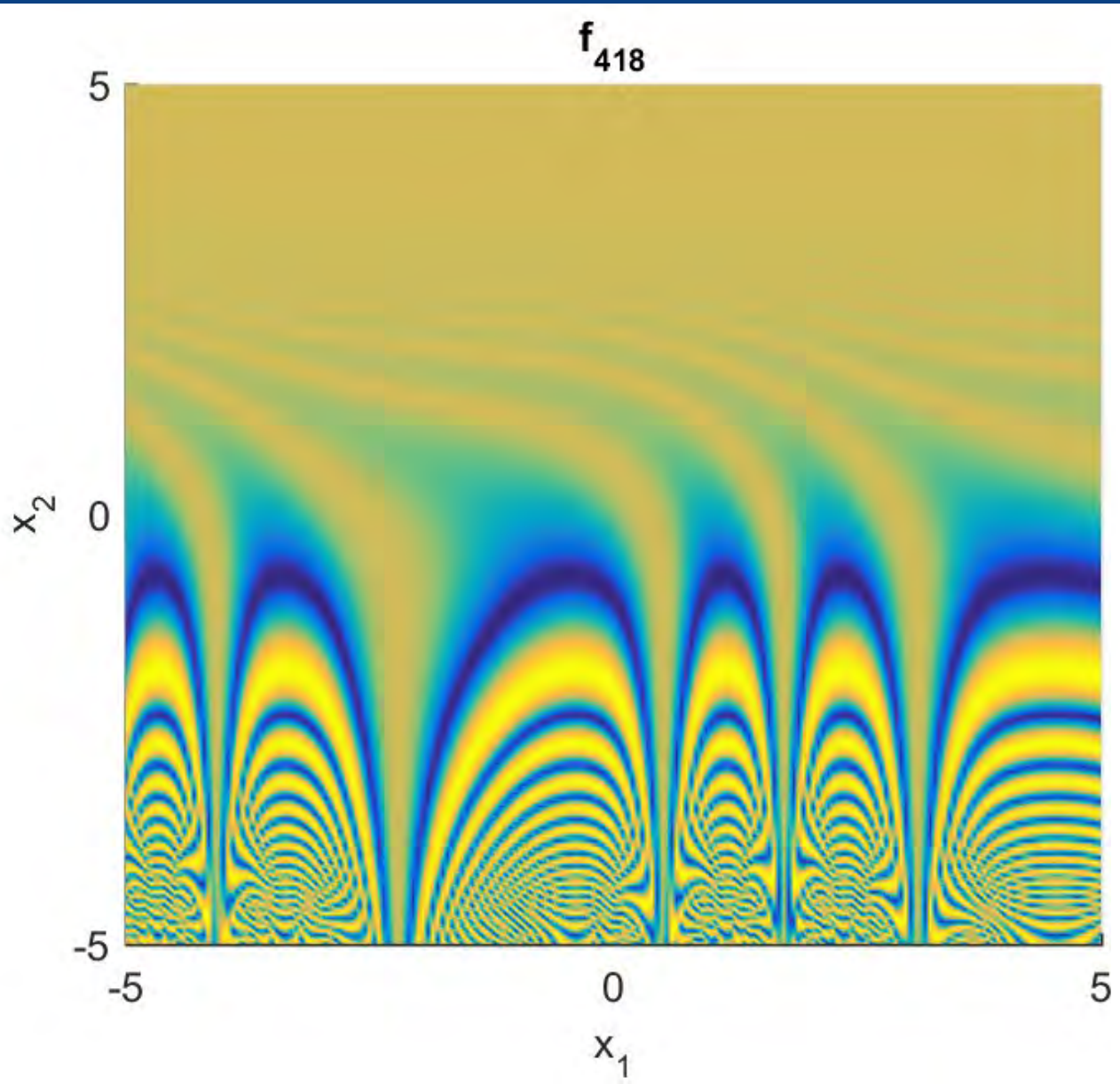


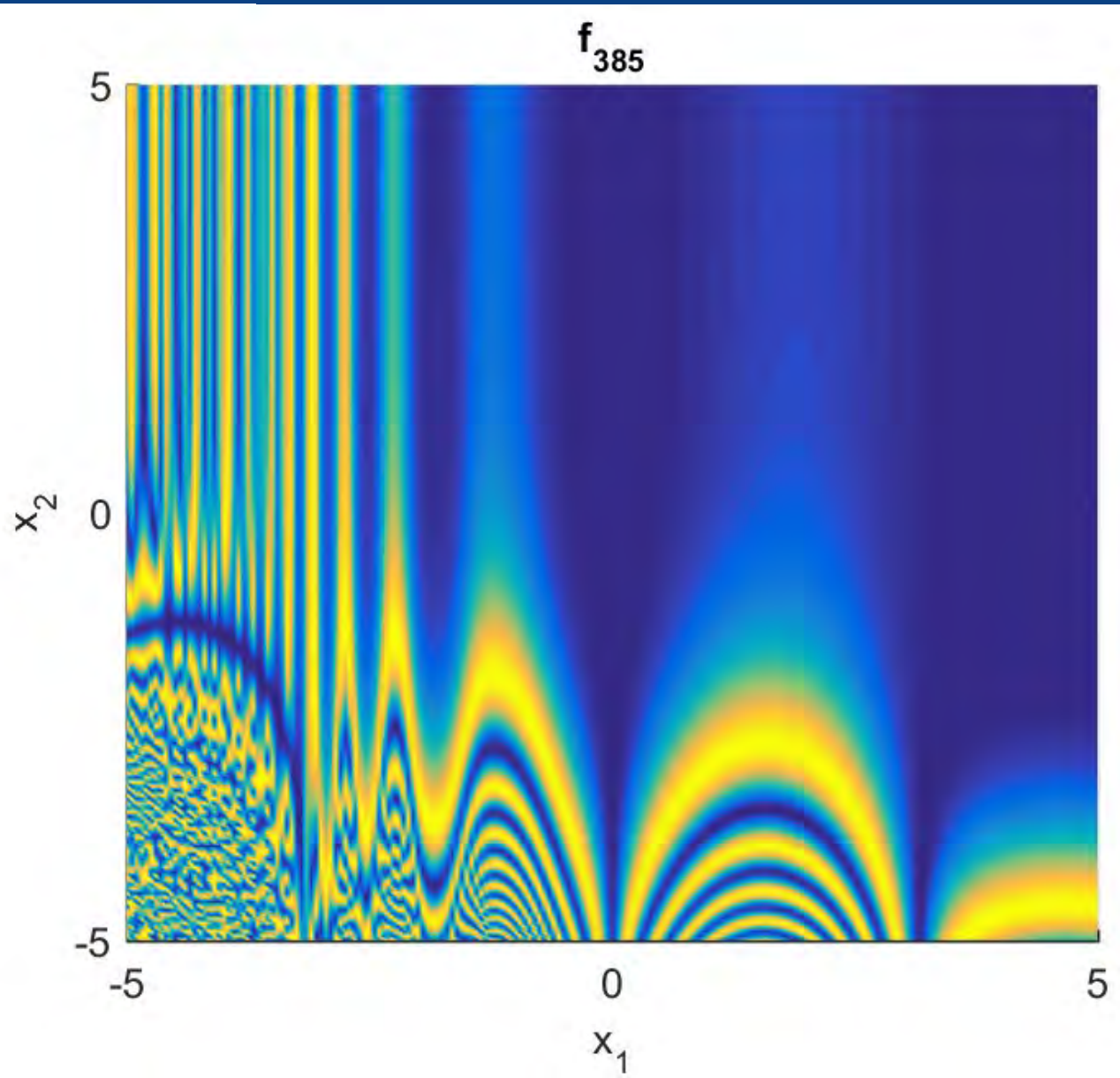

f_{374}
5
0
-5
x_2
-5
0
5
x_1

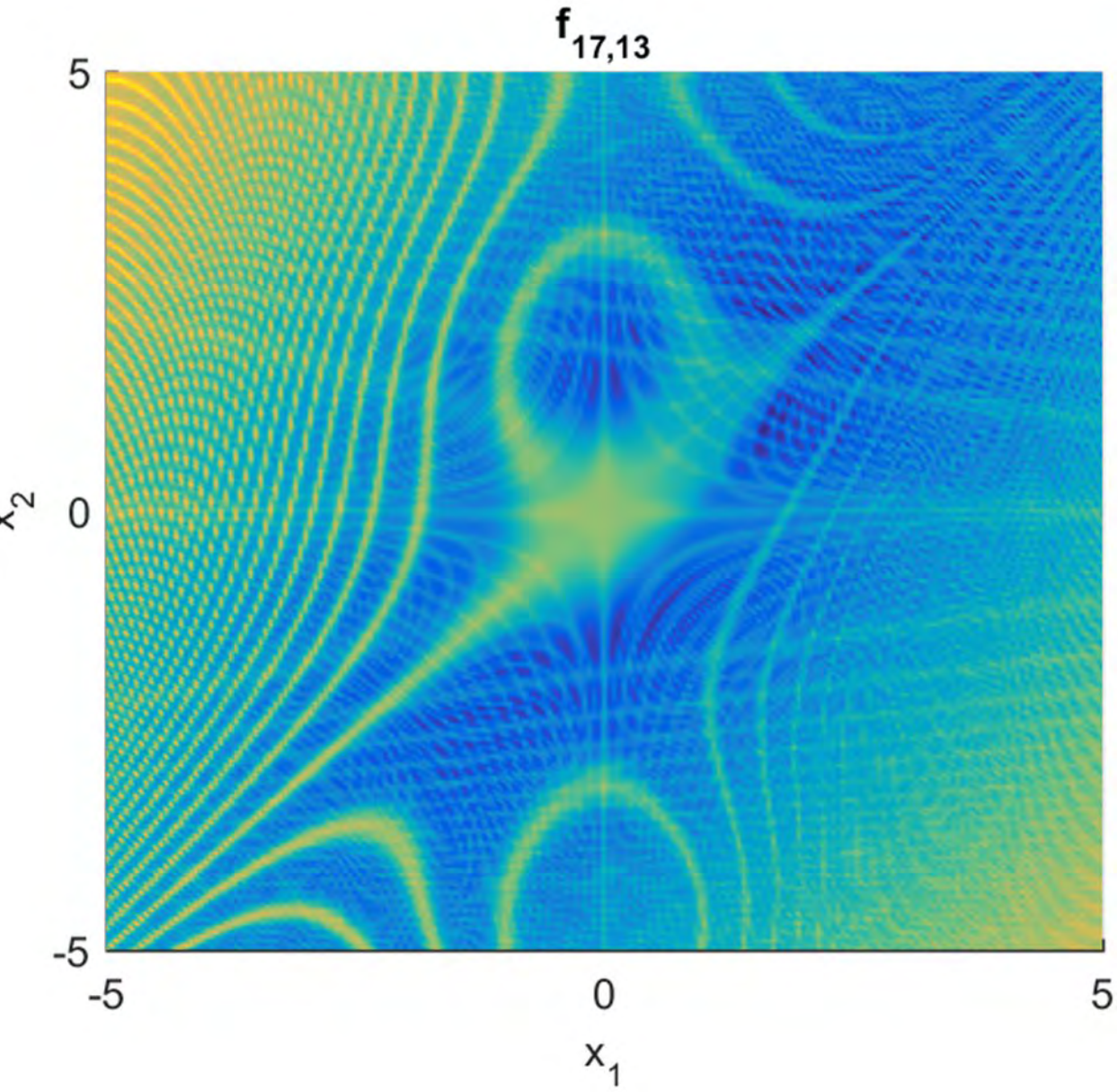

$f_{24,8}$
5
0
-5
x_2
-5
0
5
x_1

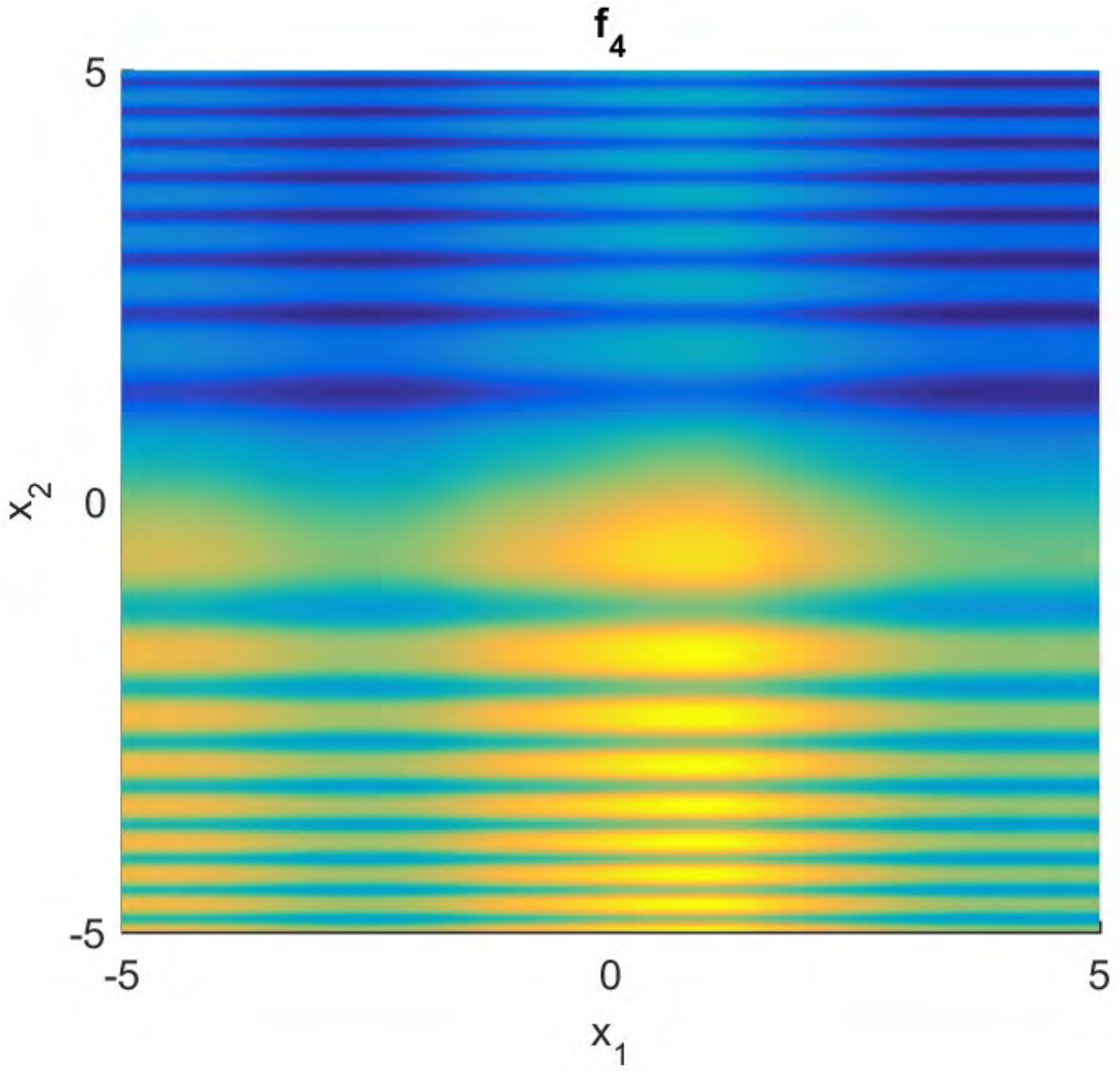

f_{277}
5
0
-5
x_2
-5
0
5
x_1


f_{316}
5
0
-5
x_2
-5
0
5
x_1

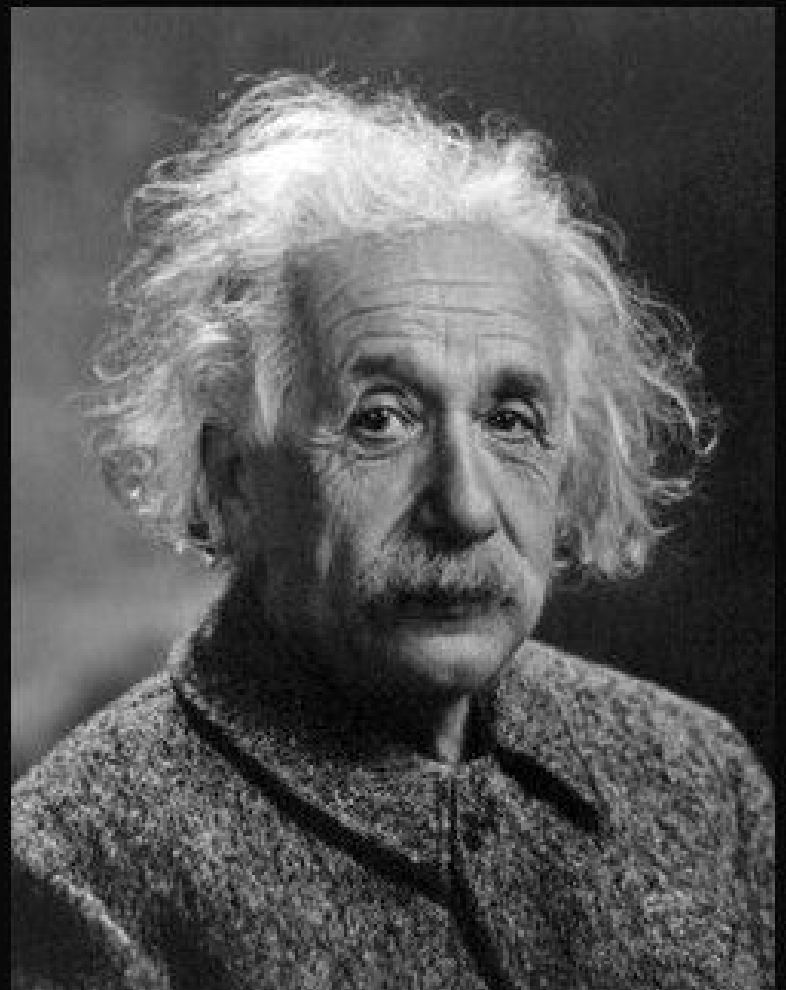

f_{418}
5
0
-5
x_2
-5
0
5
x_1


f_{385}
5
0
-5
x_2
-5
0
5
x_1


$f_{17,13}$
5
0
-5
x_2
-5
0
5
x_1


f_4
5
0
-5
x_2
