On the Role of Genetic Algorithms in the Pattern Recognition Task of Classification

Isaac Sherman

Electrical Engineering and Computer Science

April 6, 2017

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Outline

Outline

Basic Genetic Algorithm

```
BasicGA():
        Population := RandomInitialization()
        while True:
                for each Solution in Population:
                        Evaluate Solution
                        Assign Fitness to Solution
                newPopulation = SurviveAndBreed(Population)
                Population = newPopulation
        end while
```

Example

Consider an equation of the form

$$a\ op_1\ b\ op_2\ c\ op_3\ d = x$$

If we know x, we can use a GA to find operators and values which satisfy it. For instance,

$$3 \times 7 \times 3 + 5 = 68$$

Each digit can be encoded with 4 bits, each operator with 2. Values over 9 for digits are rerolled until valid.

Operators for Example

Bits	Operator
00	$+$
01	$-$
10	$\times$
11	$\div$

Therefore,

3	$\times$	7	$\times$	3	+	5
0011	11	1011	11	0011	00	0101

Example-> Fitness function

Here, we maximize $\frac{1}{|S_i - x|}$ and stop when $S_i = x$.
So, if x is 72, then the fitness of

$$3 \times 7 \times 3 + 5 = 68$$

is $\frac{1}{|68-72|} = 0.25$

Motivation

- Where do GAs fit in the greater scheme of pattern recognition?
- Given primitive mechanics, can they match or exceed theoretically-based methods?
- Can we build a generic, universal genetic algorithm for classification?

Outline

Outline

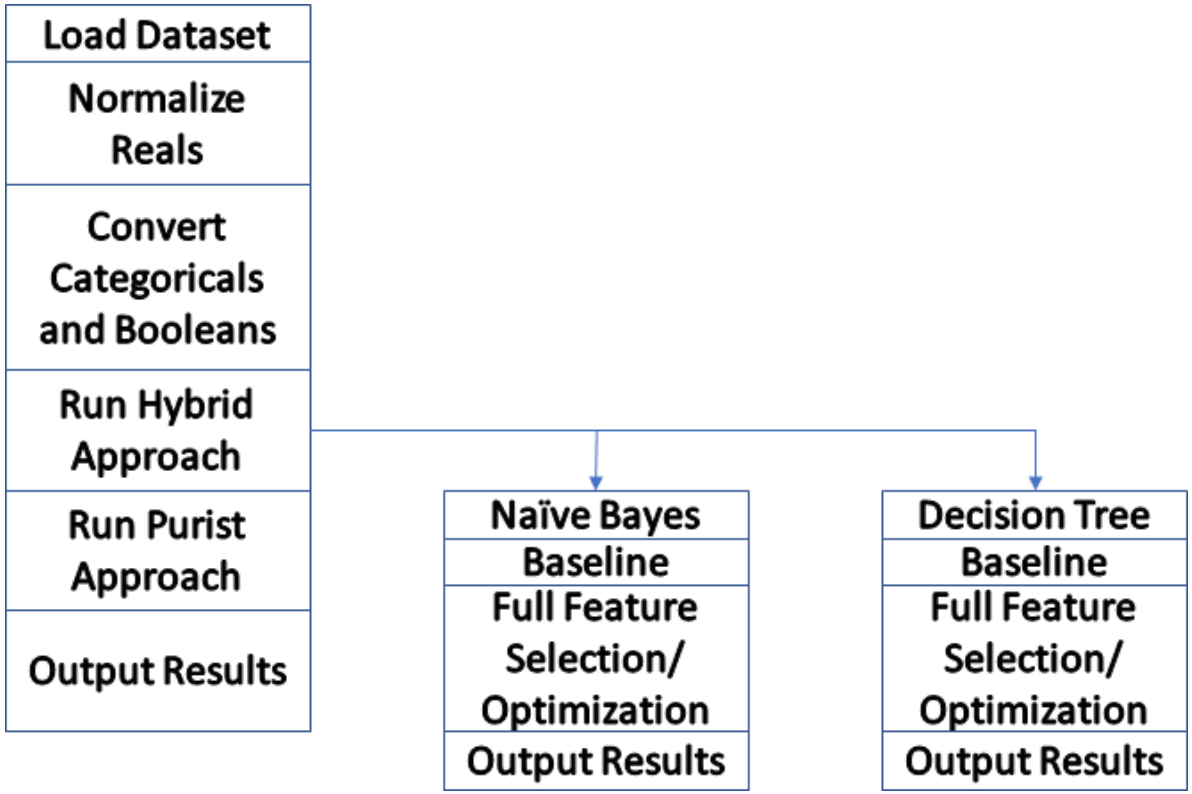


Figure: Birds-eye view of the flow of the program. Hybrid approaches are not run in parallel, because at time of coding MATLAB doesn't support multi-threading via a COM server.

For Reals

Categorical feature types:

$$
\begin{aligned}
X &= X_{\mathbb{R}} \cup X_{cat} \cup X_{bool} \\
X_{\mathbb{R}} &= \{x_1, x_2, \ldots x_n\} \\
X_{cat} &= \{x_{n+1}, x_{n+2}, \ldots x_c\} \\
X_{bool} &= \{x_{c+1}, x_{c+2}, \ldots x_b\} \\
1 &\leq i \leq c - n \\
L &= \{x_i | x_i \epsilon X_{cat}\} \\
C_\ell &= \{x, i | x \epsilon X \wedge x_{n+i} = \ell \epsilon L\} \\
R(\ell) &= \frac{\sum_{x \epsilon C_\ell} \sum_{j=1}^{n} x_j}{|C_\ell| n}
\end{aligned}
$$

Booleans

True is converted to .75, false to .25.

Outline

Hybrid Approach

- Optimizers are solutions which optimize, in this case, classifiers.
- Their purpose is to optimize which settings a classifier uses as well as selecting features.
- Each feature is represented as a bit, usually at the front of the Optimizer.
- The remainder of the genome represent parameters to the classifier
- Two classifiers, Naïve Bayes (McNB) and Classification Tree (CTree)
- Further, two versions of each: with and without feature selection (baseline)
- Fitness is defined as the average accuracy per class, or $\overline{A}$

Fitness

Consider C =

True:	ω_1	ω_2	ω_3	**Total**
Predicted ω_1	4	6	5	**15**
Predicted ω_2	2	2	9	**13**
Predicted ω_3	8	2	110	**120**
Total	**14**	**10**	**124**	**148**
TPR	0.286	0.20	0.89	.459

In this case, $A = \frac{4+2+110}{148} = 78.4\%$, but $\overline{A} = .459$, which better reflects its performance.

