

Genetic Algorithms and Evolutionary Computing

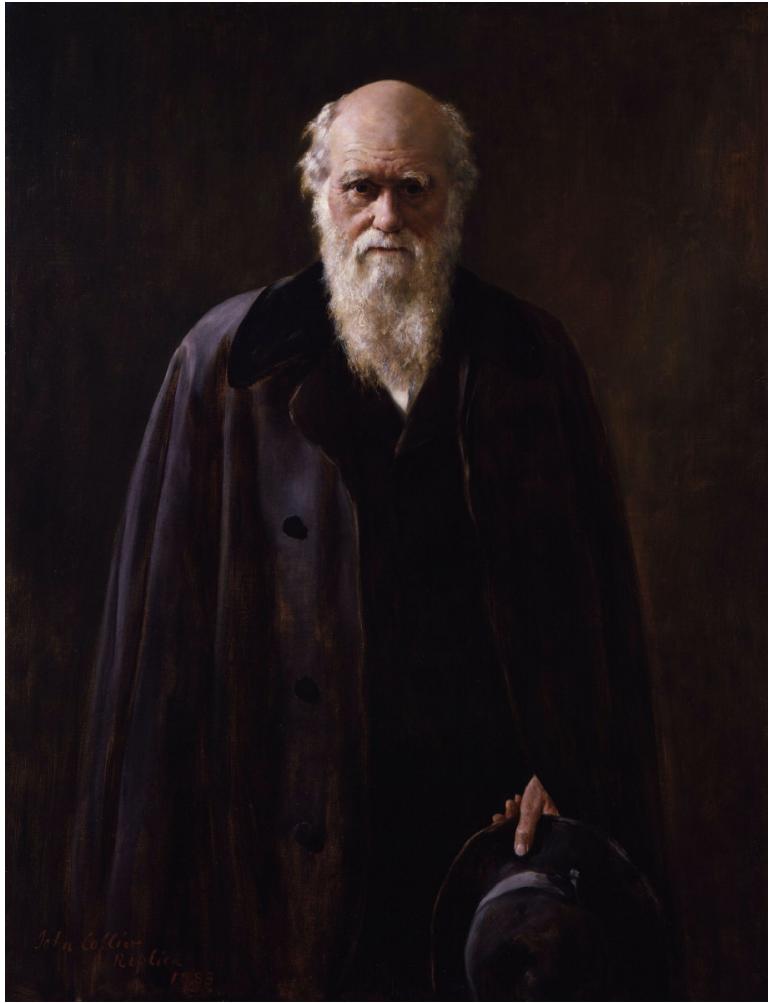
Special Topics in Data Science



Northeastern University

J. Rachlin

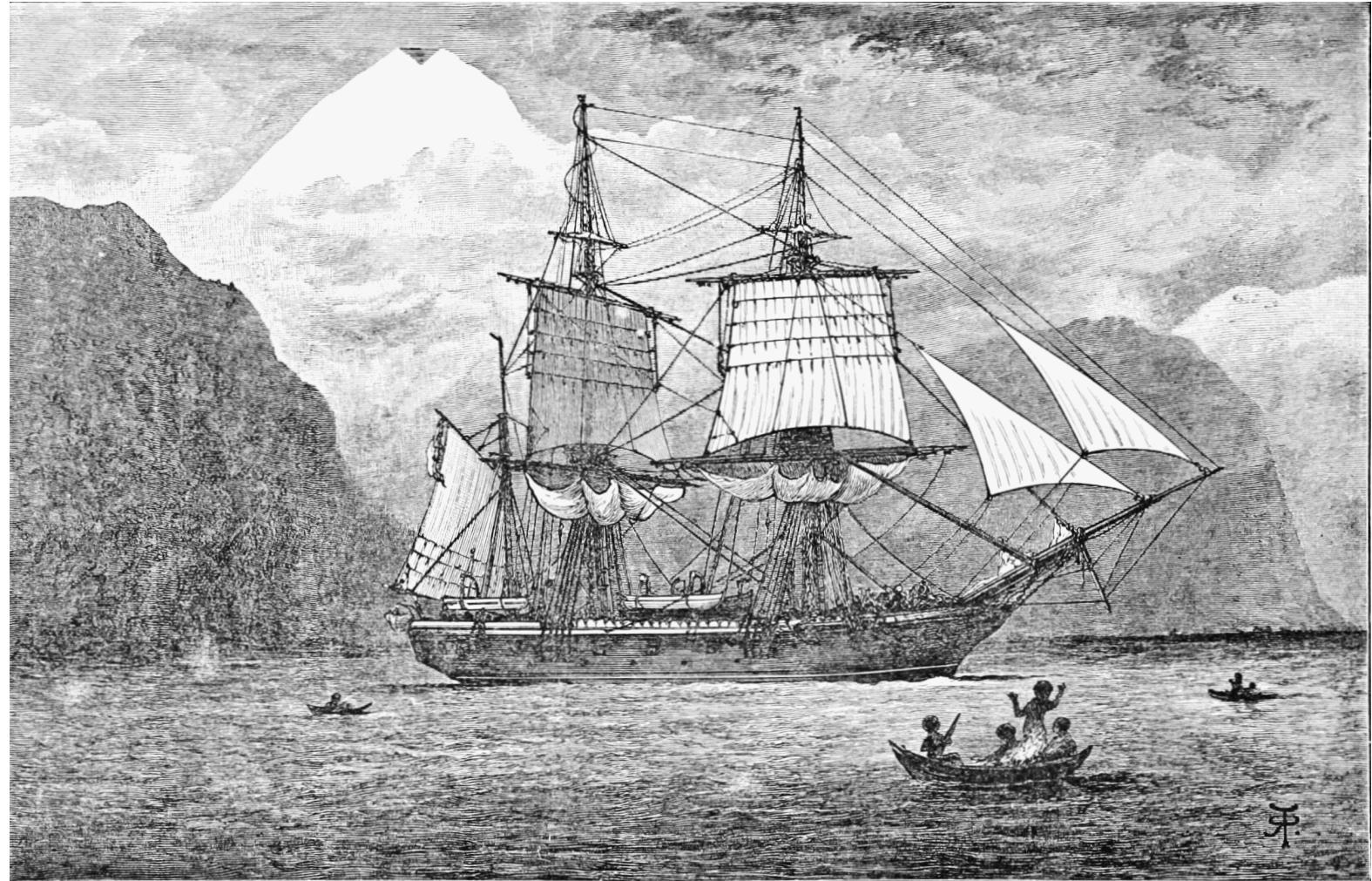
Charles Darwin



1809 - 1882

On the Origin of Species (1859)

Northeastern University



HMS Beagle

Voyage of the Beagle: 1831 – 1836

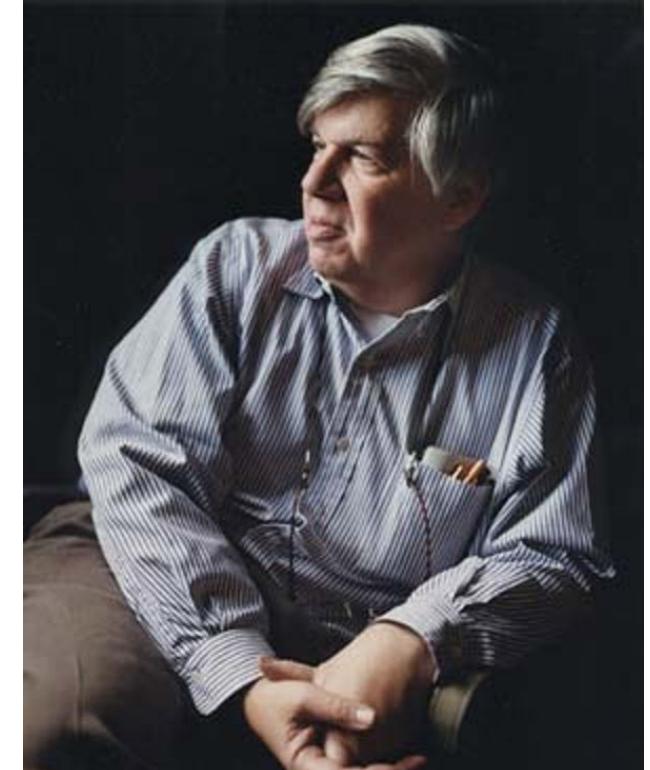


Natural Selection

[The] basis of natural selection is simplicity itself – two undeniable facts and an inescapable conclusion.

-- *Ever Since Darwin* (1977)

Stephen Jay Gould
Harvard Paleontologist
1941 - 2002



Variation and Inheritance

1. Organisms vary, and these variations are inherited (at least in part) by their offspring.



Population explosion

2. Organisms produce more offspring than can possibly survive.

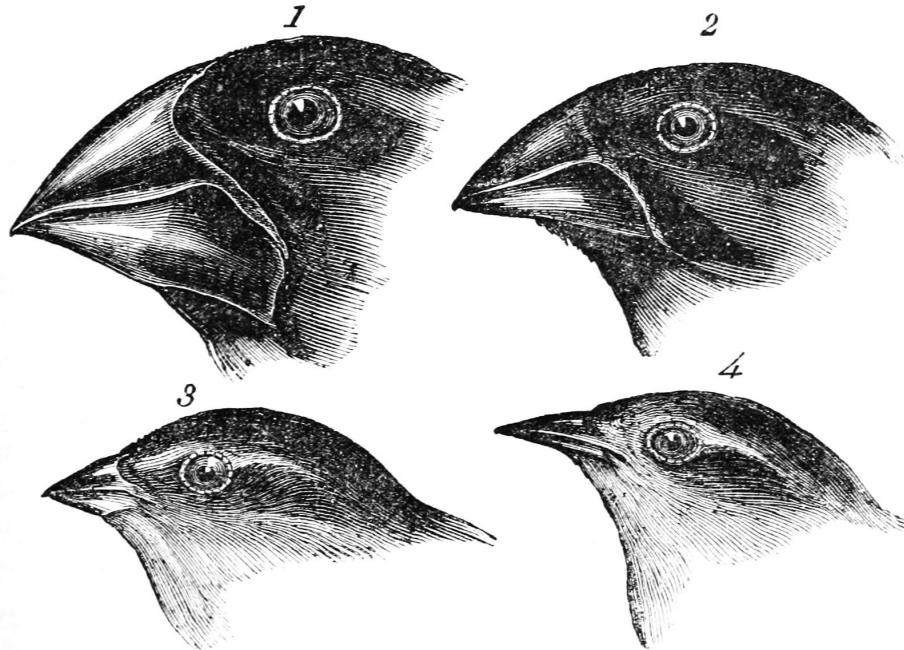


Survival of the Fittest

*3. On average,
offspring that vary
most strongly in
directions favored
by the environment
will survive and
propagate.*



Evolution by Natural Selection



1. *Geospiza magnirostris*.
3. *Geospiza parvula*.

2. *Geospiza fortis*.
4. *Certhidea olivacea*.