-5
0
5
x_1

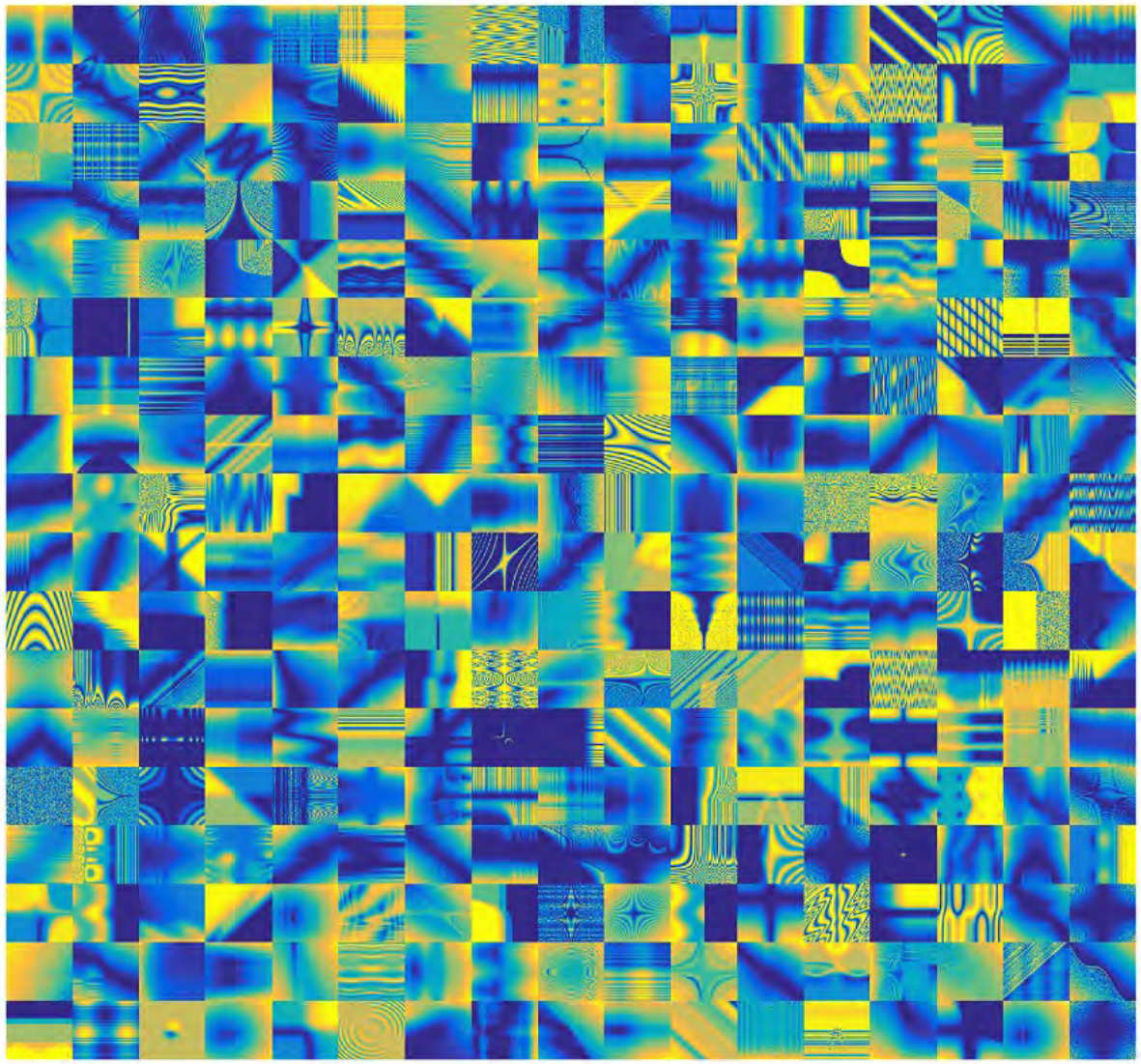
Artistic Motivations



After a certain high level of technical skill is achieved, science and art tend to coalesce in esthetics, plasticity, and form. The greatest scientists are always artists as well.

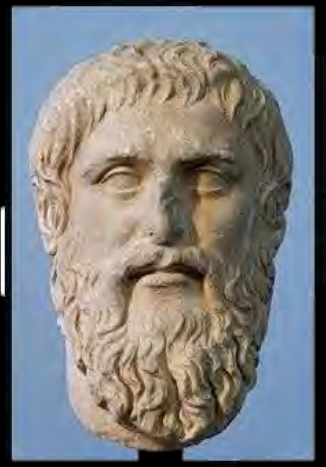
(Albert Einstein)

“Which is your favourite function from 306 choices?”



- Montage: What is an aesthetically pleasing arrangement of these functions?