Outline

Naïve Bayes

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

Where:

- $P(\omega_j|x)$ is the posterior probability that x belongs to class j
- $P(\omega_j)$ is the rate of occurrence of class j, called the prior probability.
- $p(x|\omega_j)$ is the likelihood of x belonging to ω_j
- $p(x)$ is a normalizing constant to constrain values to $\epsilon[0, 1]$

Shamelessly reproduced from Dr. Hairong Qi's ECE 471 slide

Naïve Bayes

Optimizers optimize the following:

- Features, 1 bit per feature
- Distribution, 2 bits
- Kernel, 2 bits
- Score Transform, 3 bits
- Priors, 3 bits per class

Naïve Bayes

Distribution uses 2 bits and can be one of the following:

- **Kernel** uses a smoothing function to build a distribution
- **Multinomial** represents every class as a single multinomial distribution
- **Multivariate multinomial** characterizes each feature as an independent multinomial distribution based on the unique values found in the feature.
- **Normal**

Naïve Bayes

Kernel uses 2 bits and can be one of the following:

- **Box** uses a uniform, box-like smoothing window.
- **Epanechnikov** is a very efficient, rounded kernel.
- **Gaussian** is a standard normal function but used in this case for smoothing.
- **Triangular** is another form of smoothing, with a peak of 1 at 0 and zero at -1 and 1.

Naïve Bayes

Score transform uses 3 bits and can be any of the following:

- **DoubleLogit** transforms the score to $\frac{1}{1+e^{-2x}}$
- **Invlogit** $log(\frac{x}{1-x})$
- **Logit** $\frac{1}{1+e^{-x}}$
- **None** x
- **Sign** $\frac{x}{|x|}$, or 0 when x = 0.
- **Symmetric** $2x - 1$
- **Symmetricismax** 1 if max of class, 0 otherwise
- **Symmetriclogit** $\frac{2}{1+e^{-x}} - 1$

Outline

- Program Flow
- Optimizers
  - Naïve Bayes
  - Classification Tree
- Hunters
- Metrics
- Datasets
- Yeast
- Cardiotocography (NSP)
- Cardiotocography (Morphology)
- Bach's Chorales
- Population Stats

Classification Tree

- **Features**, 1 bit per feature
- **Merge Leaves**, 1 bit
- **Maximum Splits**, 6 bits
- **Min Leaf Size**, 5 bits
- **Split Criterion**, 2 bits

Classification Tree

Merge Leaves takes 1 bit and is either on or off. Merge leaves looks at leaves from a parent node and if the amount of their risk and that of their offspring is at or greater than that of a parent.

Maximum Splits defines how many splits a tree can have. The tree is built iteratively, layer by layer, splitting as needed until it hits this number. It can take on values of 3-66.

Classification Tree

Min Leaf Size This is the minimum number of samples that need to reach this node to be considered a standalone leaf. Beyond this number (specifically, at twice this number) a leaf become a parent node split into two children. Takes on values between 1 and 32.

Classification Tree

Split Criterion can take on 3 values.

- **Gini's Diversity Index** Aims for maximally diverse cuts
- **Twoing** Aims for a balanced tree
- **Deviance** Minimizes entropy

Evolver

```
AdvanceGeneration():
        EvaluateAllOptimizers(P) //Multithreaded evaluation possible
        GetMetrics()
        P.ReverseSort() //P is the population, an instance variable
        P = GenerateNextGeneration(P)
        RemoveDuplicates(P)

GenerateNextGeneration(P):
        BreedingPop := StochasticRUS(P)
        NextGen := Elitism(P)
        FillListFromBreedingPop(NextGen, BreedingPop,
                                P.Count, UniformXOver)
        MutateNonElites(NextGen)
        return NextGen
```

Evolver

```
FillListFromBreedingPop(N, B, size, Func):
E := B.Count * ElitePercent
while(N.Count < size):
        k := j := RNG.Next(0, E)
        while(j==k)
                k = RNG.Next(0, B.Count)
        for each offspring in Func(B[j], B[k])
        N.Add(offspring)

while (N.Count > size)
        N.pop()
```

Outline

Hunters

- Hunters are made up of 1 or more Chromosomes
- Chromosomes are made up of 1 or more Cells
- Cells form the bulk of the genome of the Hunter
- Fitness is a modified $\overline{A}$

Examine each component from the bottom up

Hunter Fitness

Fitness is equal to $\overline{A}$ with 2 scalars, one for complexity and one for ignoring classes:

$$F_{Hunter} = \overline{A}\left(\frac{C_{Max} - C}{C_{Max}}\right)\left(\frac{E - Z}{E}\right)$$

Outline

Cells

Cells comprise the following:

- Function Index, $\lceil log_2(F) \rceil$ bits, where F is the number of features
- Upper Limit, 8 bits
- Lower Limit, 8 bits
- Not Flag, 1 bit
- Join Bit, 1 bit

Cells

Functions (F_i) are simply looking at features, looking at the values of the normalized dataset.
Limits are a binary number shifted to the -8th power. It allows for values between 0 and $\frac{511}{512}$. If the upper limit (L_u) is lower than the lower limit(L_ℓ), the bits are swapped.
The **Not Flag** (N) reverses the vote of the cell.
So when a cell votes, it returns $V = N \oplus (L_\ell \leq F_i \leq L_u)$
The Join Bit indicates whether to include the next cell in the voting process.