Survival of the
Fittest

Inheritance

*Natural
Selection*

Overpopulation

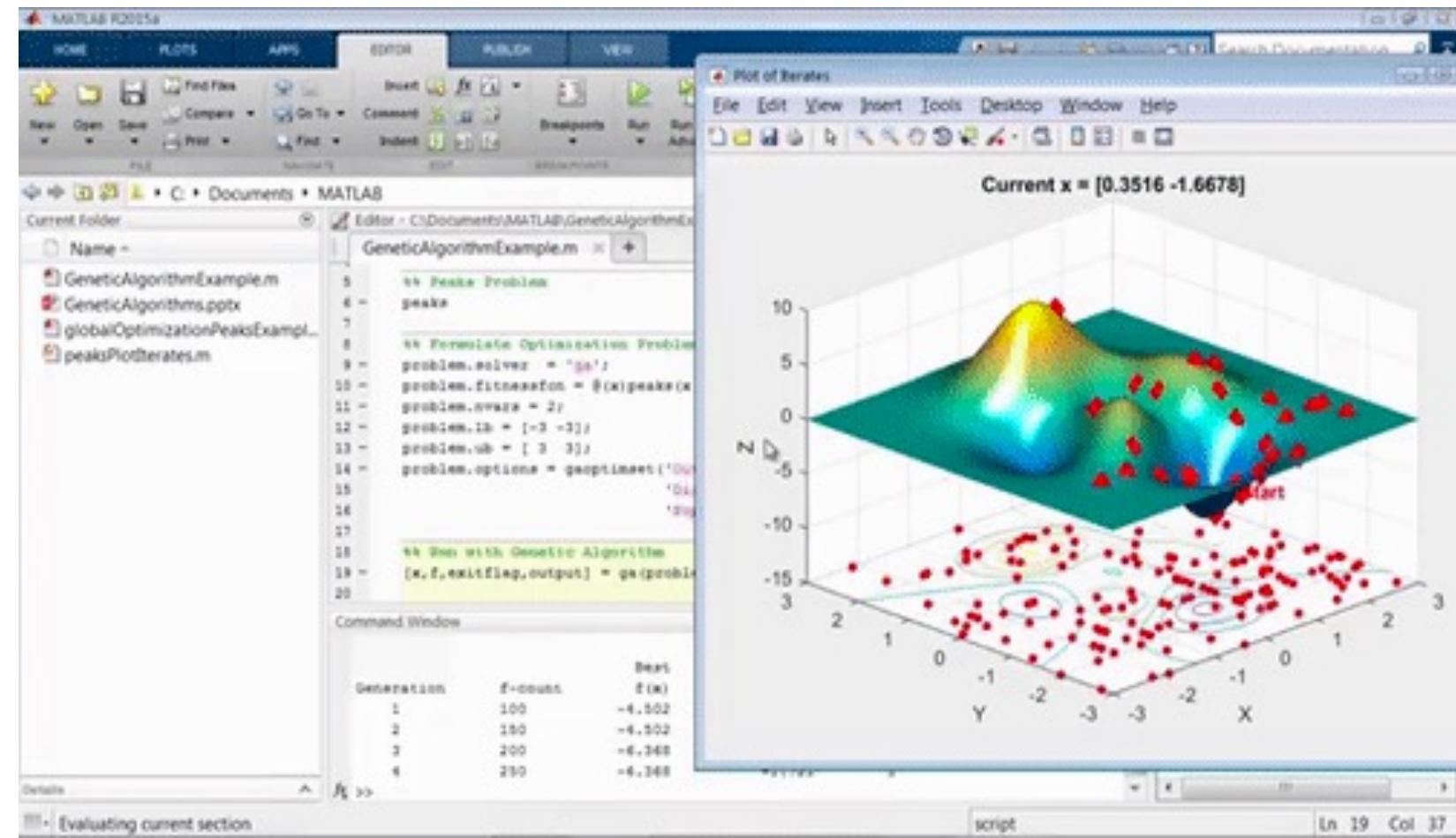
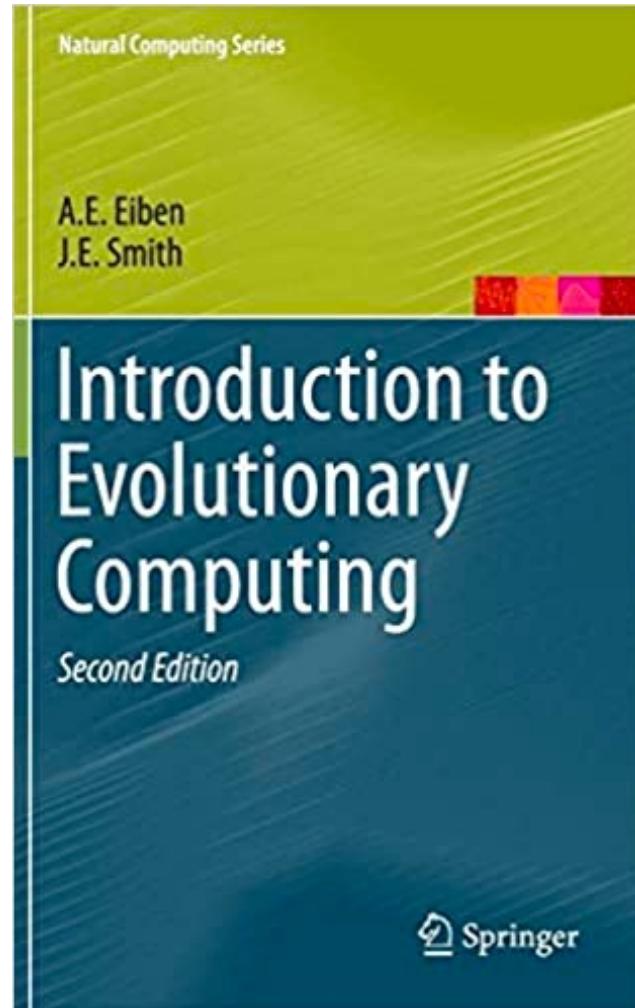


Biological systems are...

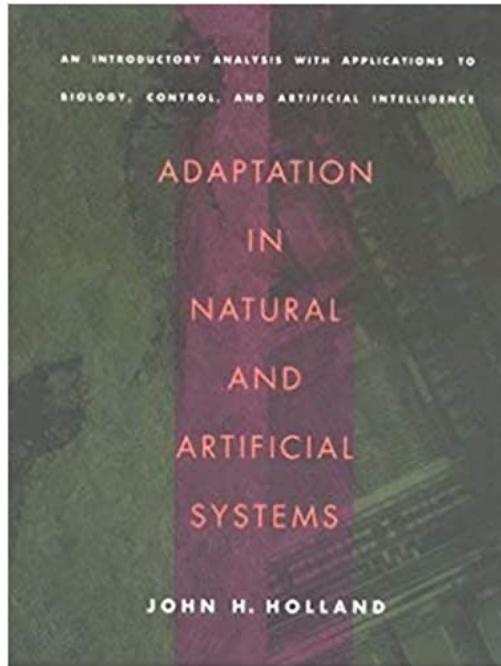
- Complex
- Diverse
- Adaptive
- Resilient
- Self-regulating (homeostatic)
- Synergistic (cooperative)
- Self-sustaining
- *CREATIVE!*



Algorithms that can evolve.



John Holland



© 1975

John Holland
(1929 – 2015)

Genetic Algorithms

Computer programs that “evolve” in ways that resemble natural selection can solve complex problems even their creators do not fully understand

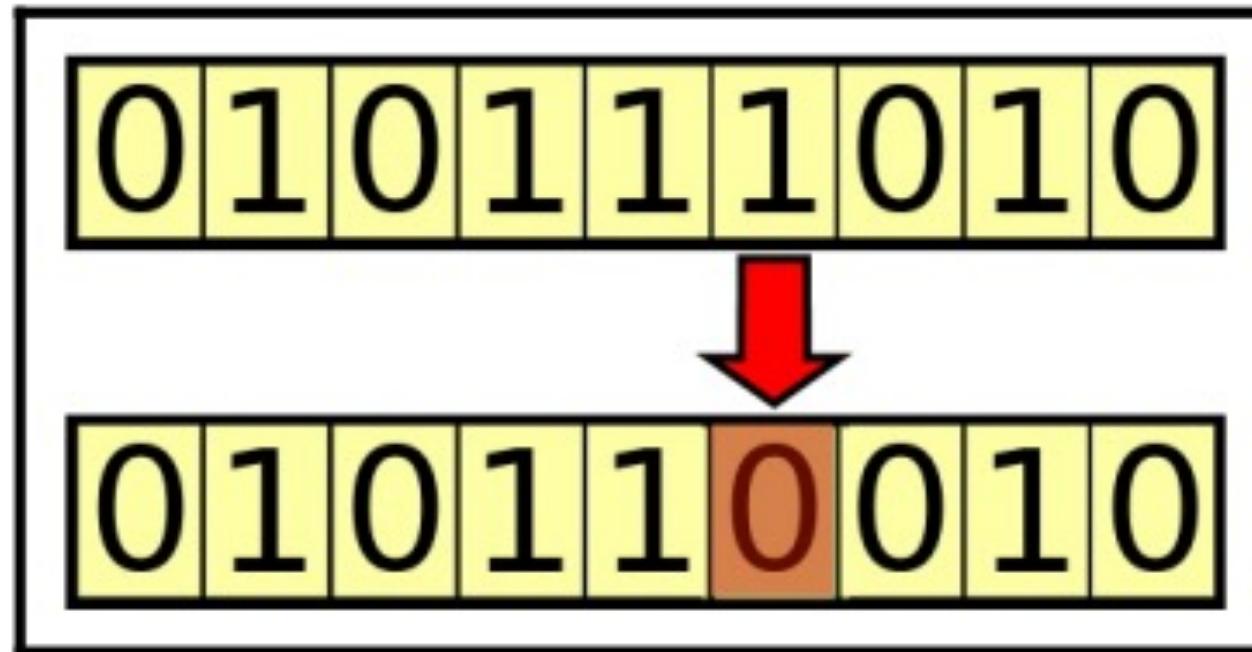
by John H. Holland

Living organisms are consummate problem solvers. They exhibit a versatility that puts the best computer programs to shame. This observation is especially galling for computer scientists, who may spend months or years of intellectual effort on an algorithm, whereas organisms come by their abilities through the apparently undirected mechanism of evolution and natural selection.