- Very subjective!
- Depends on value a person places on order and global structure versus randomness
- Aesthetic preferences and personality?

Early musings on the objectivity of beauty (absolute)

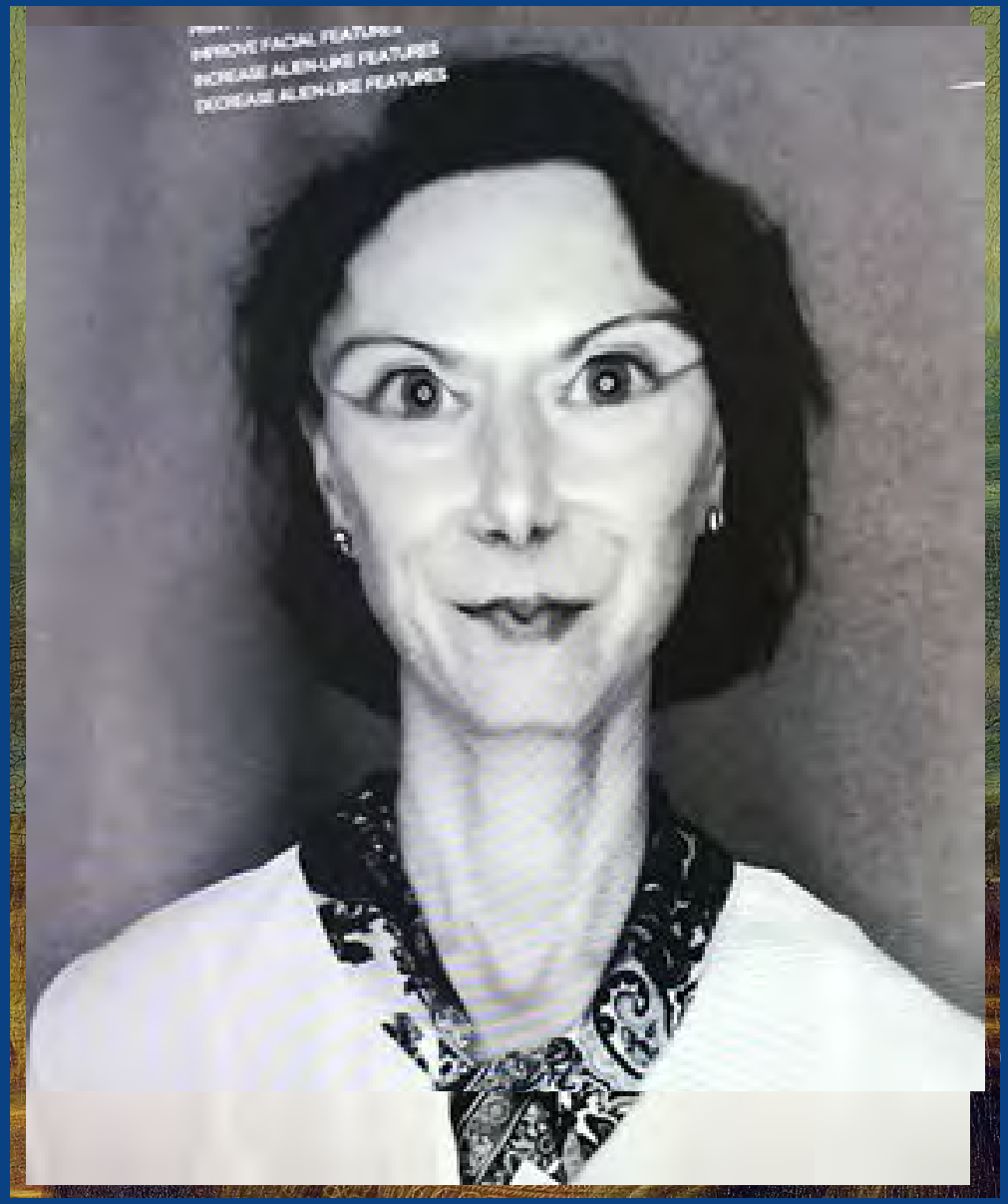


Beauty of style and harmony and grace and good rhythm depend on simplicity — I mean the true simplicity of a rightly and nobly ordered mind and character, not that other simplicity which is only a euphemism for folly.