Chromosomes

Chromosomes comprise the following:

- Class Bits, $\lceil log_2(\Omega) \rceil$ bits where Ω is the number of classes
- Affinity Bits, 2
- A List of Cells

Chromosomes

Chromosomes comprise the following:

- Class Bits, $\lceil log_2(\Omega) \rceil$ bits where Ω is the number of classes
- 2 Affinity Bits, discussed in detail later
- A List of Cells

Hunters

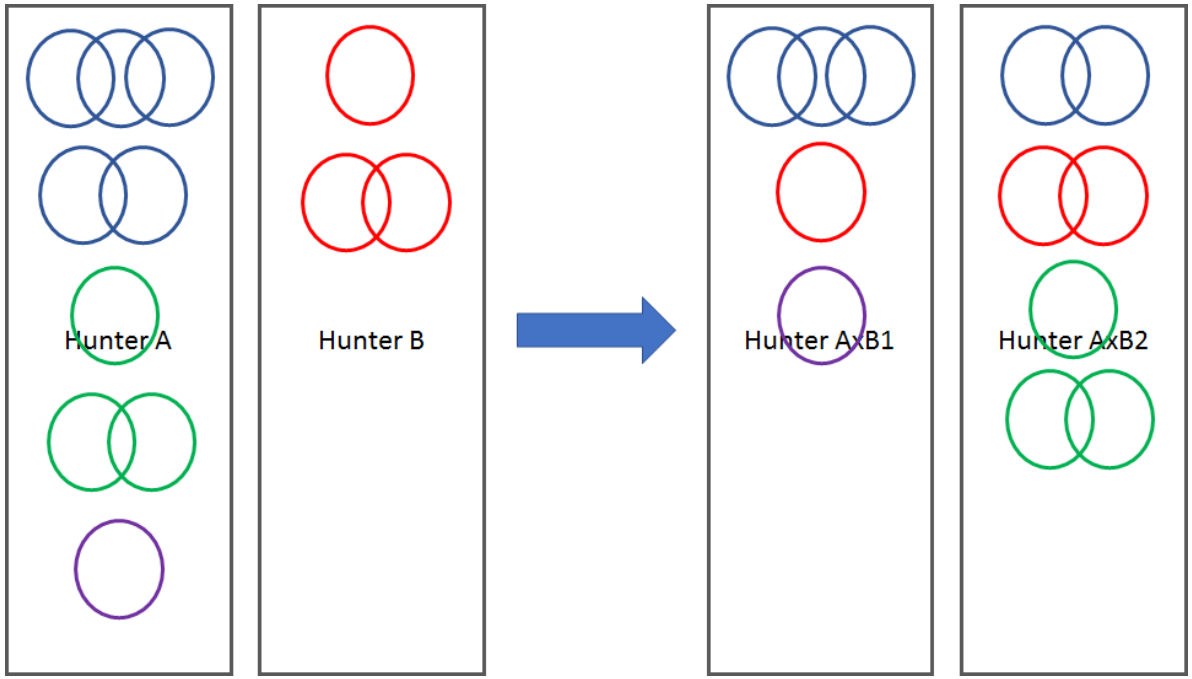
Hunters are merely a housing for a list of Chromosomes. They don't contain any genetic information of their own. However, breeding operations and fitness is calculated at the Hunter level.

Outline

Crossover

Crossover functions differently because Hunters are of variable length. It performs uniform crossover on the lengths which match, but then randomly assigns entire chromosomes to either offspring following the uniform crossover style.

Crossover



Merge

Merge takes 2 hunters and returns a single hunter with a combination of the material of both. The action occurs mostly at the chromosomal level.
Chromosomes have 2 affinity bits which controls the merge operation.

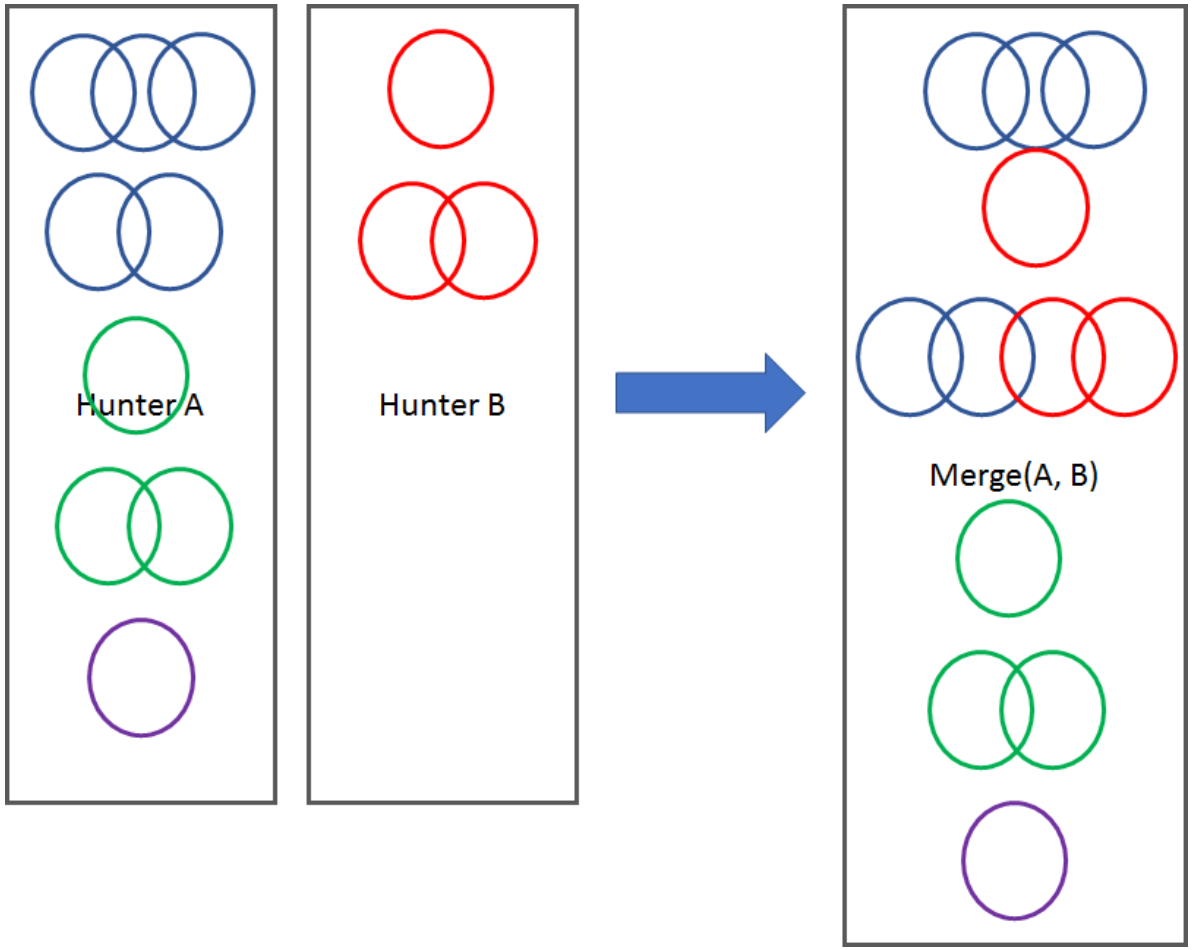
Individually, bits may be understood as follows:

		Meaning
0	0	No preference.
0	1	Prefers to be at the rear.
1	0	Prefers to be at the front.
1	1	Considers itself complete.