Northeastern University

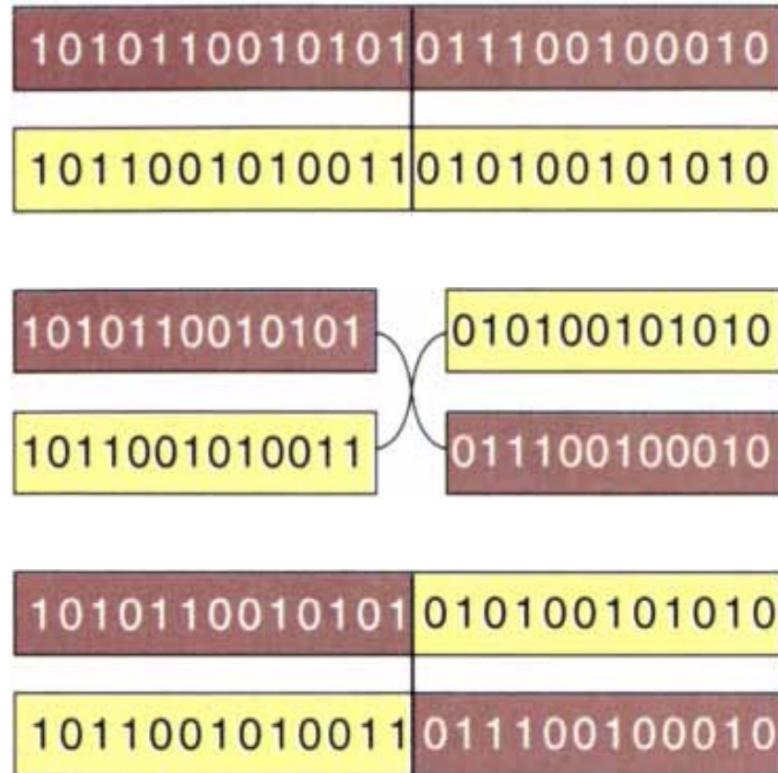
Mutation



Mutation operator applied to a binary-coded chromosome



Crossover



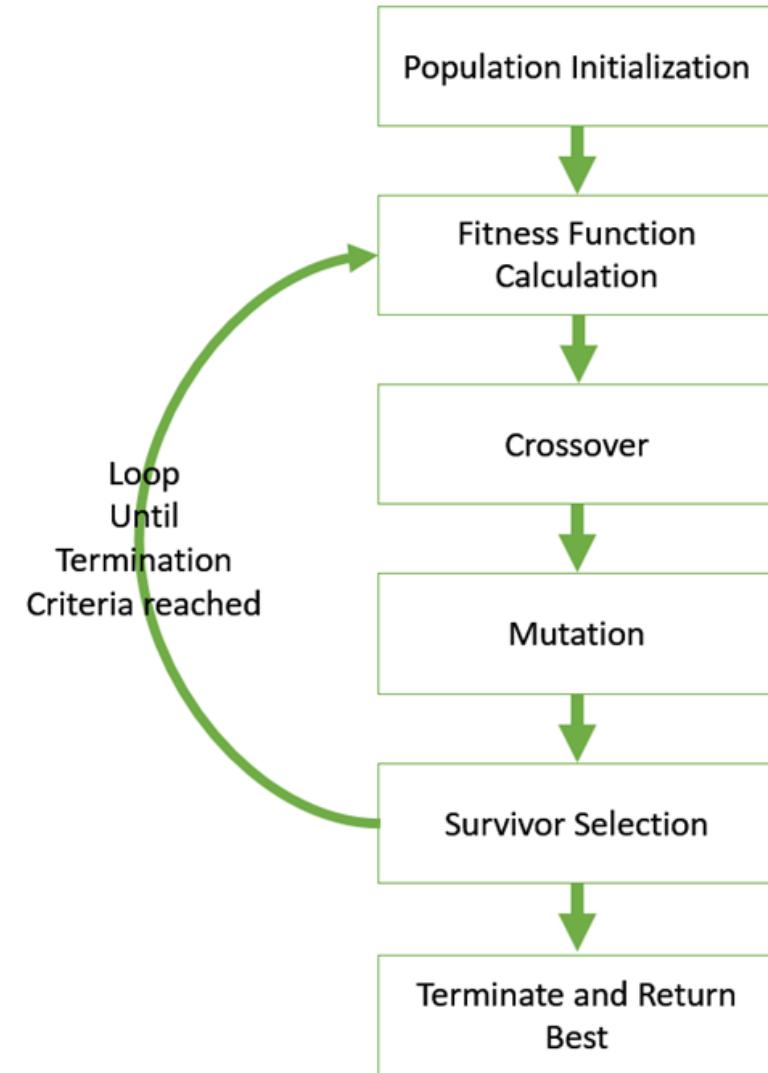
CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.



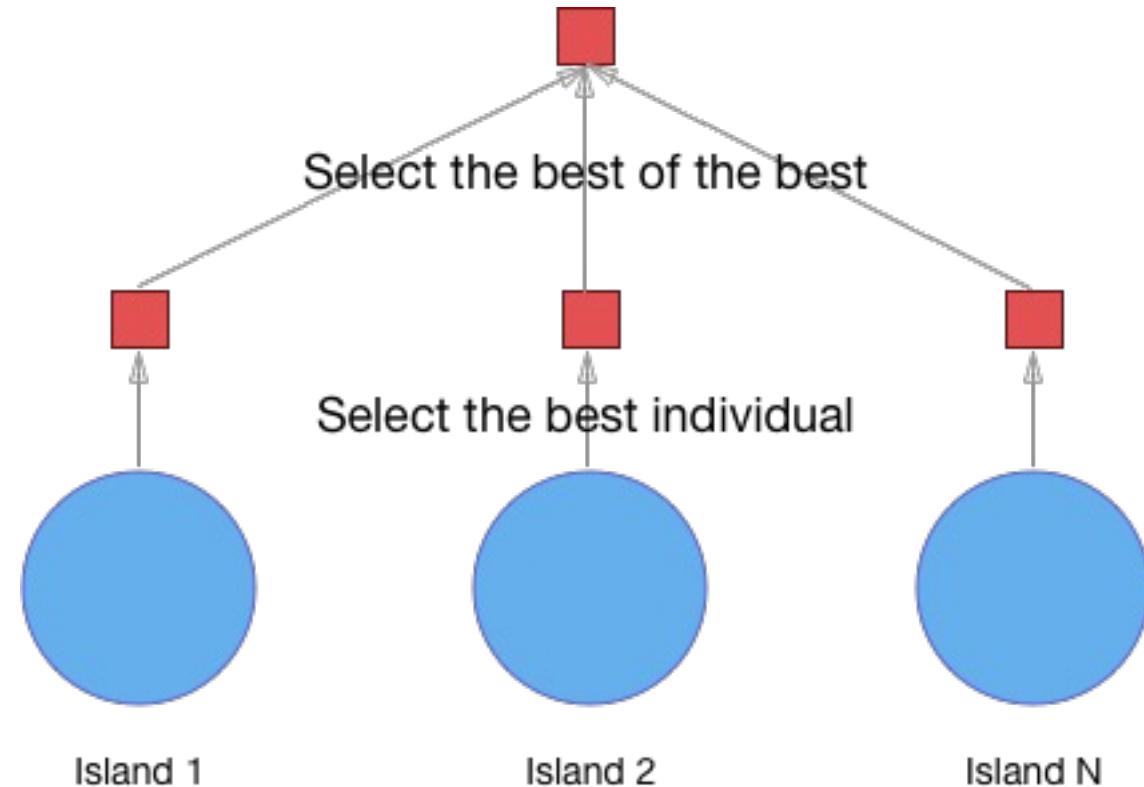
The typical flow of a GA

- Solutions are encoded as bit strings
- Binary strings are mapped to a particular domain-specific solution and evaluated according to the problem objectives.
- Solutions produce offspring through mutation and crossover
- Good solutions survive, bad solutions die.
- Repeat. The population *evolves* towards an optimal results



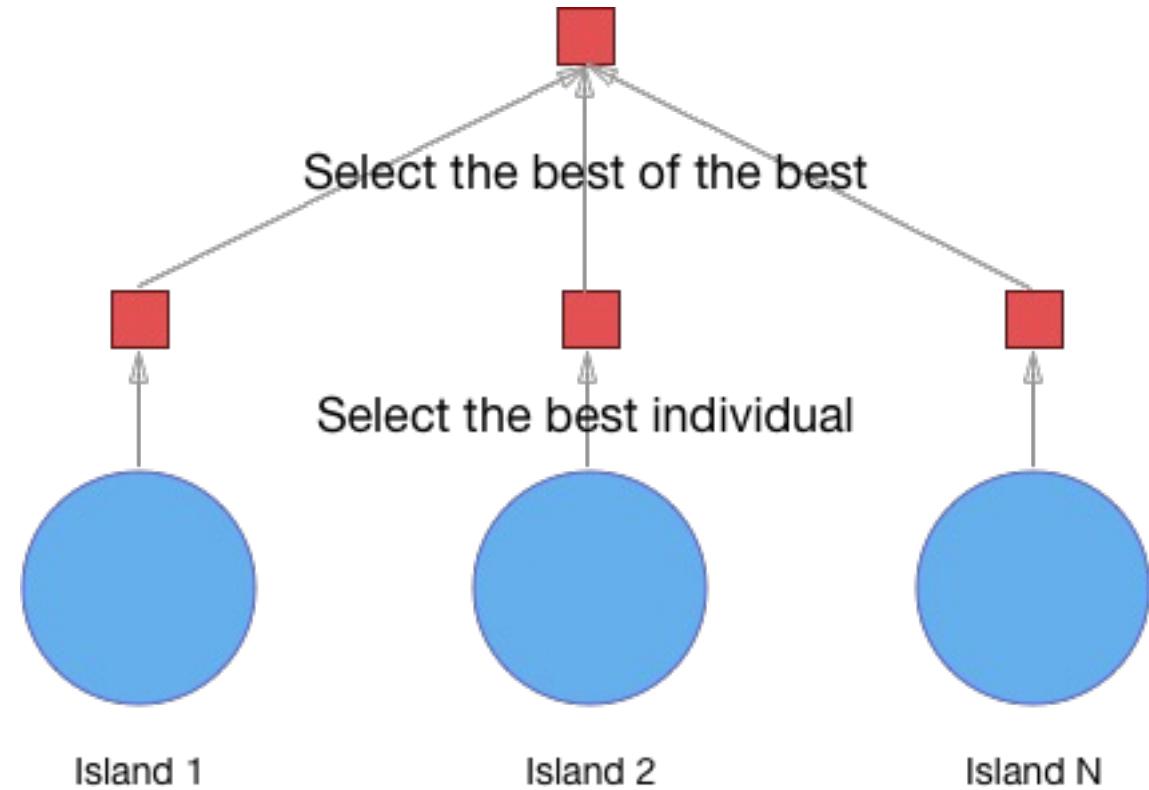
Advantages of GAs / Evolutionary Computing

- 1. Ability to find global optima:** Cross-over provides a mechanism for escaping local optima.
- 2. Adaptability:** GAs are domain-neutral. Can solve a broad range of optimization problems.
- 3. Scalability:** GA's can be parallelized to enable agents to run across multiple cores, multiple servers or other HPC environments.