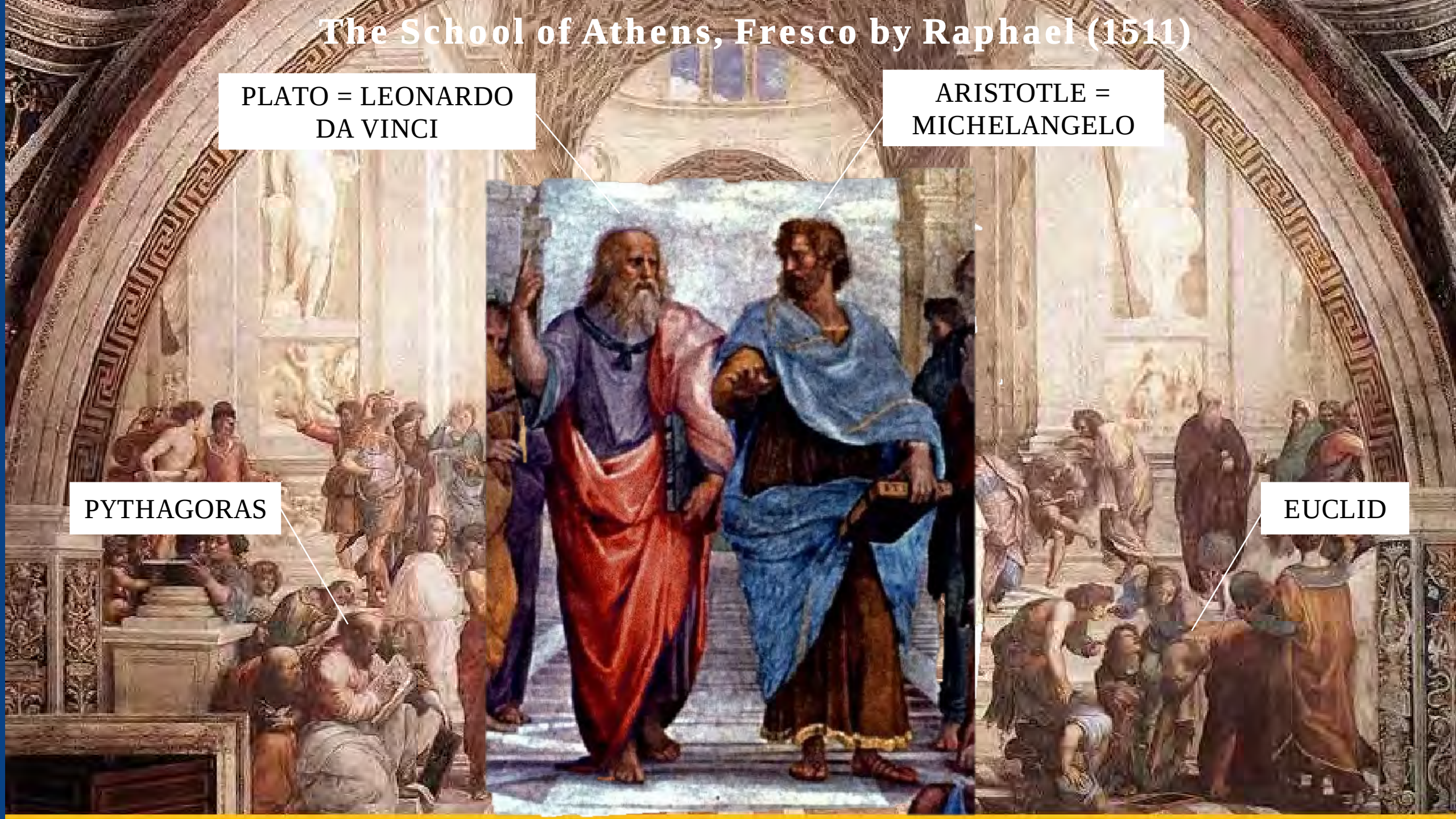
Plato, 370 BC

The chief forms of beauty are order and symmetry and definiteness, which the mathematical sciences demonstrate in a special degree.

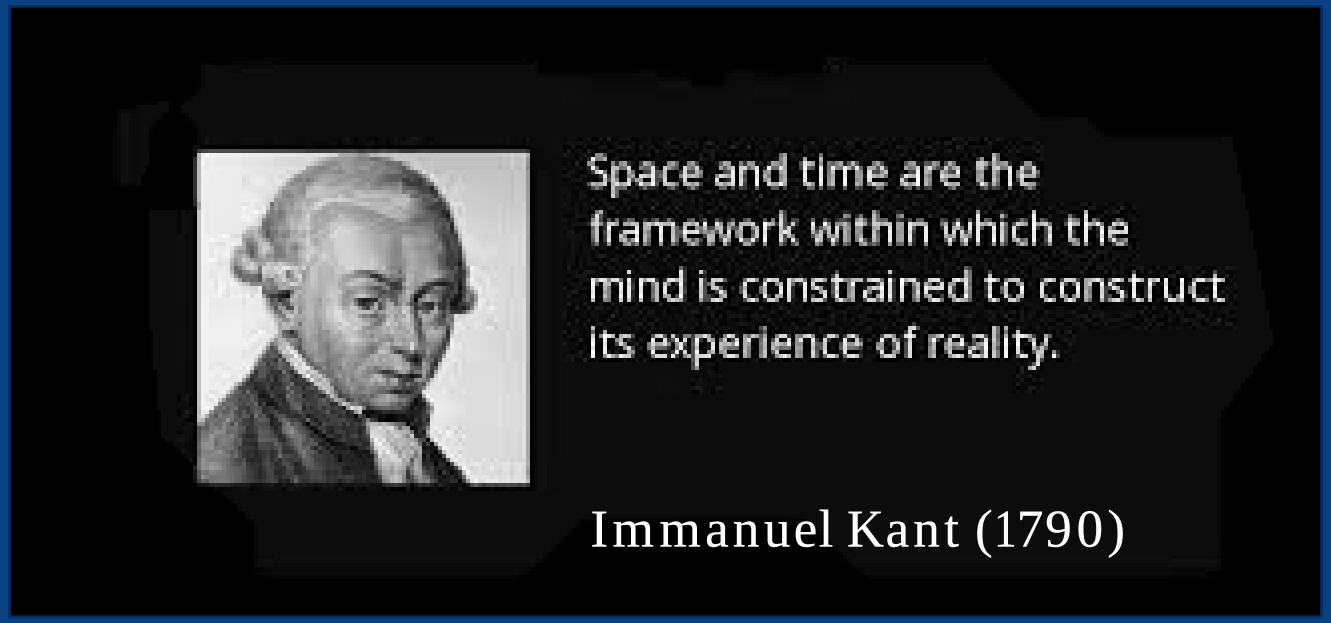
Aristotle, 350 BC



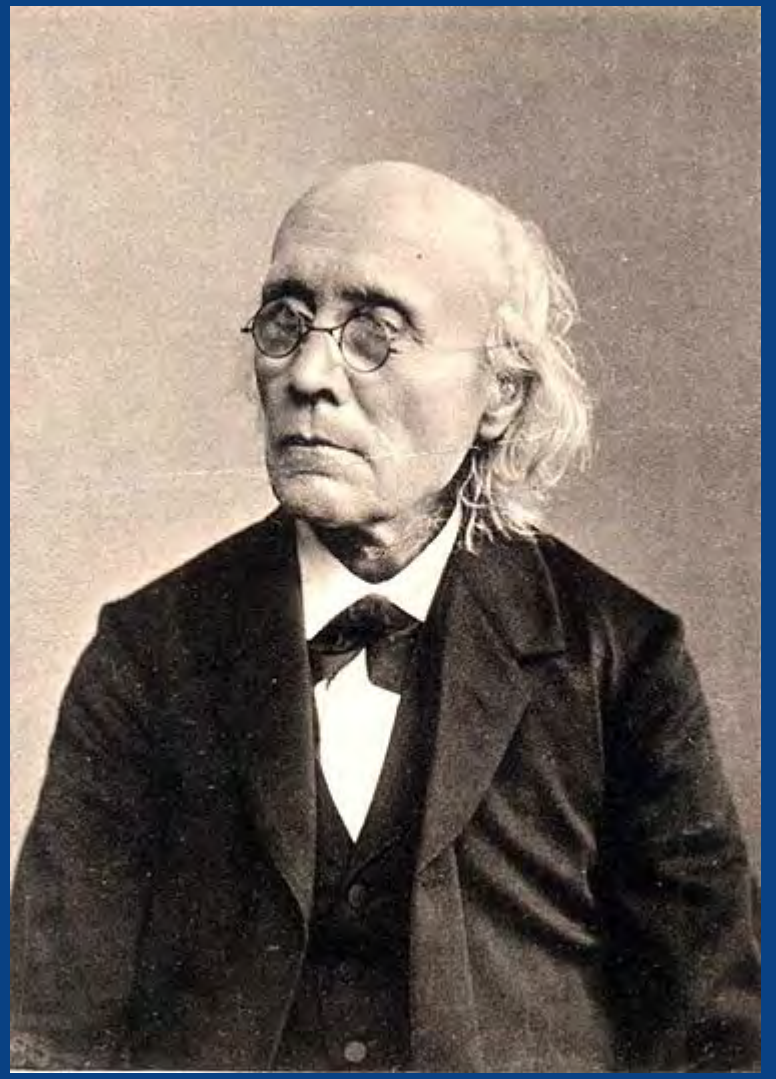
The School of Athens, Fresco by Raphael (1511)



Beauty is in the eye of the beholder … (relative)



- Experimental Aesthetics emerged in 19th Century
  - hedonic analysis = feelings & emotions
  - cognitive analysis = knowledge & understanding
- Neuro-Aesthetics emerging in 21st Century



GUSTAV FECHNER (1801-1887)

What do mathematicians find "beautiful"?

BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 44, Number 4, October 2007, Pages 623–634
S 0273-0979(07)01168-8
Article electronically published on May 2, 2007

WHAT IS GOOD MATHEMATICS?

TERENCE TAO

ABSTRACT. Some personal thoughts and opinions on what "good quality mathematics" is and whether one should try to define this term rigorously. As a case study, the story of Szemerédi's theorem is presented.

1. THE MANY ASPECTS OF MATHEMATICAL QUALITY

We all agree that mathematicians should strive to produce good mathematics. But how does one define "good mathematics", and should one even dare to try at all? Let us first consider the former question. Almost immediately one realises that there are many different types of mathematics which could be designated "good". For instance, "good mathematics" could refer (in no particular order) to

(i) Good mathematical *problem solving* (e.g. a major breakthrough on an important mathematical problem);
(ii) Good mathematical *technique* (e.g. a masterful use of existing methods or the development of new tools);
(iii) Good mathematical *theory* (e.g. a conceptual framework or choice of notation which systematically unifies and generalises an existing body of results);
(iv) Good mathematical *insight* (e.g. a major conceptual simplification or the realisation of a unifying principle, heuristic, analogy, or theme);
(v) Good mathematical *discovery* (e.g. the revelation of an unexpected and intriguing new mathematical phenomenon, connection, or counterexample);

- Mathematicians often describe an argument or proof as "beautiful"
- Tao (2007) suggests at least 20 dimensions define "good mathematics"
  - Beauty
  - Elegance
  - Rigour
  - Creativity
  - Depth
  - etc.

What does beautiful mean to a mathematician?

- Survey of 255 mathematicians
- "Think of your favourite proof"
- Rank 80 adjectives
- Statistical analysis (PCA, R^2)
  - Each proof can be described as a point in a 4d space
  - Beauty is aligned to *aesthetic*
  - Simplicity is aligned to *intricacy*
  - No correlation between beauty and simplicity

- *aesthetic dimension*
  - e.g. elegance, ingenuity, creativity
- *intricacy dimension*
  - simplicity and confusion at opposite ends of the scale
- *utility dimension*
  - e.g. applicability, practicality, informativeness
- *precision dimension*
  - e.g. rigor, carefulness, meticulousness

Beauty strongly correlated with: "elegant", "pleasing", "ingenious", "inspired", "enlightening" and "creative"

Elegance

- "The quality of being pleasingly ingenious and simple; neatness" (Oxford Dictionary)

- Contradiction:
  - Beauty means elegance to mathematicians
  - Elegance means simplicity to everyone else
  - For mathematicians, simplicity doesn't create elegance or beauty

- Survey suggests that what is elegant to a mathematician can include a significant amount of *complexity* without detracting from its beauty.

The Role of Complexity and Order in Aesthetics

"Order and complexity cannot exist without each other.

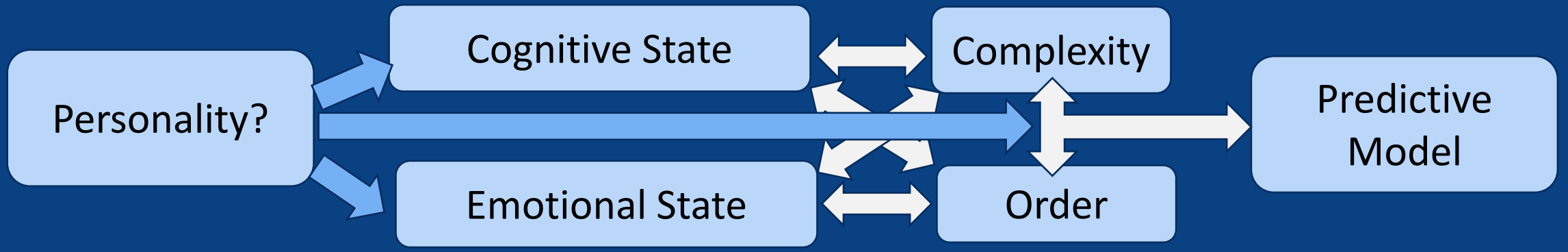
Complexity without order produces confusion; Order without complexity produces boredom.

It has long been recognized that great works combine high order with high complexity."

Toward a Psychology of Art: Collected Essays
Rudolf Arnheim (1972)

Understanding individual preferences

- Studies show people's perception of aesthetic value of visual art is influenced by :
  - *Subjective* individual preferences for complexity and order
    i) amount and variety of elements;
    ii) way elements are organized; ("unity in variety")
    iii) asymmetry
  - *Objective* (consistently agreed) properties such as color combinations, balance points, spatial properties of objects and shapes



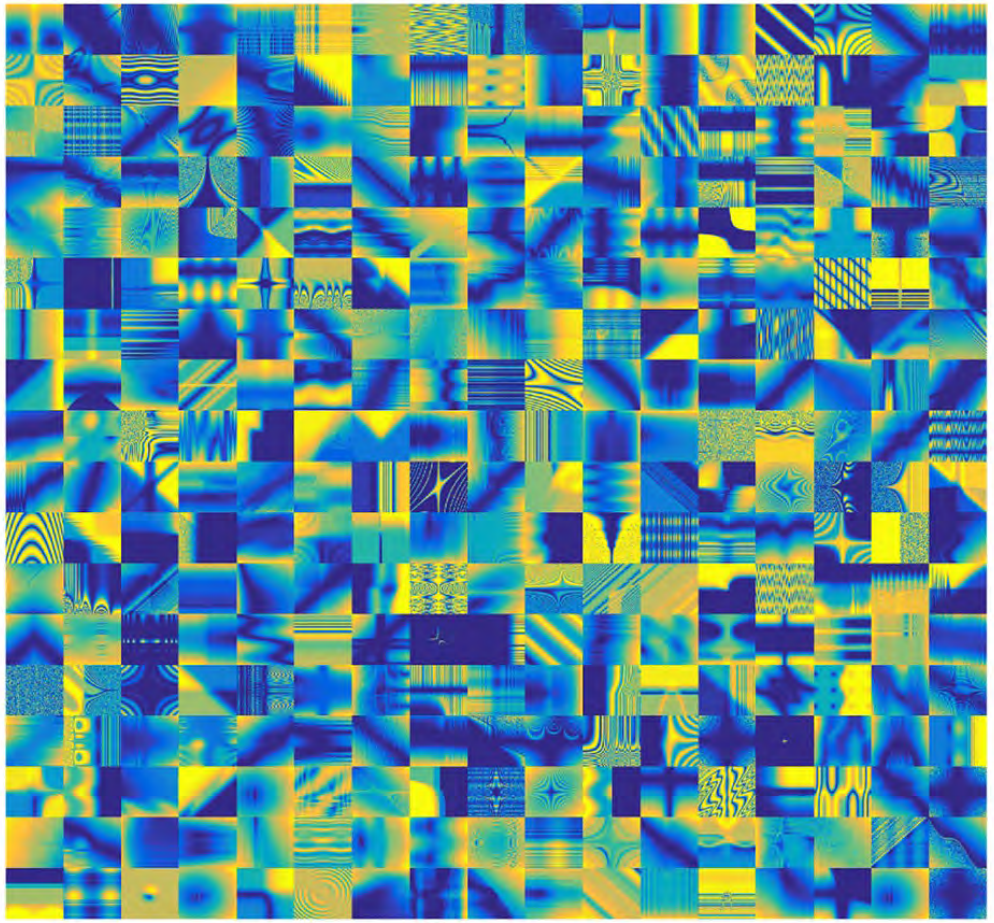
Role of Personality in Order-Disorder preference?

Personality traits have been shown to correlate with aesthetic taste

- conscientiousness
- neuroticism
- agreeableness
- extroversion
- openness-to-experience
- unconventionality
- sensation-seeking

Balancing Order and Complexity in our Montage

- Survey with 10 random montages
- True random appealed to no-one
- Some preferred *global structure* created by connecting blue rivers; unsatisfied when blue rivers stopped abruptly (macro order);
- Others preferred “random”, meaning no imposed macro order that distracts from appreciating micro diversity (“cheap”, “hideous”)



Beauty as “unity in variety” or “order in complexity” … at what scale?