When two chromosomes go to merge, the results are determined as follows:

A	B	Result
00	00	
00	01	
10	00	
10	01	Laid out Horizontally with A in front
00	10	
01	00	
01	10	Laid out Horizontally with B in front
11	**	
**	11	
10	10	
01	01	Laid out Vertically

Merge



Outline

Metrics

Matthews Correlation Coefficient (MCC) and Confusion Entropy (CEN) are two promising additional methods for analyzing confusion matrices.

$$MCC = \frac{cov(X, Y)}{\sqrt{cov(X, X) \cdot cov(Y, Y)}}$$

- Falls in the range [-1,1]
- 1 indicates perfect classifier
- -1 indicates perfect anti-classifier
- 0 indicates one or more columns is equal to 0

Metrics

CEN is a bit more complicated:

$$CEN = -\sum_{j}^{N+1} P_j \sum_{k}^{N+1} P^j_{j,k} log_N(P^j_{j,k}) + P^j_{k,j} log_N(P^j_{k,j})$$

$$P^i_{i,j} = \frac{C_{i,j}}{\sum_{k=1}^{N+1} C_{i,k} + C_{k,i}}$$

$$P^j_{i,j} = \frac{C_{i,j}}{\sum_{k=1}^{N+1} C_{j,k} + C_{k,j}}$$

$$P_j = \frac{\sum_{k=1}^{N+1} C_{j,k} + C_{k,j}}{2 \sum_{k,l}^{N+1}} C_{k,l}$$

Metrics

Let's return to our example confusion matrix:

True:	ω_1	ω_2	ω_3	**Total**
Predicted ω_1	4	6	5	**15**
Predicted ω_2	2	2	9	**13**
Predicted ω_3	8	2	110	**120**
Total	**14**	**10**	**124**	**148**
TPR	0.286	0.20	0.89	.459

MCC and CEN are 0.293 and 0.356, respectively.

This is actually fairly good for CEN.

MCC tells a different story.

Metrics

A few more examples of CEN and MCC applied to toy matrices, adapted from [Wei et al., 2010].

Box 1

$(a)\ S = 60$

$$C = \begin{pmatrix} 15 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 15 \end{pmatrix}$$

$\mathrm{MCC} = 1$
$\mathrm{CEN} = 0$

$(b)\ S = 60$

$$C = \begin{pmatrix} 0 & 5 & 5 & 5 \\ 5 & 0 & 5 & 5 \\ 5 & 5 & 0 & 5 \\ 5 & 5 & 5 & 0 \end{pmatrix}$$

$\mathrm{MCC} = -0.333$
$\mathrm{CEN} = 1$

$(c)\ S = 60$

$$C = \begin{pmatrix} 0 & 15 & 0 & 0 \\ 0 & 15 & 0 & 0 \\ 0 & 15 & 0 & 0 \\ 0 & 15 & 0 & 0 \end{pmatrix}$$

$\mathrm{MCC} = 0$
$\mathrm{CEN} = 0.337$

$(d)\ S = 10\ 002$

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 5000 & 0 \\ 0 & 5000 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$\mathrm{MCC} = -0.999$
$\mathrm{CEN} = 0.387$

Outline

Outline

- Program Flow
- Optimizers
- Hunters
- Metrics
- Datasets
  - Yeast
  - Cardiotocography
  - Bach's Chorales
- Yeast
- Cardiotocography (NSP)
- Cardiotocography (Morphology)
- Bach's Chorales
- Population Stats

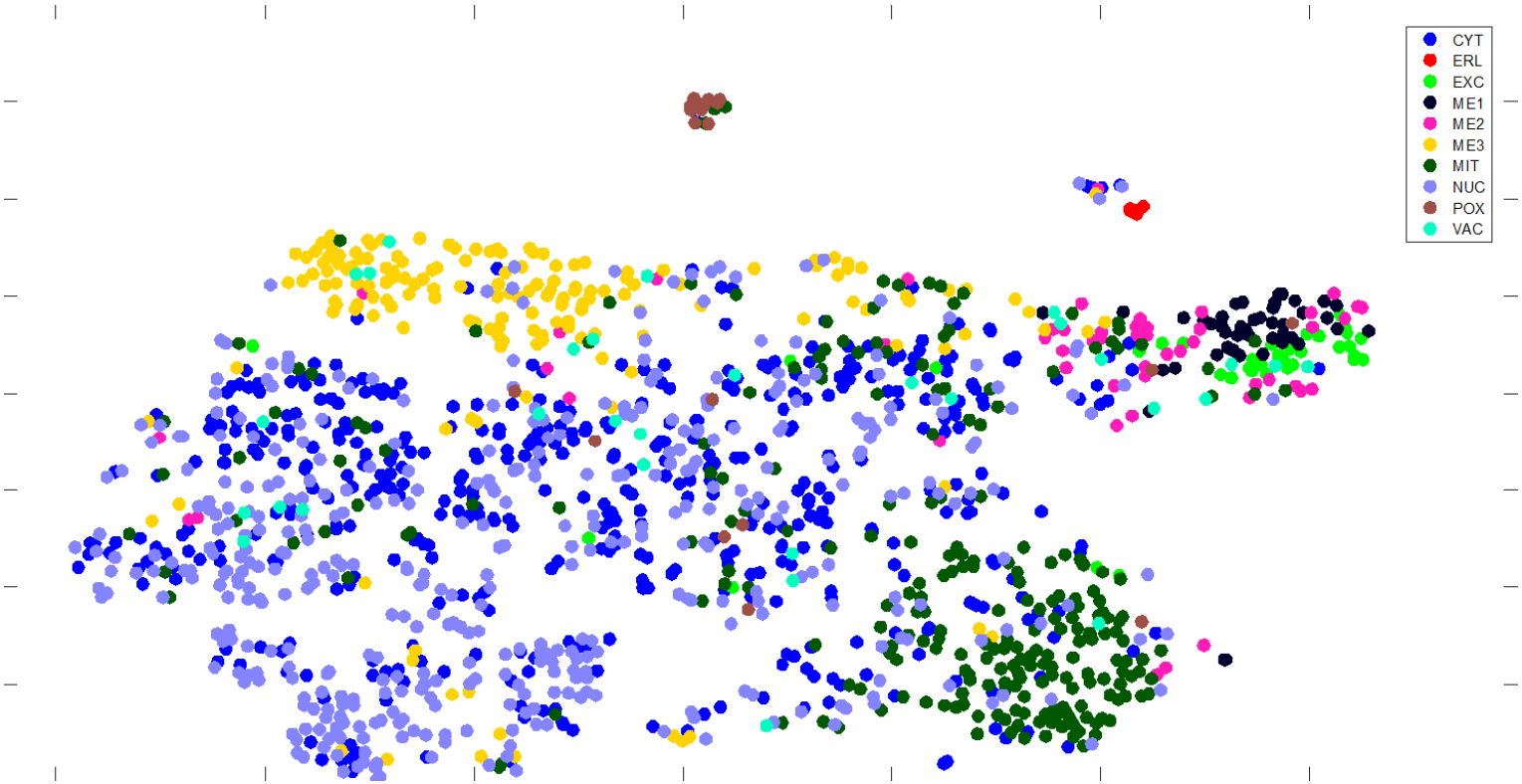


Figure: T-SNE visualization of Yeast dataset.