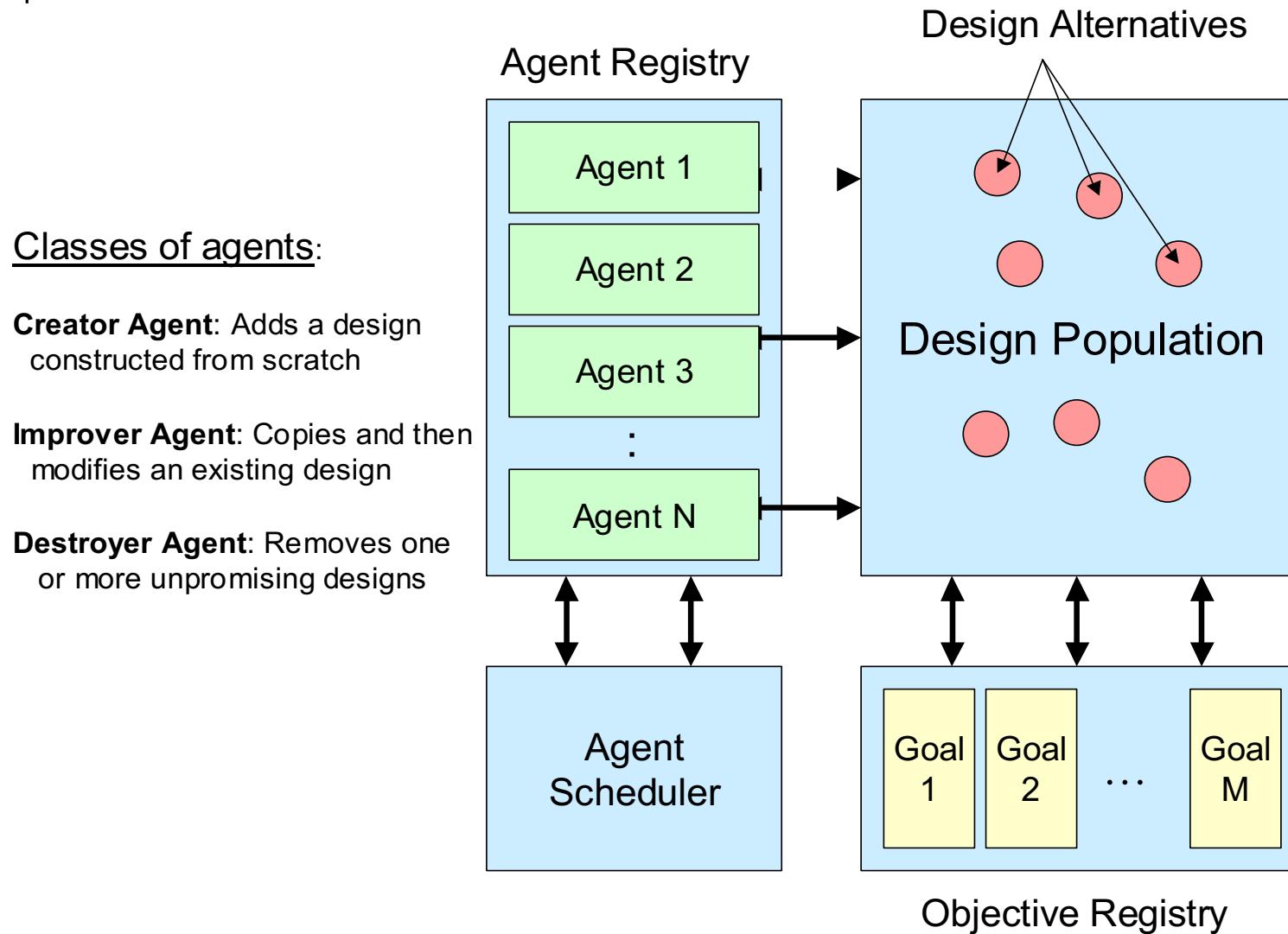


Disadvantages of GAs / Evolutionary Computing

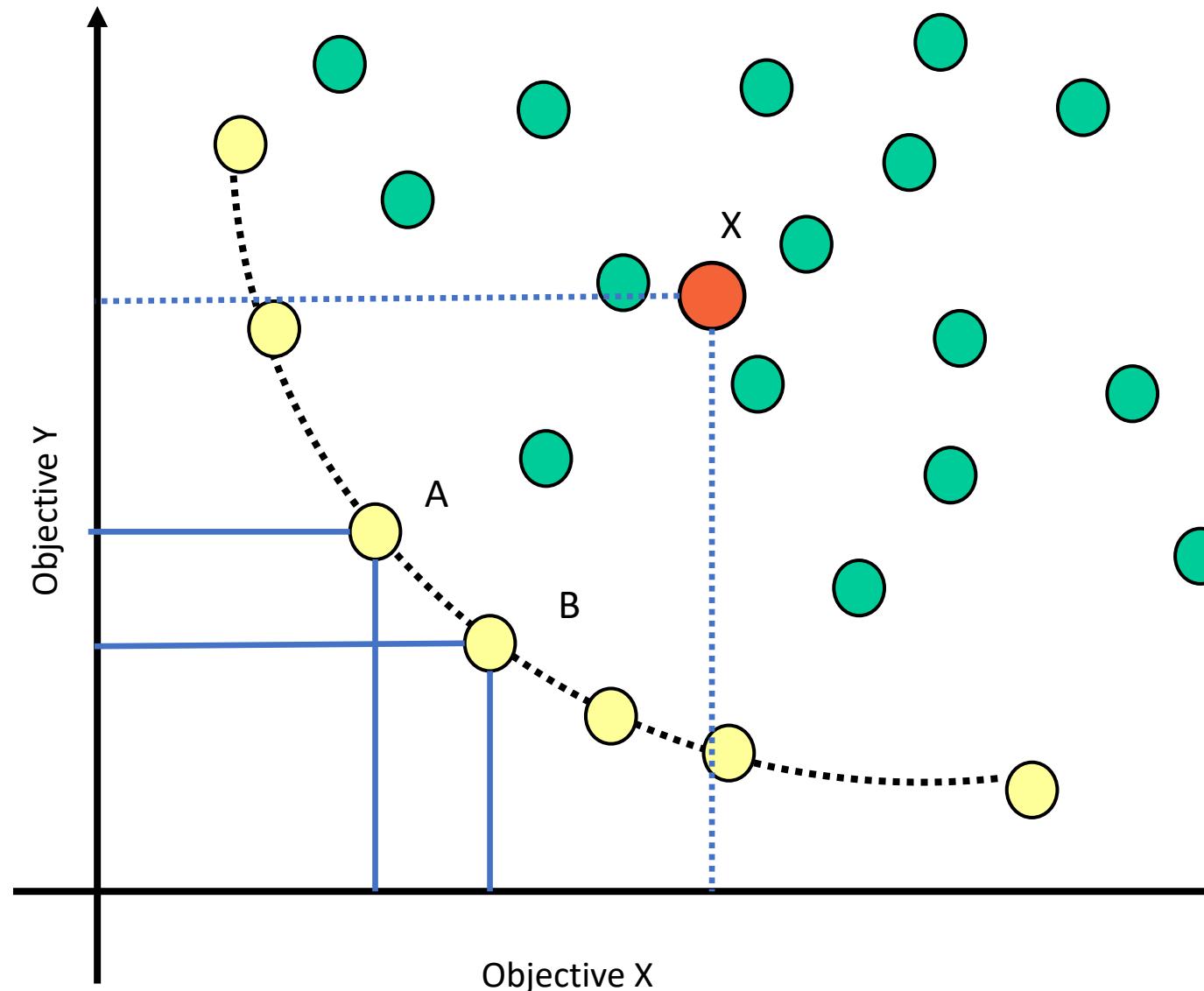
1. **Slow / Compute Intensive:** Requires many iterations to find the best solutions.
2. **Lack of transparency:** Difficult to interpret how a GA is arriving at its solution.
3. **Reproducibility:** Different runs on the same problem instance may produce different results.
4. **The encoding problem:** Mapping from binary strings to a domain-specific solution may be difficult, making it hard to evaluate a given solution string.



An architecture for **multi-objective** optimization using evolutionary computing



Evolving Tradeoffs

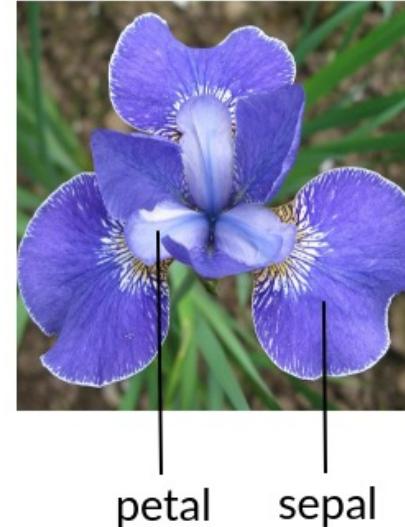


Euclidean Distance on Iris Dataset

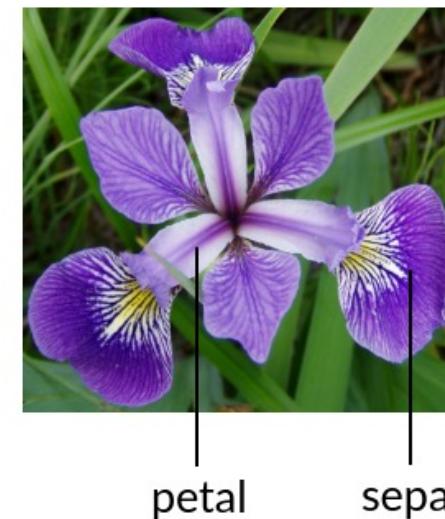
$$\text{diff} = \sqrt{(\Delta Slen)^2 + (\Delta Swid)^2 + (\Delta Plen)^2 + (\Delta Pwid)^2}$$

Slen = Sepal Length
Swid = Sepal Width
Plen = Petal Length
Pwid = Petal Width

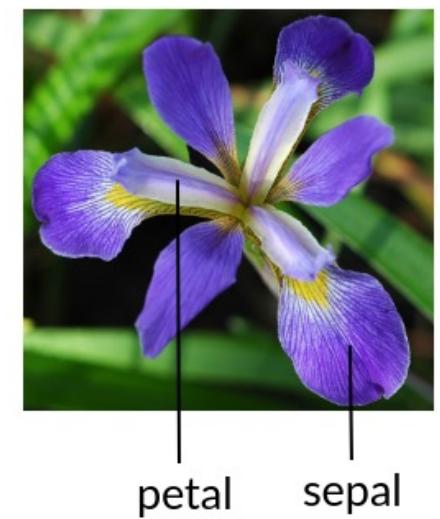
iris setosa



iris versicolor



iris virginica



Distance metrics

Some frequently used distance functions.

Camberra :

$$d(x, y) = \sum_{i=1}^m \frac{|x_i - y_i|}{|x_i + y_i|} \quad (2)$$

Minkowsky :

$$d(x, y) = \left(\sum_{i=1}^m |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (3)$$

Chebychev :

$$d(x, y) = \max_{i=1}^m |x_i - y_i| \quad (4)$$

Euclidean :

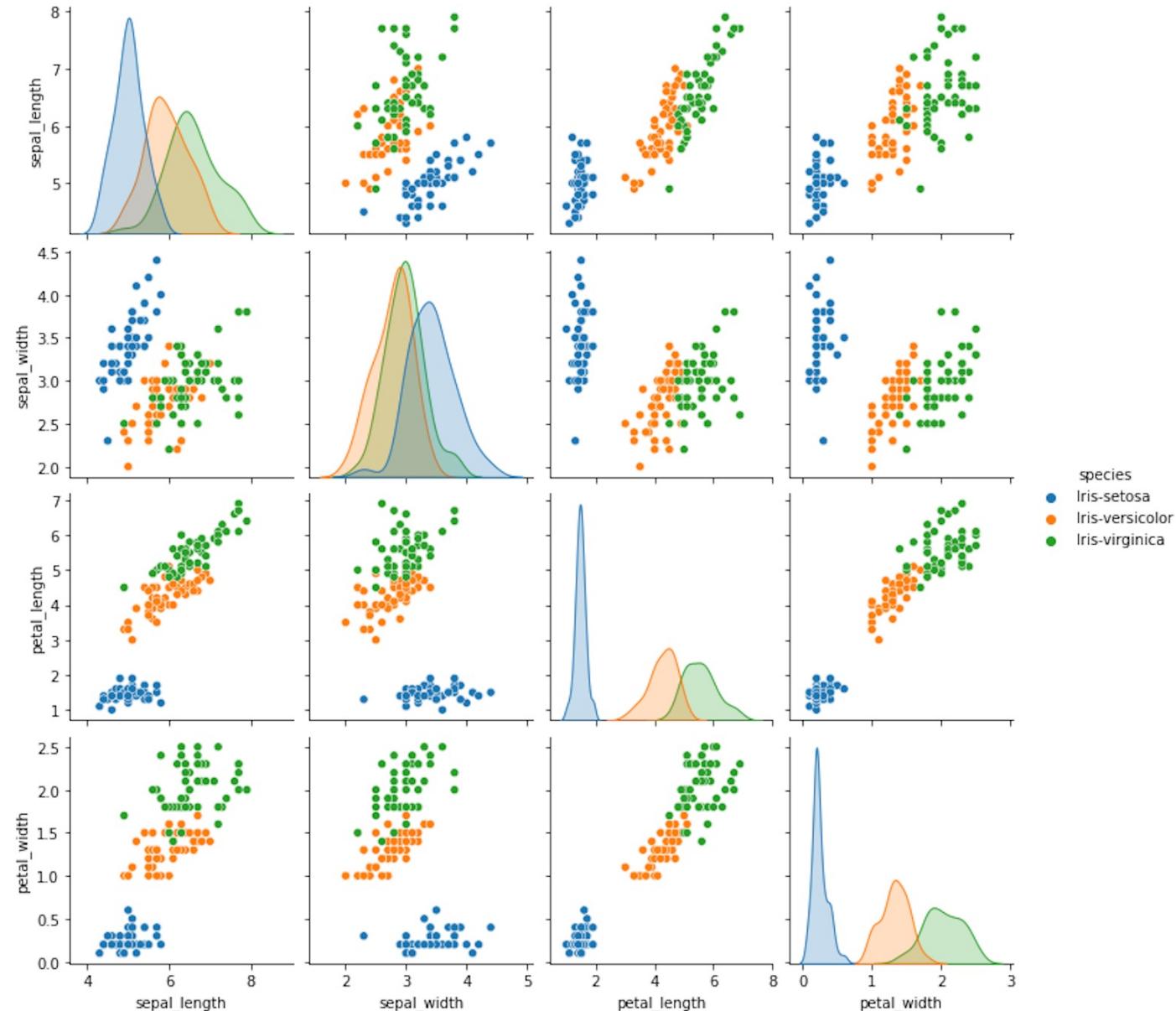
$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (5)$$