New Artistic Intention

- Fascinating to observe relationship between
  personality traits → order-disorder preference → aesthetic taste
  at what scale do we seek order?

No longer aiming to select the most aesthetic image, or construct the most aesthetic montage, to satisfy the most people

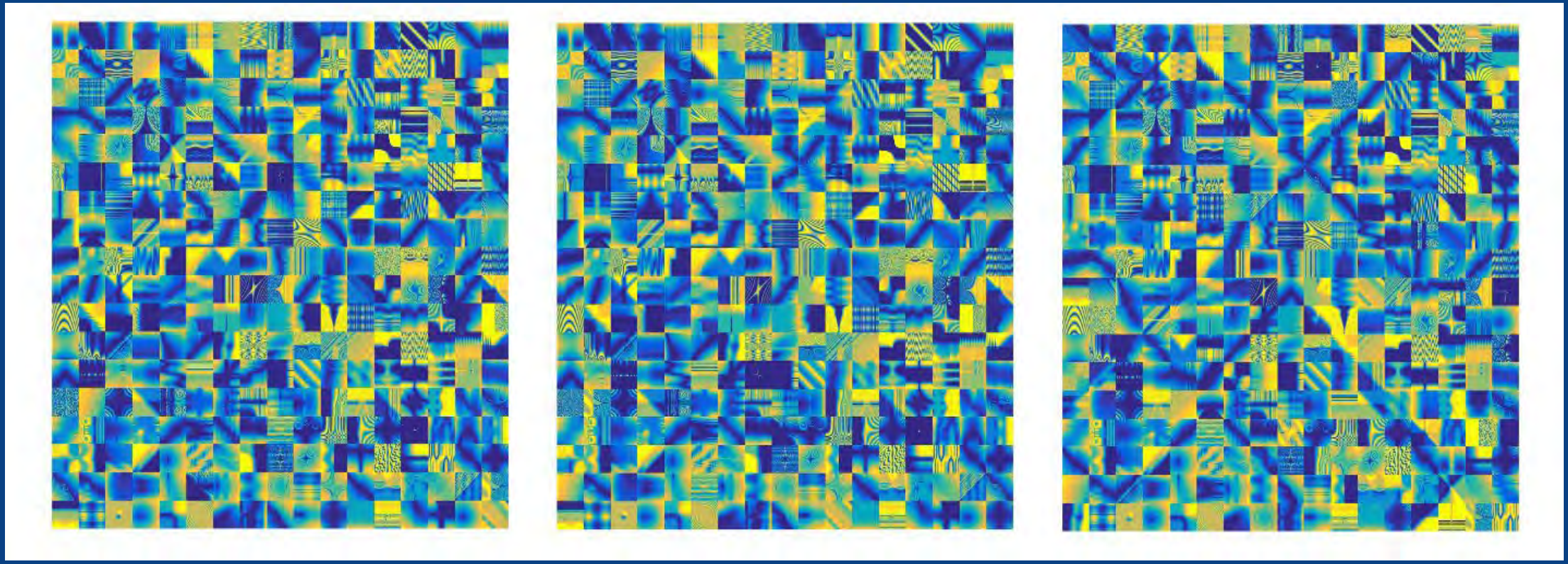
- Aim: Create an artwork that conveys the observation

Aesthetic taste is a spectrum that reflects preference from disorder to order

- Method: Manual swaps by eye to enhance or destroy blue rivers

The making of “Negentropy Triptych” …

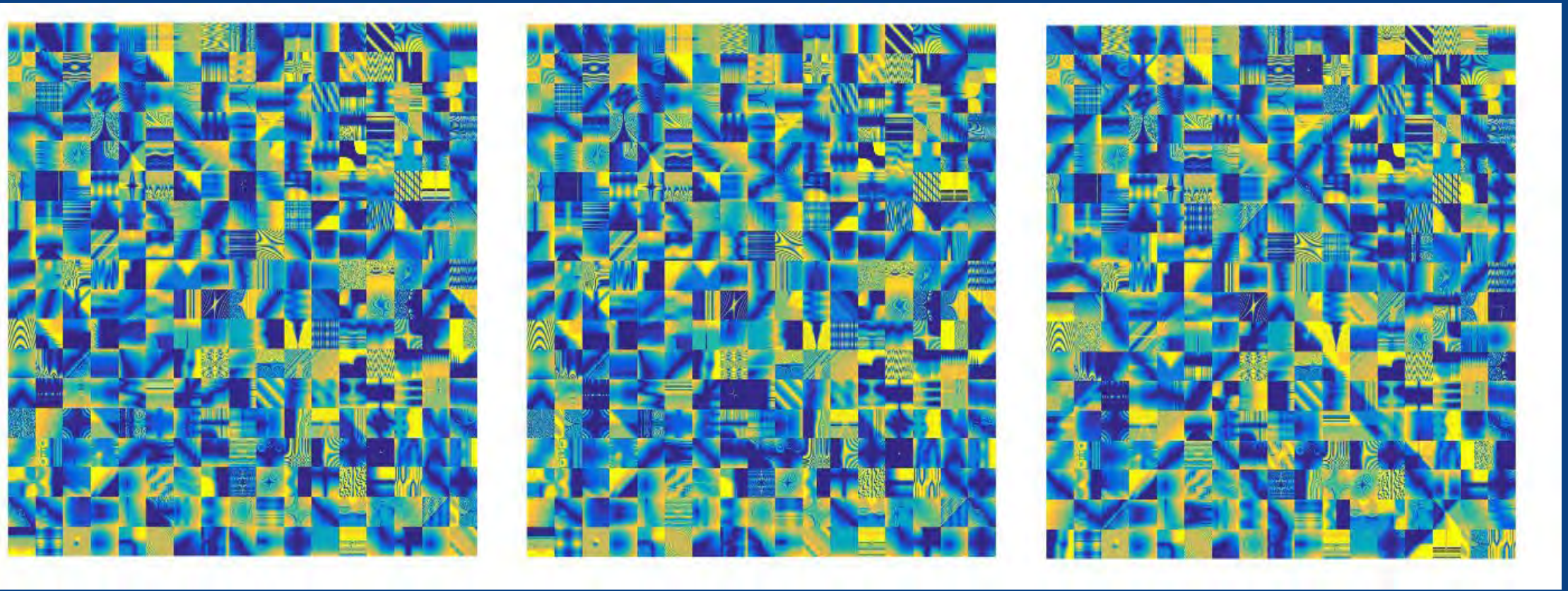
- Entropy (noun): lack of order or predictability; gradual decline into disorder
- Negentropy is the reverse: order emerging from disorder



- Random (middle); permutations to enhance background global structure (right); permutations to destroy accidental structure (left)

THE UNIVERSITY OF MELBOURNE

Negentropy Triptych



Negentropy Triptych (2019) by Kate Smith-Miles and Mario Andres Munoz Acosta

Can an optimisation algorithm do better?

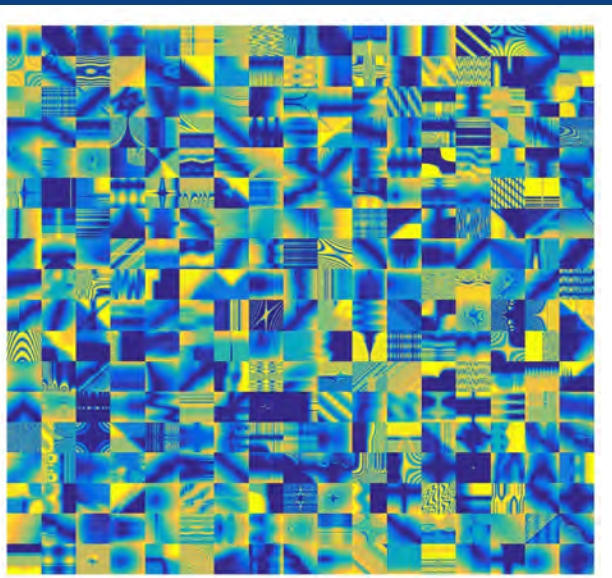
- Aim: Find an optimal placement of 306 images to maximise (or minimize) “blue river connectedness”
- Method:
  - Initial random montage;
  - Extract “dark enough” blue as binary;
  - Greedy algorithm with 7 types of swaps