Outline

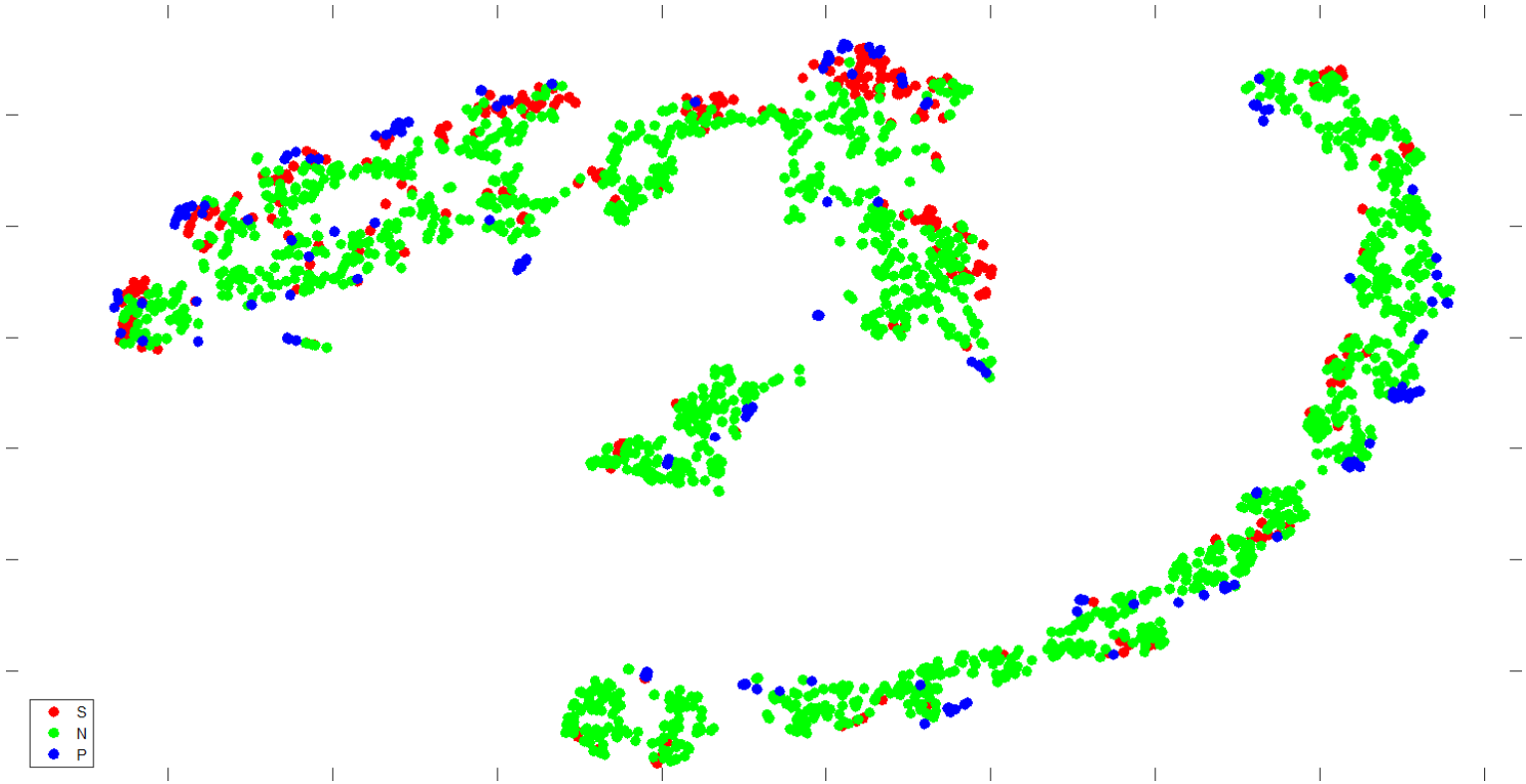


Figure: T-SNE visualization of Cardio dataset, following the NSP classification schema.