Manhattan / city - block :

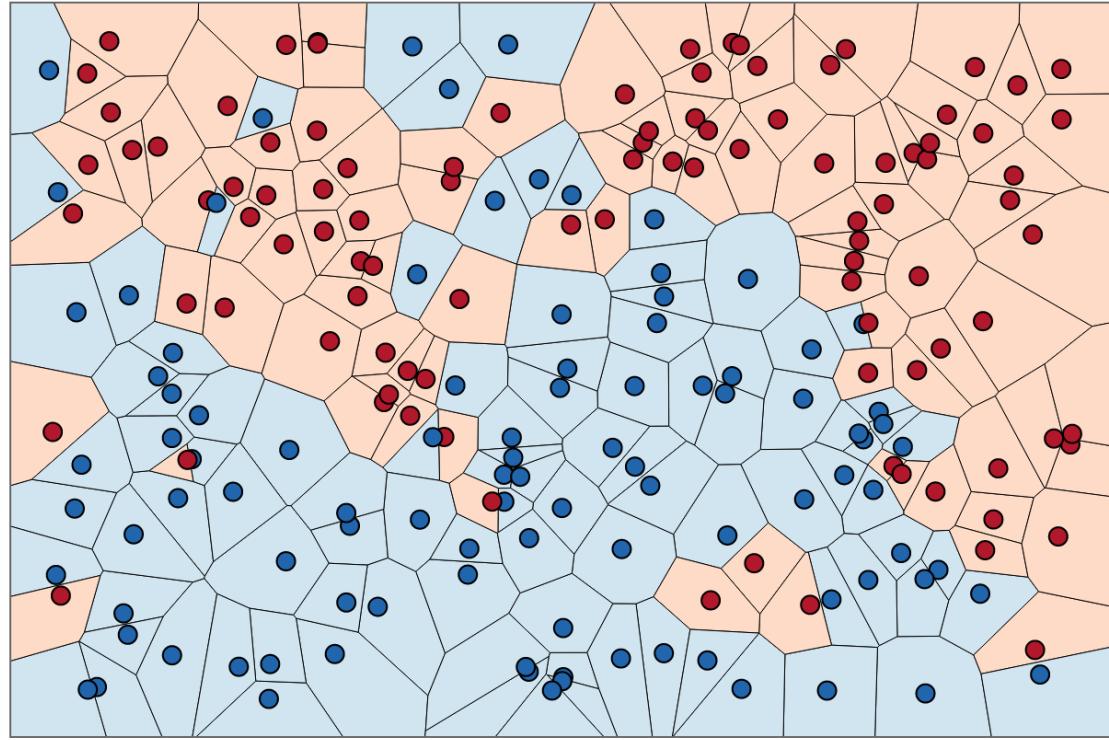
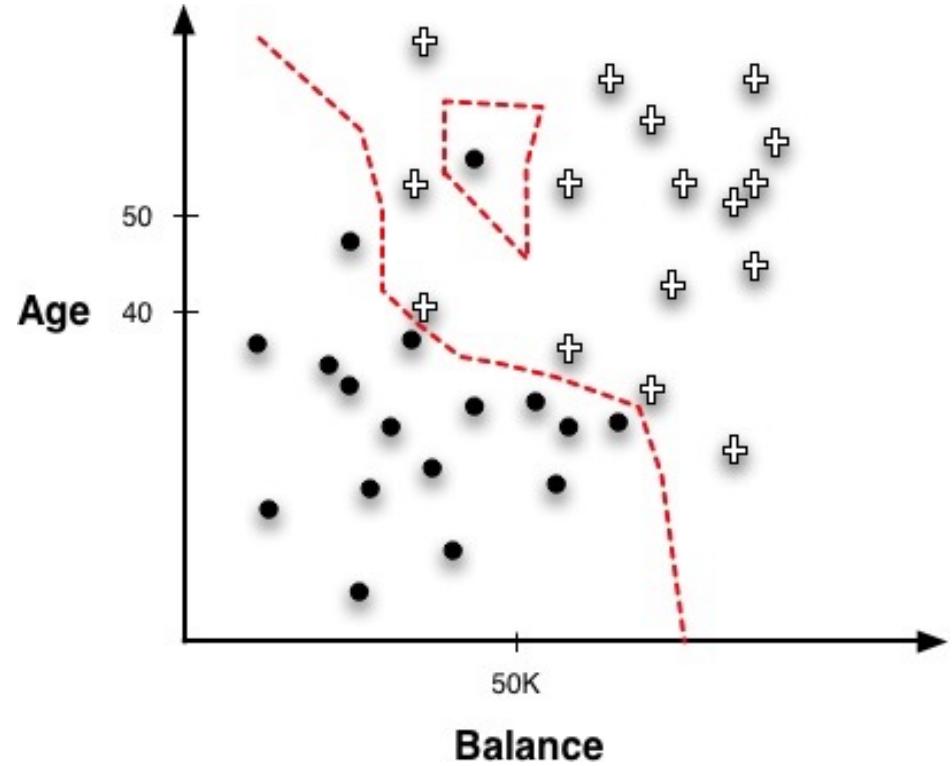
$$d(x, y) = \sum_{i=1}^m |x_i - y_i| \quad (6)$$



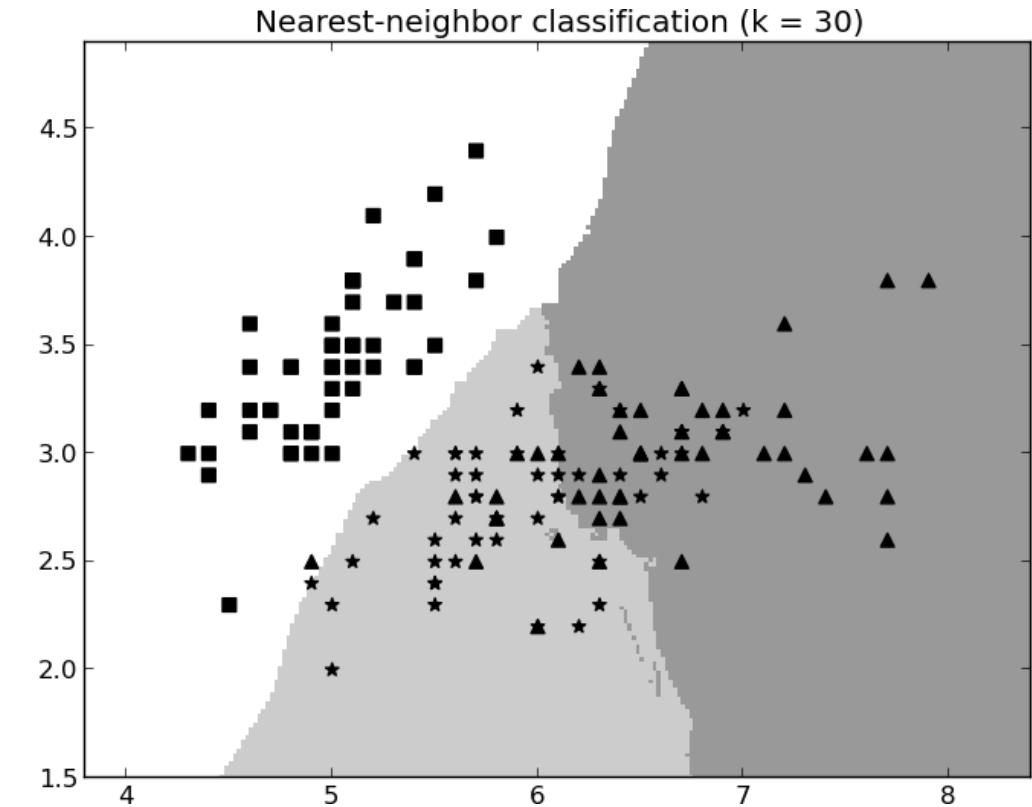
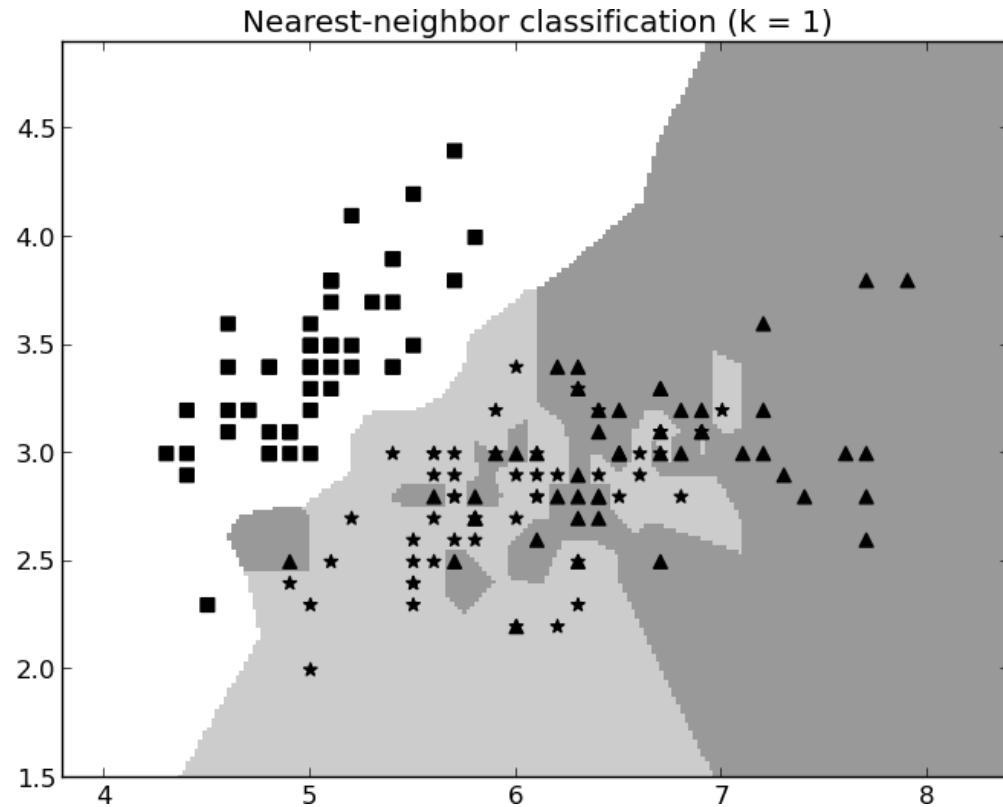
Why 1-NN makes mistakes on the iris dataset