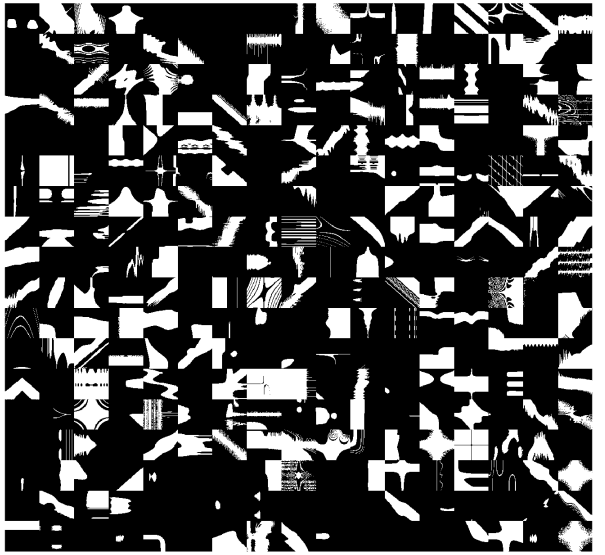
    global random; local neighbours (l,r,t,b); flips (h,v)
  - Measure of “blue river connectedness”

    Initialise with some large connected areas ;

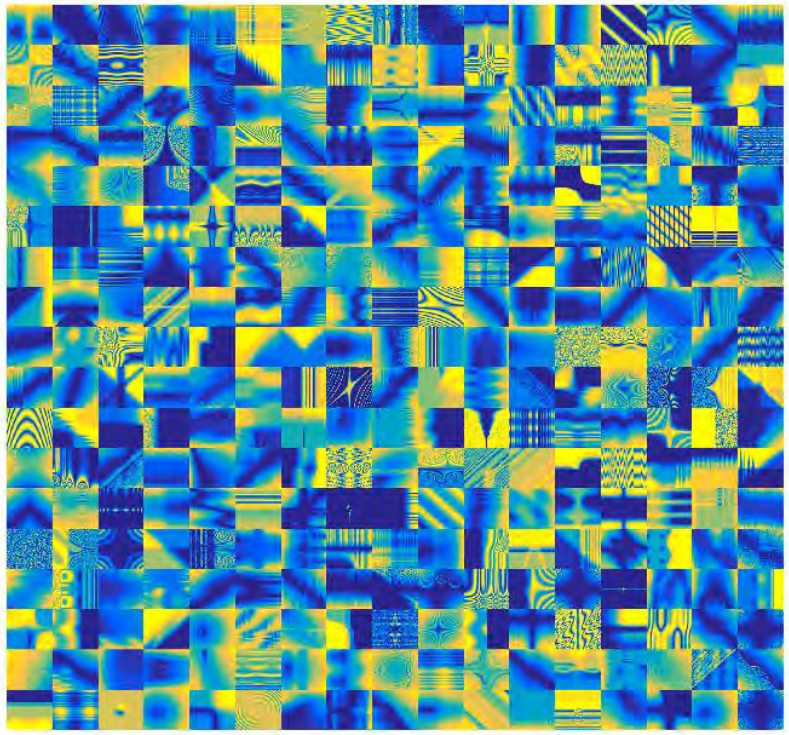
    Try to join into larger connected regions via swaps;

    → maximize total area of connected blue regions

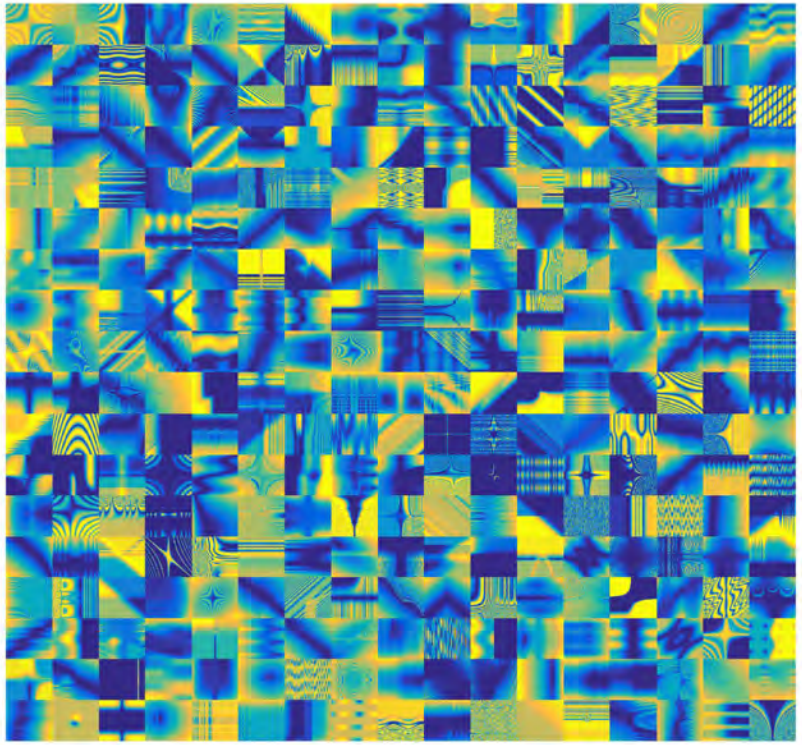




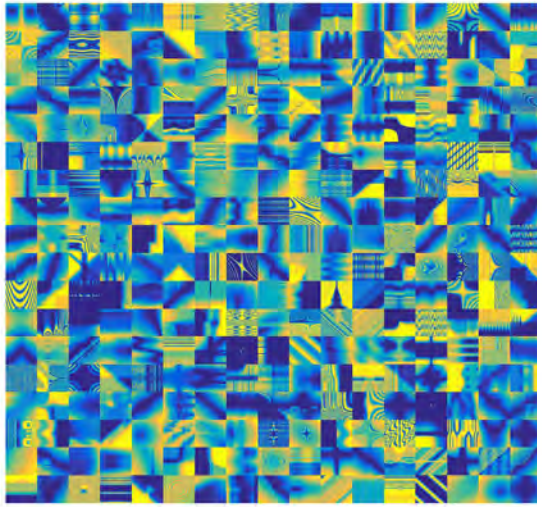
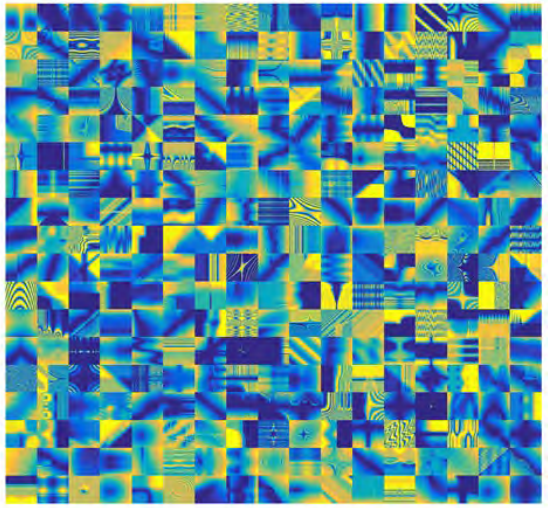
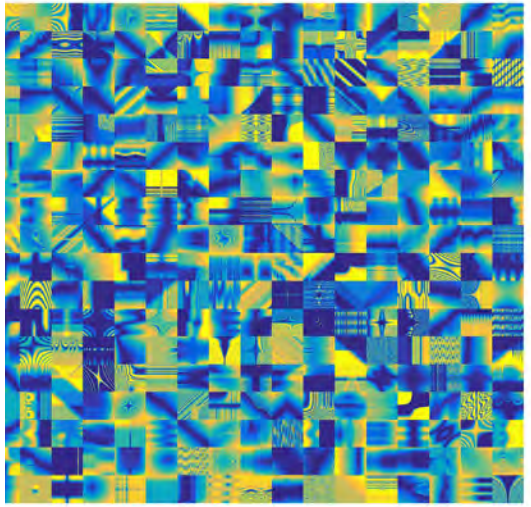
The first 1000 iterations ....



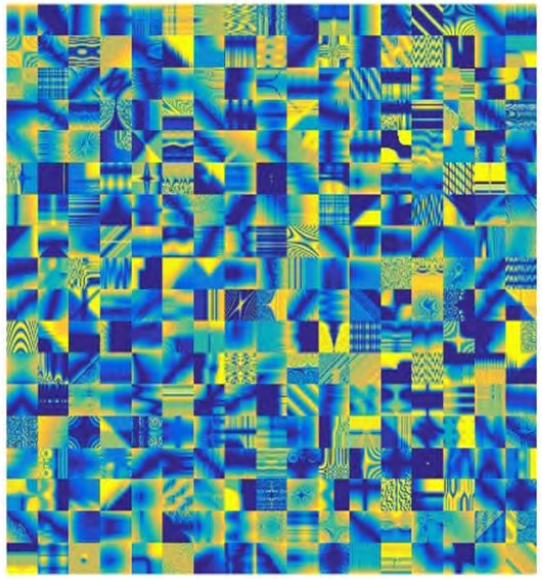
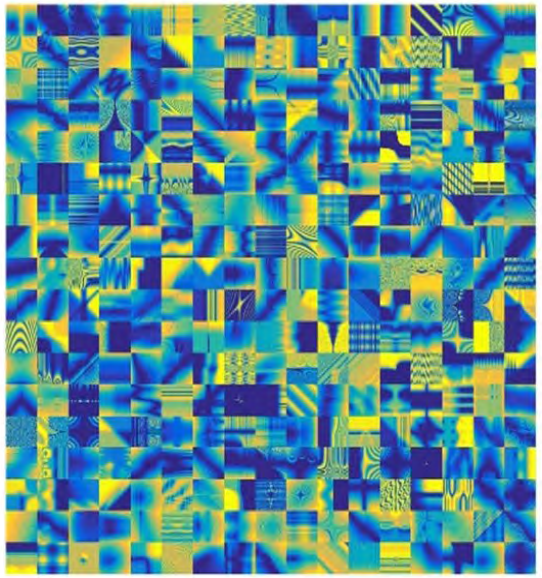
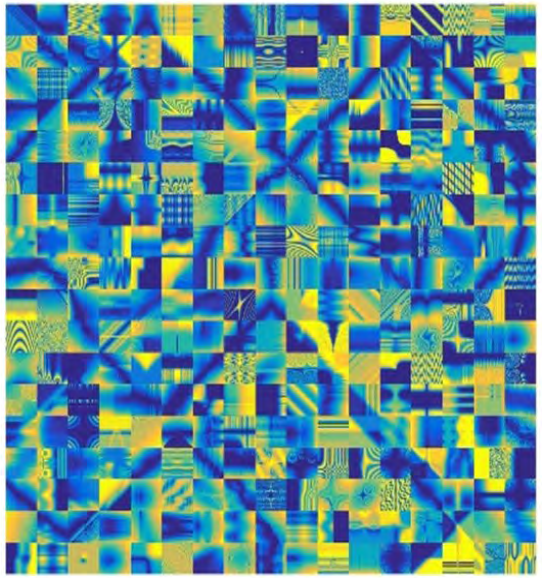
After 20K iterations ...