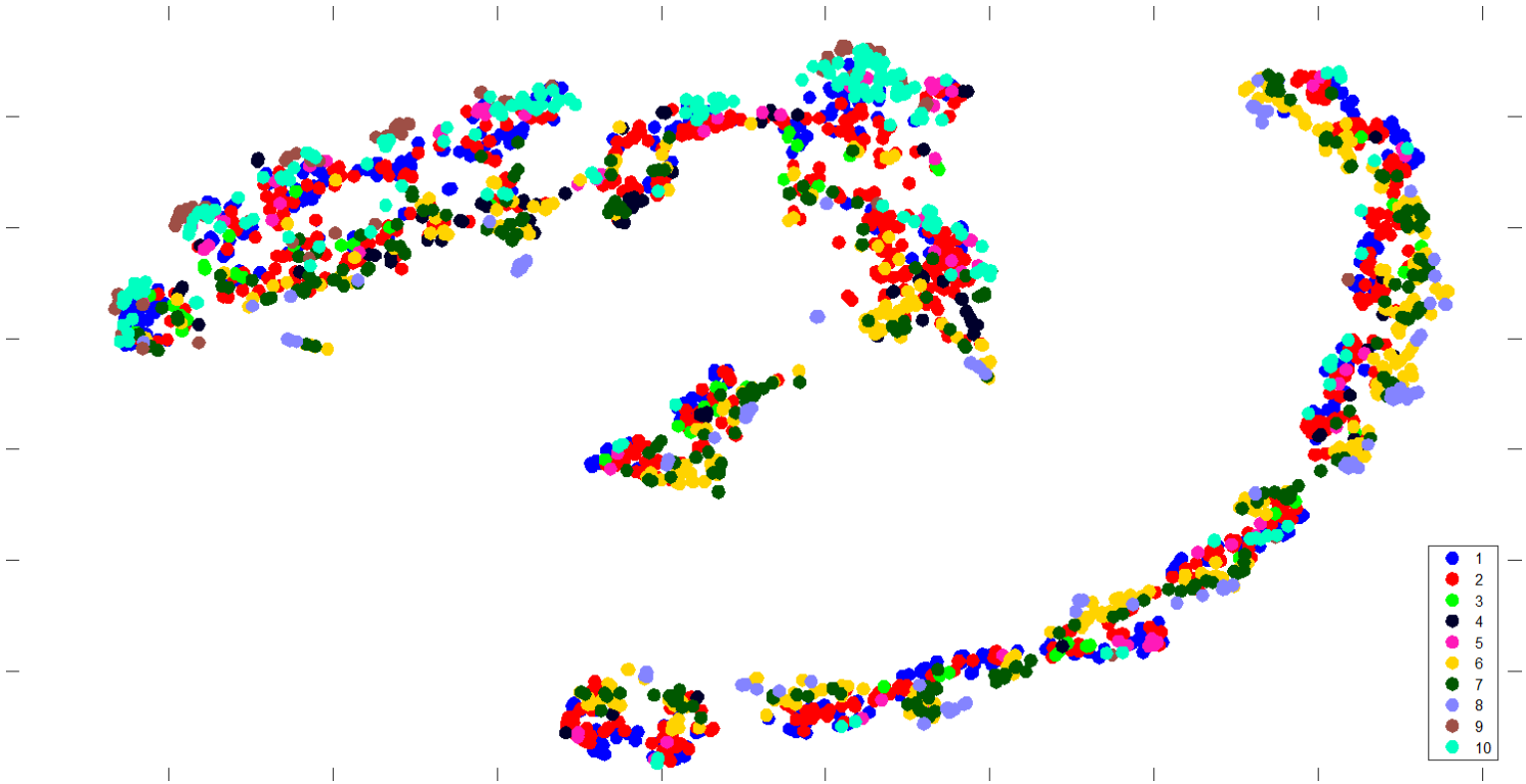


Figure: T-SNE visualization of Cardio dataset, following the 10-class morphology schema. Viewing in color is highly recommended.

Outline

- Program Flow
- Optimizers
- Hunters
- Metrics
- Datasets
  - Yeast
  - Cardiotocography
  - Bach's Chorales
- Yeast
- Cardiotocography (NSP)
- Cardiotocography (Morphology)
- Bach's Chorales
- Population Stats

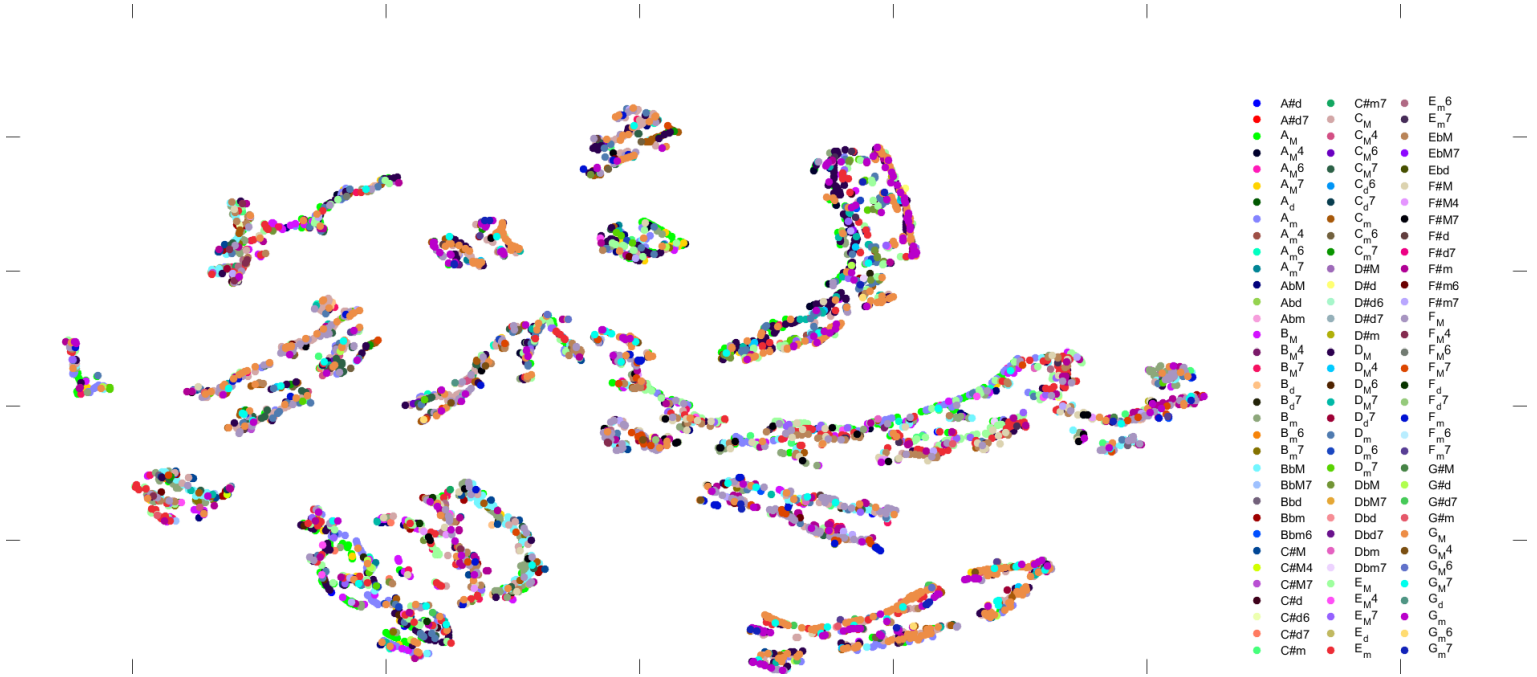


Figure: T-SNE visualization of 102 class Bach dataset. Classes are incredibly difficult to distinguish, but instead focus on the overall shapes and distinct clusters.