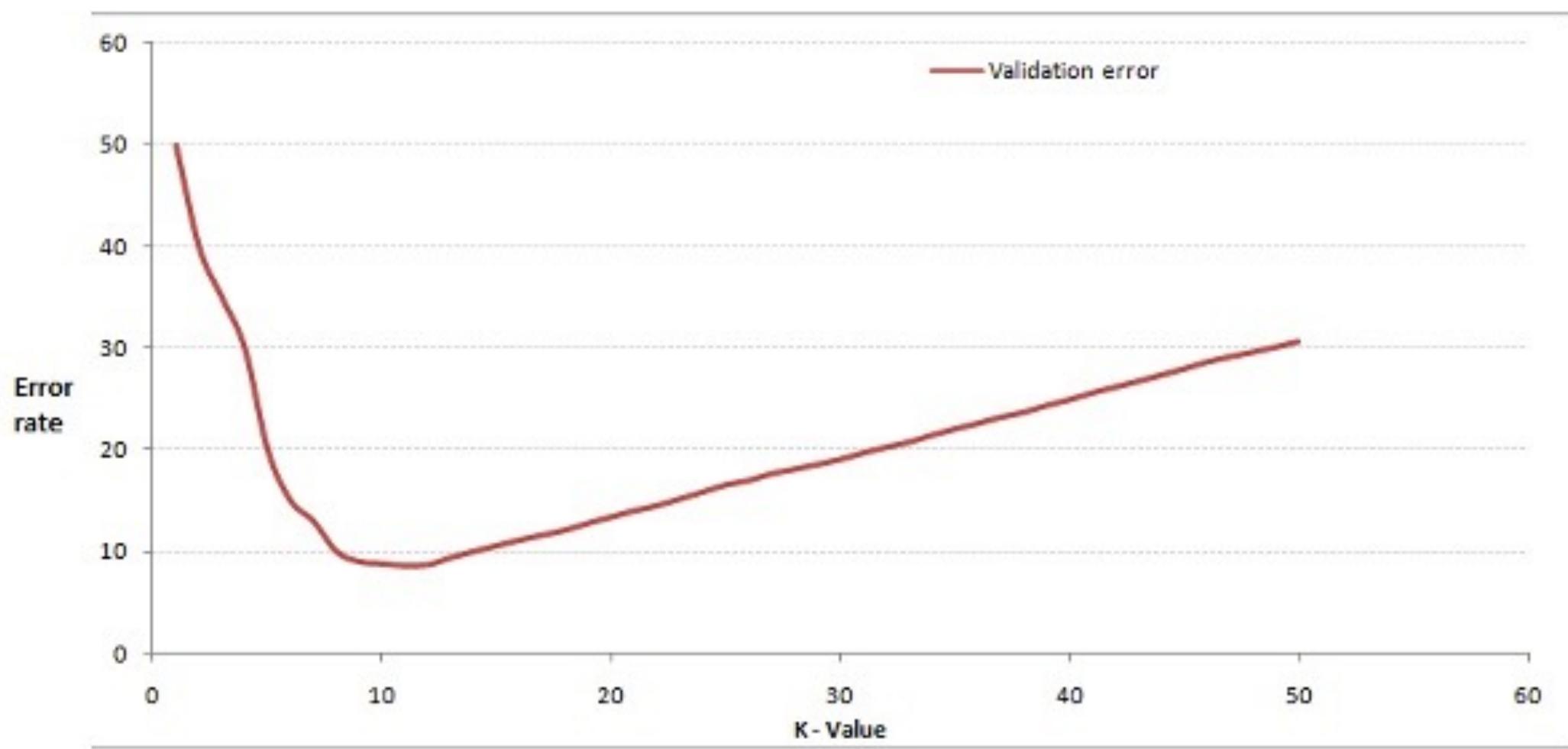
Overfitting



Justifying the k-value

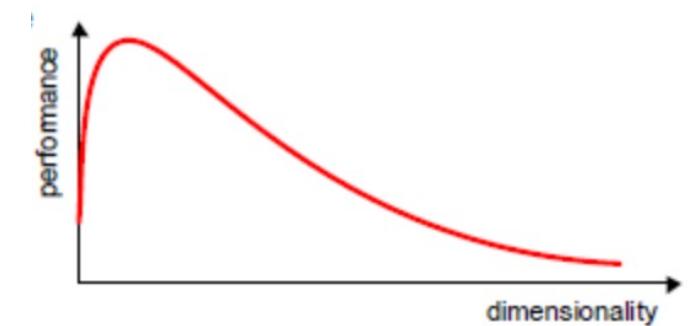
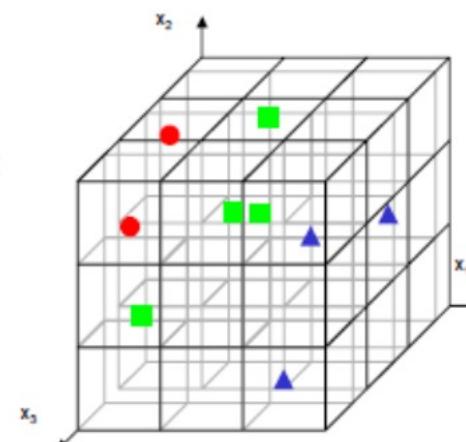
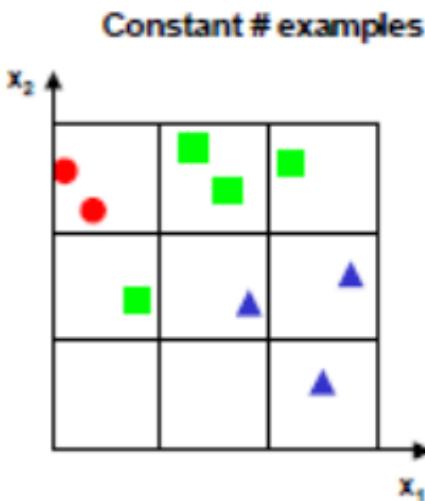
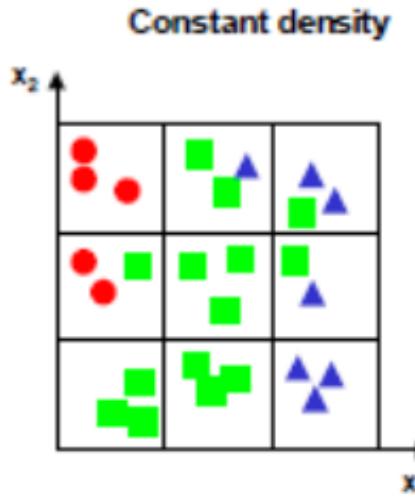


Choosing K



Curse of Dimensionality

Since all attributes contribute to the distance calculations, instance similarity can be confused and misled by the presence of too many irrelevant attributes.



General Distance Measure

$$\text{diff} = w_1 |\Delta A_1|^r + w_2 |\Delta A_2|^r + \dots + w_n |\Delta A_n|^r$$

where

ΔA_i is the difference with respect to feature i,

w_k is a feature weighting factor (0.0 to 1.0), and

r is an exponent (> 0.0)



Hyper-parameter tuning: Grid Search

Model Parameters: Parameters or coefficients that *define the model.*
e.g., coefficients for linear regression, weights for the perceptron model, etc.

Hyper Parameters: Parameters that define the architecture of the model or that control or limit our search space.

- Degree of a polynomial
- # of neighbors in k-Nearest Neighbors
- # of clusters in k-Means clustering
- Maximum depth of a decision tree
- Architecture of a neural network



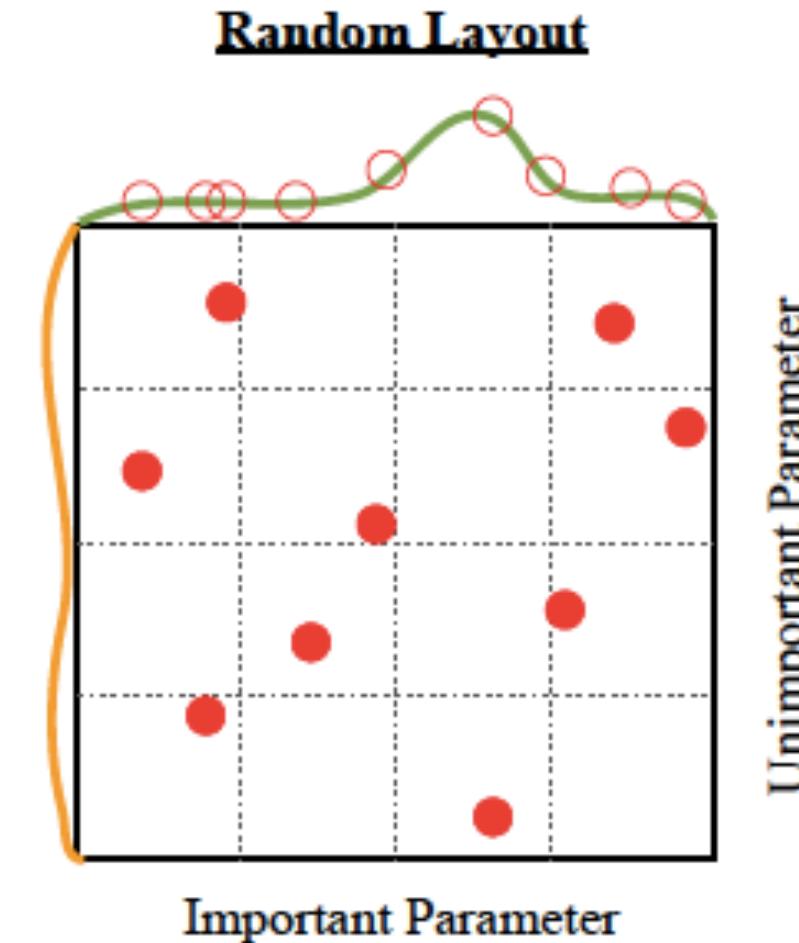
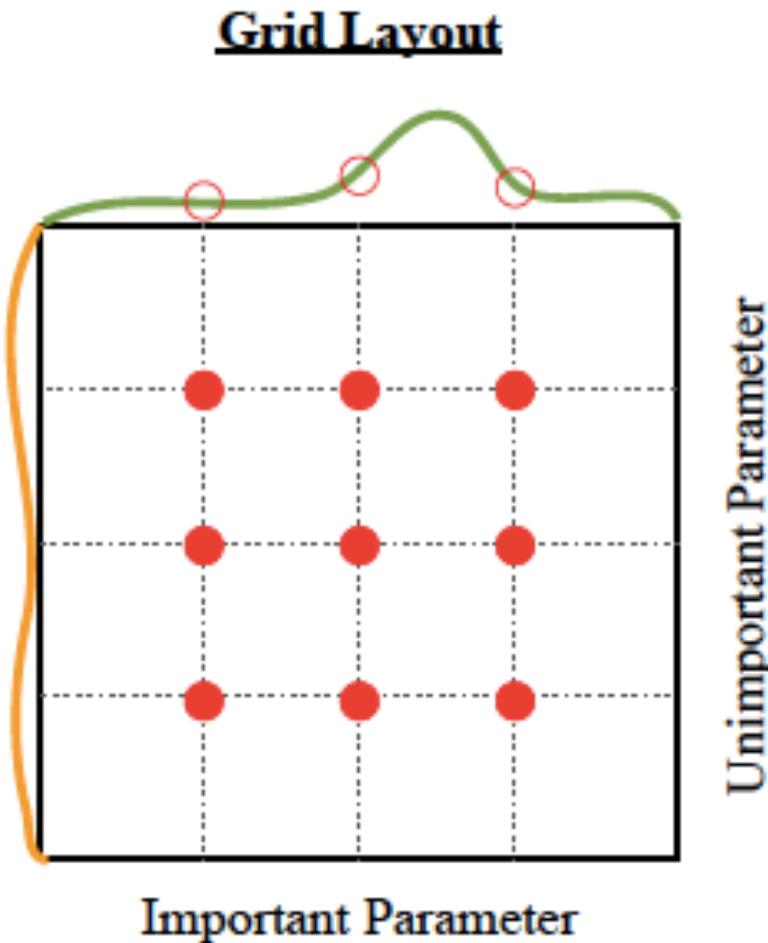
Hyper-parameter tuning in Model Development