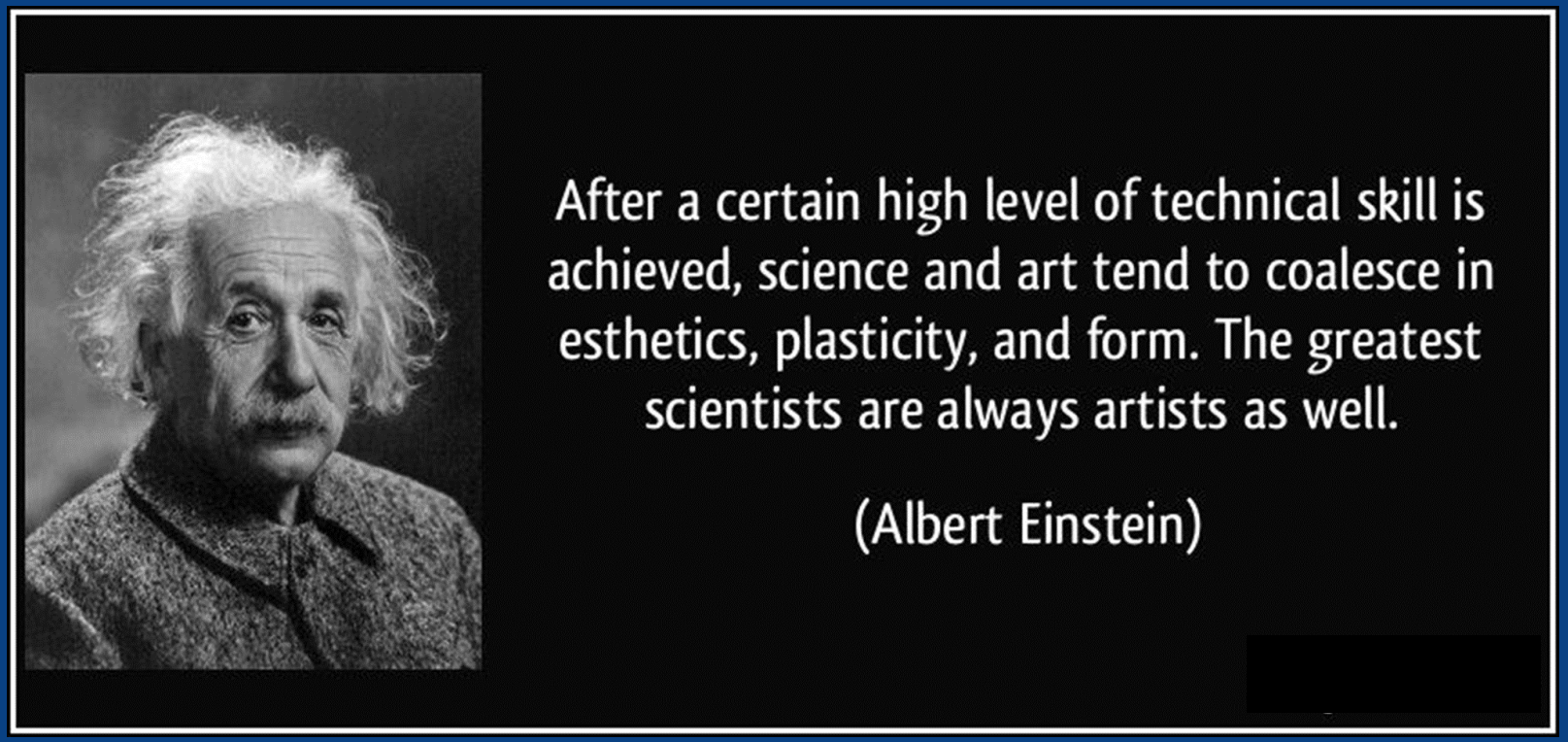
Algorithm as Artist (20k swaps)

Human as Artist (30 swaps)

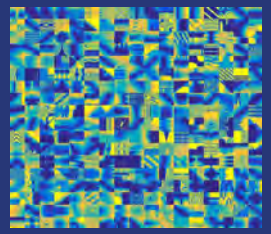
Scientists or Artists?



THE UNIVERSITY OF MELBOURNE

MELBOURNE STORY

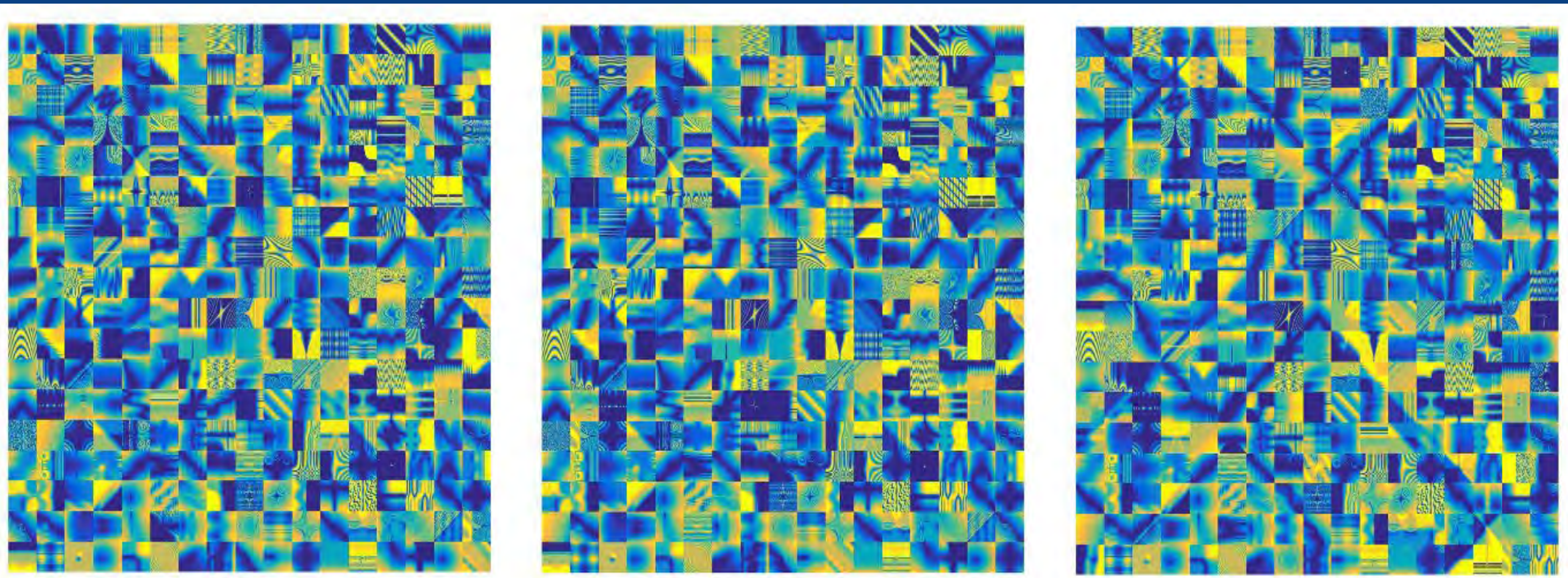
Algorithms of Art



THE UNIVERSITY OF MELBOURNE

Thank you … questions?

email: smith-miles@unimelb.edu.au



Acknowledgement: Negentropy Triptych (2019) by Kate Smith-Miles and Mario Andres Munoz Acosta is an unintended outcome of the project “Stress-testing algorithms: generating new test instances to elicit insights", funded by the Australian Research Council’s Laureate Fellowship scheme