Outline

Outline

Yeast: McNB Baseline

Class	MIT	NUC	CYT	ME1	EXC	ME2	ME3	VAC	POX	ERL	Total
MIT	119	10	10	0	0	0	1	0	1	0	141
NUC	9	108	17	0	0	1	0	0	0	0	135
CYT	18	35	152	0	1	0	0	1	0	0	207
ME1	4	4	2	17	0	0	0	0	0	0	27
EXC	4	6	2	0	18	0	0	0	0	0	30
ME2	12	9	5	0	0	32	2	0	0	0	60
ME3	8	10	5	0	0	0	71	0	0	0	94
VAC	1	5	0	0	0	0	0	4	0	0	10
POX	5	8	5	0	0	0	0	0	13	0	31
ERL	0	0	0	0	0	0	0	0	0	5	5
Total:	180	195	198	17	19	33	74	5	14	5	740
TPR:	0.661	0.554	0.768	1	0.947	0.97	0.959	0.8	0.929	1	0.859

Table: Yeast Dataset, Multiclass Naïve Bayes with all features included, Accuracy = 72.8%, MCC: 0.671 CEN: 0.293

Yeast: McNB

Class	MIT	NUC	CYT	ME1	EXC	ME2	ME3	VAC	POX	ERL	Total
MIT	124	22	17	0	0	0	0	0	1	0	164
NUC	8	90	21	0	0	0	0	0	0	0	119
CYT	13	36	134	0	1	0	0	0	0	0	184
ME1	3	1	1	17	0	0	0	0	0	0	22
EXC	2	3	1	0	18	0	0	0	0	0	24
ME2	9	8	4	0	0	33	0	0	0	0	54
ME3	15	20	12	0	0	0	74	0	0	0	121
VAC	2	9	3	0	0	0	0	5	0	0	19
POX	4	5	5	0	0	0	0	0	13	0	27
ERL	0	1	0	0	0	0	0	0	0	5	6
Total:	180	195	198	17	19	33	74	5	14	5	740
TPR:	0.689	0.462	0.677	1	0.947	1	1	1	0.929	1	0.87

Table: Yeast Dataset, Multiclass Naïve Bayes with feature selection included, Accuracy = 72.8%, MCC: 0.630 CEN: 0.310

Yeast: CTree Baseline

Class	MIT	NUC	CYT	ME1	EXC	ME2	ME3	VAC	POX	ERL	Total
MIT	151	14	24	0	0	5	2	1	1	1	199
NUC	8	145	36	0	1	0	4	1	3	0	198
CYT	19	30	133	0	3	4	1	2	5	1	198
ME1	1	0	0	14	2	1	0	0	0	0	18
EXC	0	0	0	0	12	0	0	0	0	0	12
ME2	0	1	1	2	1	21	0	0	0	0	26
ME3	1	4	3	1	0	2	67	0	0	0	78
VAC	0	1	0	0	0	0	0	1	0	0	2
POX	0	0	1	0	0	0	0	0	5	0	6
ERL	0	0	0	0	0	0	0	0	0	3	3
Total:	180	195	198	17	19	33	74	5	14	5	740
TPR:	0.839	0.744	0.672	0.824	0.632	0.636	0.905	0.2	0.357	0.6	0.641

Table: Yeast Dataset, Classification Tree with all features included, Accuracy = 74.6%, MCC: 0.674 CEN: 0.273

Yeast: CTree

Class	MIT	NUC	CYT	ME1	EXC	ME2	ME3	VAC	POX	ERL	Total
MIT	131	13	9	0	0	1	2	0	2	0	158
NUC	8	87	26	0	1	0	1	2	0	0	125
CYT	33	87	158	0	3	2	4	2	4	0	293
ME1	1	0	0	13	2	1	0	0	0	0	17
EXC	1	0	0	1	11	1	0	0	0	0	14
ME2	5	2	1	2	2	26	0	0	0	1	39
ME3	1	5	3	1	0	2	67	0	0	0	79
VAC	0	1	0	0	0	0	0	1	0	0	2
POX	0	0	1	0	0	0	0	0	8	0	9
ERL	0	0	0	0	0	0	0	0	0	4	4
Total:	180	195	198	17	19	33	74	5	14	5	740
TPR:	0.728	0.446	0.798	0.765	0.579	0.788	0.905	0.2	0.571	0.8	0.658

Table: Yeast Dataset, Classification Tree with feature selection, Accuracy = 68.4%, MCC: 0.607 CEN: 0.290

Yeast: Hunter

Class	MIT	NUC	CYT	ME1	EXC	ME2	ME3	VAC	POX	ERL	Total
MIT	105	17	12	0	0	0	0	0	0	0	134
NUC	2	37	10	1	0	1	3	0	0	0	54
CYT	43	94	142	0	5	3	2	1	10	0	300
ME1	4	1	0	8	1	3	7	0	0	0	24
EXC	9	2	3	0	9	7	0	0	2	0	32
ME2	1	9	6	6	4	12	1	1	0	2	42
ME3	14	5	11	2	0	6	58	1	0	0	97
VAC	2	28	13	0	0	1	3	2	0	0	49
POX	0	0	1	0	0	0	0	0	2	0	3
ERL	0	2	0	0	0	0	0	0	0	3	5
Total:	180	195	198	17	19	33	74	5	14	5	740
TPR:	0.583	0.19	0.717	0.471	0.474	0.364	0.784	0.4	0.143	0.6	0.473

Table: Yeast Dataset, Hunter Accuracy = 51.1%, MCC: 0.413 CEN: 0.409

Outline

Cardiotocography (NSP): McNB Baseline

Class	Normal	Suspect	Pathologic	Total
Normal	476	24	6	506
Suspect	37	408	2	447
Pathologic	28	7	75	110
Total:	541	439	83	1063
TPR:	0.88	0.929	0.904	0.904

Table: Cardio Dataset, NSP Labels Multiclass Naïve Bayes without feature selection included, Accuracy = 90.2% MCC: 0.797 CEN: 0.186

Cardiotocography (NSP): McNB

Class	Normal	Suspect	Pathologic	Total
Normal	490	11	1	502
Suspect	28	427	0	455
Pathologic	23	1	82	106
Total:	541	439	83	1063
TPR:	0.906	0.973	0.988	0.956

Table: Cardio Dataset, NSP Labels Multiclass Naïve Bayes with feature selection included, Accuracy = 94%, MCC: 0.889 CEN: 0.116

Cardiotocography (NSP): CTree Baseline

Class	Normal	Suspect	Pathologic	Total
Normal	519	31	9	559
Suspect	12	408	0	420
Pathologic	10	0	74	84
Total:	541	439	83	1063
TPR:	0.959	0.929	0.892	0.927