In general, this process includes:

1. Define a model
2. Define the range of possible values for all hyperparameters
3. Define a method for sampling hyperparameter values
4. Define an evaluative criteria to judge the model
5. Define a cross-validation method



Hyperparameter Tuning: Parameter Sampling



Hyper-parameter tuning: Evolve optimal *tradeoffs*

$$\text{diff} = w_1 |\Delta A_1|^r + w_2 |\Delta A_2|^r + \dots + w_n |\Delta A_n|^r$$

where

ΔA_i is the difference with respect to feature i,

w_k is a feature weighting factor (0.0 to 1.0), and

r is an exponent (> 0.0)

A solution consists of a specification of the distance metric parameters:
 $w_1 \dots w_n$, r, k (# of nearest neighbors). So, **n+2** parameters altogether.

We *evaluate* the solution by testing it against a dataset and finding multiple measures of accuracy and associated tradeoffs.



Model Optimization: The objective-function approach

Weight the relative importance of different model attributes (complexity, accuracy scores, etc.) to derive a net *score*.

$$\begin{aligned} \text{Score(Model)} = & w_1 \text{accuracy}_1 + w_2 \text{accuracy}_2 + \dots \\ & \dots + w_j \text{complexity}_1 + w_k \text{complexity}_2 + \dots \end{aligned}$$

Note: This is exactly how we envisioned the accuracy-complexity tradeoff with regularization. (Assign a weight, λ , to the regularization term.)



L1-Regularization

Recall the linear regression objective:

$$\min_{\mathbf{w}, b} \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{w}, b}(\mathbf{x}_i) - y_i)^2.$$

An L1-regularized objective looks like this:

$$\min_{\mathbf{w}, b} \left[\lambda |\mathbf{w}| + \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{w}, b}(\mathbf{x}_i) - y_i)^2 \right],$$

where $|\mathbf{w}| \stackrel{\text{def}}{=} \sum_{j=1}^D |w^{(j)}|$ and λ is a hyperparameter

Here, $\mathbf{w} = \beta_1, \beta_2, \beta_3, \beta_4, \dots$
 $b = \beta_0$



L2-Regularization

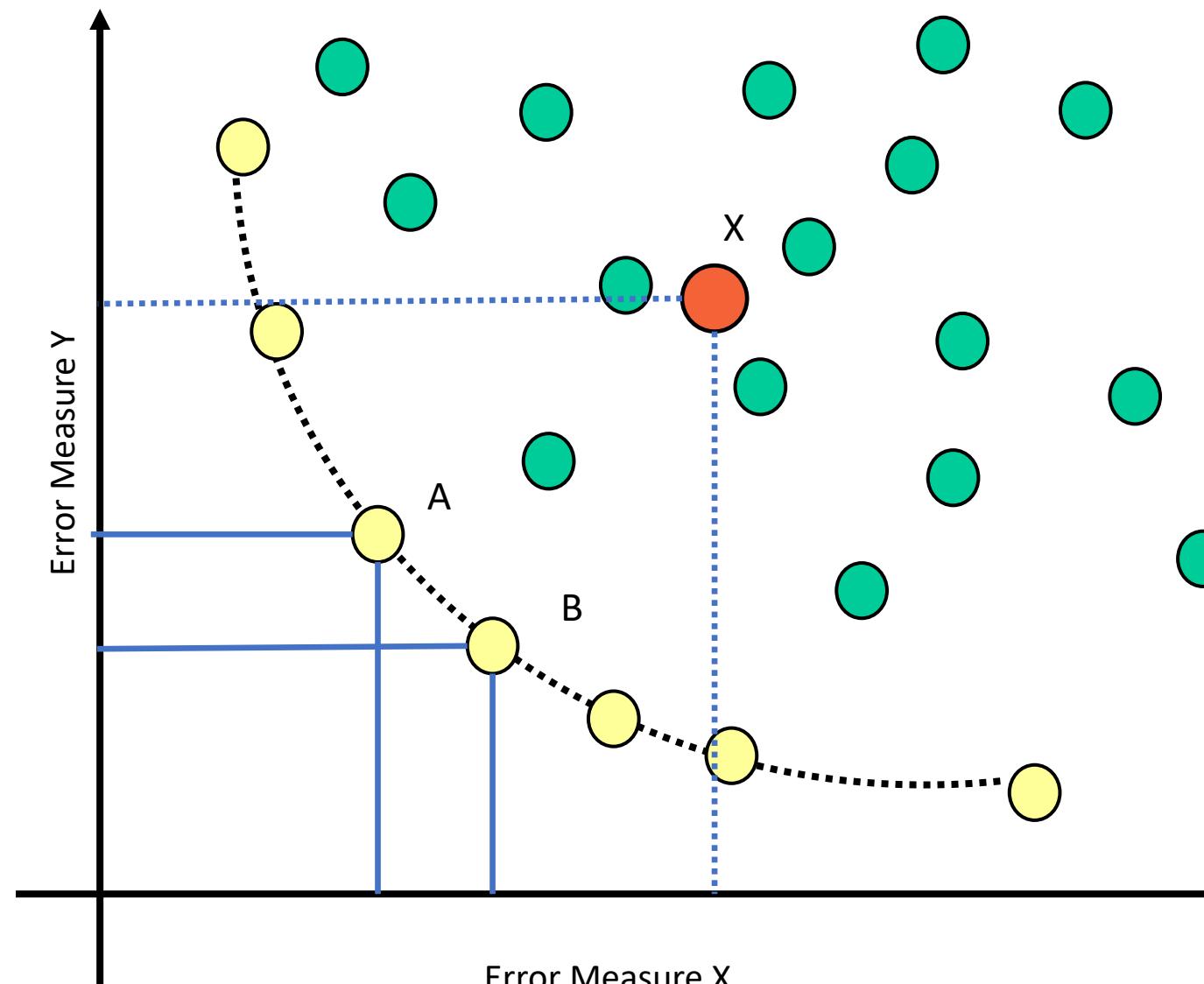
An L2-regularized objective looks like this:

$$\min_{\mathbf{w}, b} \left[\lambda \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{w}, b}(\mathbf{x}_i) - y_i)^2 \right]$$

Here, $\mathbf{w} = \beta_1, \beta_2, \beta_3, \beta_4, \dots$
 $b = \beta_0$



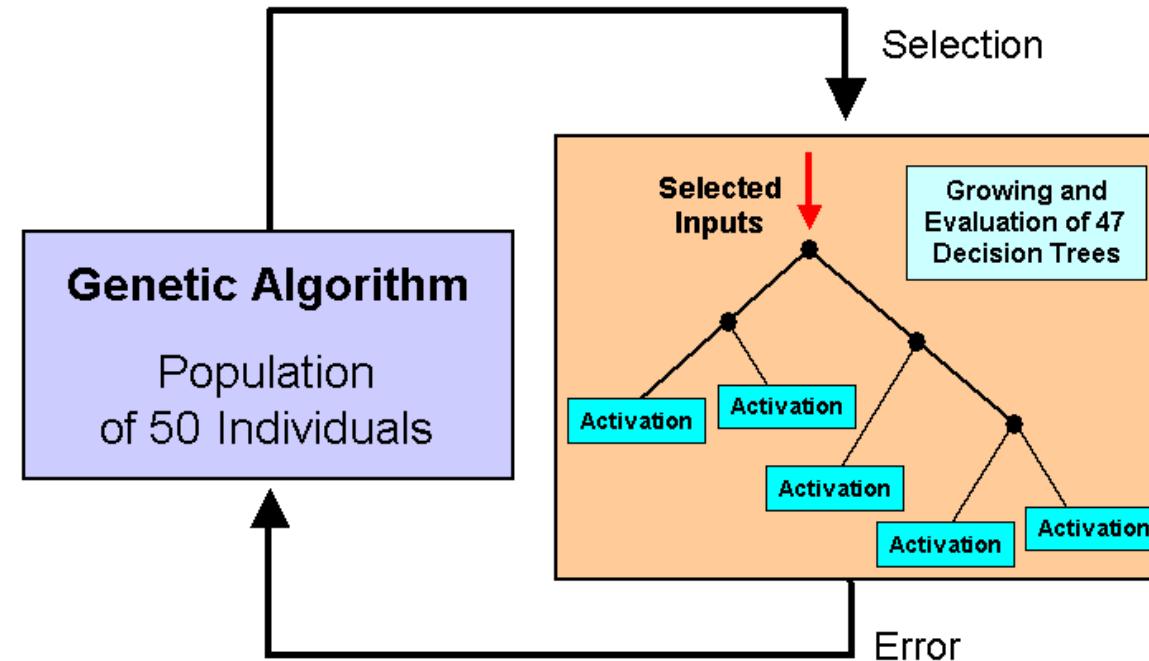
Tradeoffs Analysis (Errors are minimized)



Evolving Decision Trees

Koza J.R., (1993). *Evolving Decision Trees*. Proceedings of the 5th International conference on Genetic Algorithms.

Sakamoto Y., Yamamoto K. and Ishibuchi T., (2005). *Multiobjective Evolutionary Algorithms for Constructing Compact and Accurate Decision Trees*. IEEE Transactions on Evolutionary Computation.



Evolutionary Deep Learning



Contents lists available at [ScienceDirect](#)

Neurocomputing

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



Survey paper

Evolutionary deep learning: A survey

Zhi-Hui Zhan ^{a,*}, Jian-Yu Li ^a, Jun Zhang ^b



ABSTRACT

As an advanced artificial intelligence technique for solving learning problems, deep learning (DL) has achieved great success in many real-world applications and attracted increasing attention in recent years. However, as the performance of DL depends on many factors such as the architecture and hyperparameters, how to optimize DL has become a hot research topic in the field of DL and artificial intelligence. Evolutionary computation (EC), including evolutionary algorithm and swarm intelligence, is a kind of efficient and intelligent optimization methodology inspired by the mechanisms of biological evolution and behaviors of swarm organisms. Therefore, a large number of researches have proposed EC algorithms to optimize DL, so called evolutionary deep learning (EDL), which have obtained promising results. Given the great progress and rapid development of EDL in recent years, it is quite necessary to review these developments in order to summarize previous research experiences and knowledge, as well as provide references to benefit the development of more researches and applications. For this aim, this paper categorizes existing works in a two-level taxonomy. The higher level includes four categories based on *when* the EC can be adopted in optimizing the DL, which are the four procedures of the whole DL lifetime, including *data processing*, *model search*, *model training*, and *model evaluation and utilization*. In the lower level, related works in each category are further classified according to the functionality and the aim of using EC in the corresponding DL procedure, i.e., *why* using EC in this DL procedure. As a result, the taxonomy can clearly show *how* an EC algorithm can be used to optimize and improve DL. Moreover, this survey also discusses the potential research directions to provide the prospect of EDL in the future.

© 2022 Published by Elsevier B.V.

Zhan, Li, and Zhang (2022).
Evolutionary Deep Learning: A Survey. *Neurocomputing* **483**:42-58.



Northeastern University

Tweaking / modifying a solution

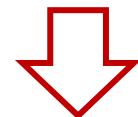
weights

r

k

1.0	1.0	1.0	1.0	2	1
-----	-----	-----	-----	---	---

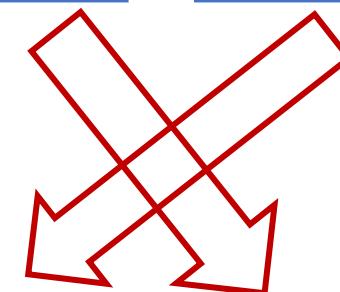
Mutate:



1.0	1.0	0.95	1.0	2	1
-----	-----	------	-----	---	---

0.20	0.90	1.00	0.45	2	5	0.38	0.29	1.00	0.93	1.5	3
------	------	------	------	---	---	------	------	------	------	-----	---

Crossover:



0.20	0.90	1.00	0.93	1.5	3	0.38	0.29	1.00	0.45	2	5
------	------	------	------	-----	---	------	------	------	------	---	---



Predicting recurrence of breast cancer

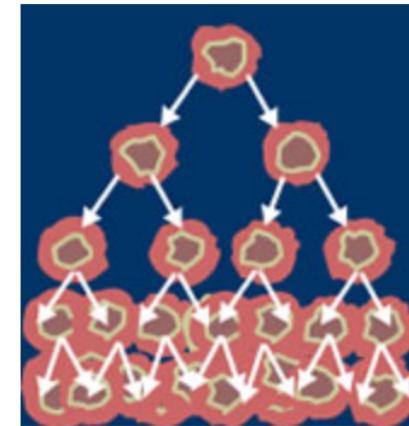


Breast Cancer Wisconsin (Prognostic) Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Prognostic Wisconsin Breast Cancer Database

R=Recur, N=Nonrecur



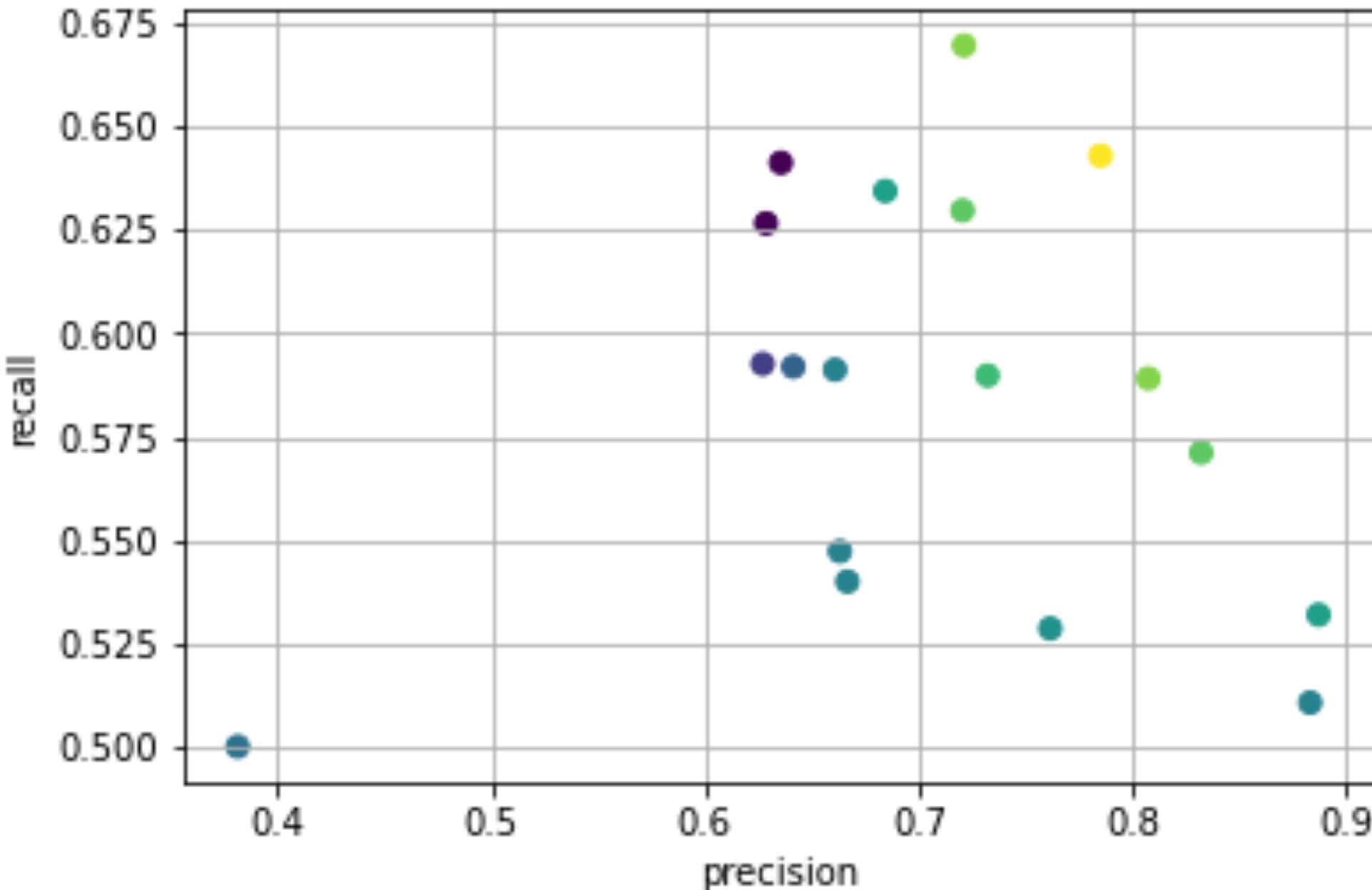
Data Set Characteristics:	Multivariate	Number of Instances:	198	Area:	Life
Attribute Characteristics:	Real	Number of Attributes:	34	Date Donated	1995-12-01
Associated Tasks:	Classification, Regression	Missing Values?	Yes	Number of Web Hits:	224238



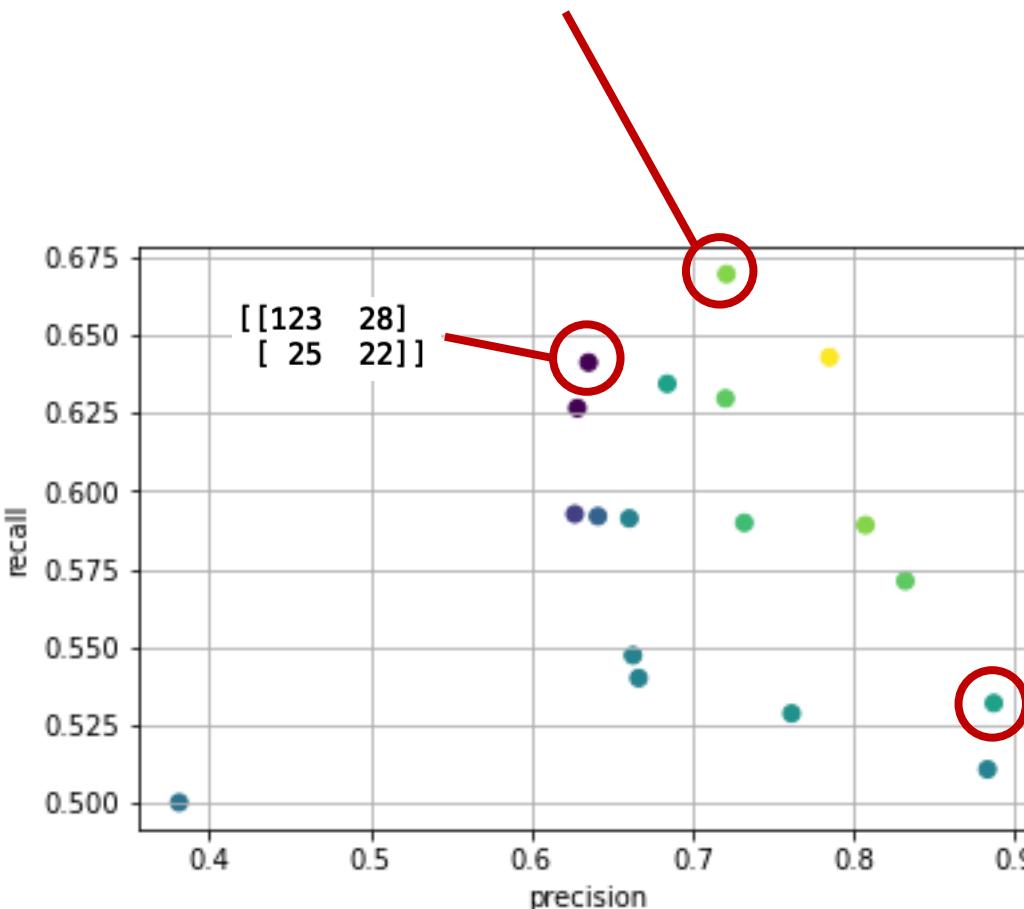
Northe

		predicted					
		N	R				
actual	N	[[151 0]					
	R	[44 3]					
				precision	recall	f1-score	support
		N	0.77436	1.00000	0.87283	151	
		R	1.00000	0.06383	0.12000	47	
	accuracy				0.77778	198	
	macro avg	0.88718	0.53191	0.49642	198		
	weighted avg	0.82792	0.77778	0.69413	198		





actual	predicted		R=1.7, K=3, Features = 3			
	N	R	precision	recall	f1-score	support
N	[138 13]		0.83636	0.91391	0.87342	151
R	[27 20]		0.60606	0.42553	0.50000	47
accuracy				0.79798		198
macro avg	0.72121	0.66972		0.68671		198
weighted avg	0.78170	0.79798		0.78478		198



Recall: What fraction of the recurring cancers are we detecting (predicted to recur)?
 $(20 / 47 = 0.42553)$

Precision: What fraction of those cancers predicted to recur actually recurred?
 $(20 / 33 = 0.60606)$

actual	predicted		R=2.3, K=17, Features = 1			
N	R	precision	recall	f1-score	support	
N	[151 0]	0.77436	1.00000	0.87283	151	
R	[44 3]	1.00000	0.06383	0.12000	47	
accuracy				0.77778	198	
macro avg	0.88718	0.53191	0.49642		198	
weighted avg	0.82792	0.77778	0.69413		198	