Table: Cardio Dataset, NSP Labels Classification Tree without feature selection included, Accuracy=94.1%, MCC: 0.967 CEN: 0.043

Cardiotocography (NSP): CTree

Class	Normal	Suspect	Pathologic	Total
Normal	514	34	3	551
Suspect	12	405	0	417
Pathologic	15	0	80	95
Total:	541	439	83	1063
TPR:	0.95	0.923	0.964	0.946

Table: Cardio Dataset, NSP Labels Classification Tree with feature selection included, Accuracy: 93.9% MCC: 0.962 CEN: 0.051

Cardiotocography (NSP): Hunter

Class	Normal	Suspect	Pathologic	Total
Normal	197	77	10	284
Suspect	183	322	4	509
Pathologic	161	40	69	270
Total:	541	439	83	1063
TPR:	0.364	0.733	0.831	0.643

Table: Cardio Dataset, NSP Labels Hunter, Accuracy = 55%, MCC: 0.333
CEN: 0.568

Outline

Cardiotocography (Morphology): McNB Baseline

Class	J	F	A	H	G	B	D	I	E	C	Total
J	99	0	2	0	0	0	0	0	0	0	101
F	0	169	0	0	0	1	0	0	0	1	171
A	2	0	189	0	0	2	0	0	0	1	194
H	0	0	0	47	0	0	0	0	0	0	47
G	0	1	0	1	121	0	0	0	0	0	123
B	0	5	9	0	2	276	1	0	1	0	294
D	0	0	0	0	0	0	40	0	0	0	40
I	2	0	0	0	0	0	0	30	0	0	32
E	2	0	0	0	0	0	0	0	38	0	40
C	0	0	0	0	0	0	0	0	0	21	21
Total:	105	175	200	48	123	279	41	30	39	23	1063
TPR:	0.943	0.966	0.945	0.979	0.984	0.989	0.976	1	0.974	0.913	0.967

Table: Cardio Dataset Morphological Labels Multiclass Naïve Bayes without feature selection, Accuracy= 96.9% MCC: 0.963, CEN: 0.05

Cardiotocography (Morphology): McNB

Class	J	F	A	H	G	B	D	I	E	C	Total
J	100	0	1	0	0	0	0	0	0	0	101
F	0	171	0	0	0	0	0	0	0	0	171
A	0	0	187	0	1	4	0	0	1	1	194
H	0	0	0	47	0	0	0	0	0	0	47
G	0	1	0	0	122	0	0	0	0	0	123
B	0	4	2	0	1	285	1	0	1	0	294
D	0	0	0	0	0	0	40	0	0	0	40
I	0	0	0	0	0	0	0	32	0	0	32
E	0	0	0	0	0	0	0	0	40	0	40
C	0	0	0	0	0	0	0	0	0	21	21
Total:	100	176	190	47	124	289	41	32	42	22	1063
TPR:	1	0.972	0.984	1	0.984	0.986	0.976	1	0.952	0.955	0.981

Table: Cardio Dataset Morphological Labels Multiclass Naïve Bayes with feature selection, Accuracy = 98.3%, MCC: 0.98, CEN: 0.03

Cardiotocography (Morphology): CTree Baseline

Class	J	F	A	H	G	B	D	I	E	C	Total
J	97	0	2	0	0	0	0	0	2	0	101
F	0	159	0	1	5	6	0	0	0	0	171
A	4	0	183	0	0	5	0	0	1	1	194
H	0	0	0	42	2	0	1	0	0	2	47
G	0	1	1	0	120	1	0	0	0	0	123
B	0	0	1	0	0	286	3	0	3	1	294
D	0	1	0	0	0	1	38	0	0	0	40
I	3	0	1	0	0	0	0	28	0	0	32
E	2	0	5	0	0	1	0	0	31	1	40
C	0	0	2	0	0	0	0	0	0	19	21
Total:	106	161	195	43	127	300	42	28	37	24	1063
TPR:	0.915	0.988	0.938	0.977	0.945	0.953	0.905	1	0.838	0.792	0.925

Table: Cardio Dataset, Morphological Labels Classification Tree without feature selection included, Accuracy = 94.3% MCC: 0.933, CEN: 0.086

Cardiotocography (Morphology): CTree

Class	J	F	A	H	G	B	D	I	E	C	Total
J	98	0	0	0	0	0	0	0	2	1	101
F	0	161	1	1	1	7	0	0	0	0	171
A	6	0	178	0	0	4	0	1	3	2	194
H	0	0	2	44	0	0	1	0	0	0	47
G	0	4	1	0	118	0	0	0	0	0	123
B	0	1	3	0	0	283	3	0	4	0	294
D	0	1	0	0	0	0	39	0	0	0	40
I	3	0	0	0	0	0	0	29	0	0	32
E	1	0	5	0	0	0	0	0	33	1	40
C	0	0	2	0	0	0	0	0	0	19	21
Total:	108	167	192	45	119	294	43	30	42	23	1063
TPR:	0.907	0.964	0.927	0.978	0.992	0.963	0.907	0.967	0.786	0.826	0.922

Table: Cardio Dataset, Morphological Labels Classification Tree with feature selection included accuracy: 94.3% MCC: 0.931, CEN: 0.086

Cardiotocography (Morphology): Hunter

Class	J	F	A	H	G	B	D	I	E	C	Total
J	28	19	0	1	21	0	3	2	27	0	101
F	0	5	0	0	2	163	1	0	0	0	171
A	16	86	10	0	33	5	1	40	1	2	194
H	0	1	0	38	4	3	0	1	0	0	47
G	0	43	0	1	65	3	10	1	0	0	123
B	2	3	0	0	2	284	1	2	0	0	294
D	0	0	0	0	0	40	0	0	0	0	40
I	0	6	0	16	0	0	0	10	0	0	32
E	9	7	0	0	12	0	0	8	4	0	40
C	0	11	0	0	3	0	1	6	0	0	21
Total:	55	181	10	56	142	498	17	70	32	2	1063
TPR:	0.509	0.028	1	0.679	0.458	0.57	0	0.143	0.125	0	0.351

Table: Cardio Dataset Morphological Labels Multiclass Naïve Bayes with feature selection, Accuracy = 41.7%, MCC: -0.078, CEN: 0.49

Outline

For Bach, with 102 classes, the confusion matrix is not easily repackaged. It's on Github as a spreadsheet. Instead, we'll summarize with the metrics.

Bach's Chorales: McNB Baseline

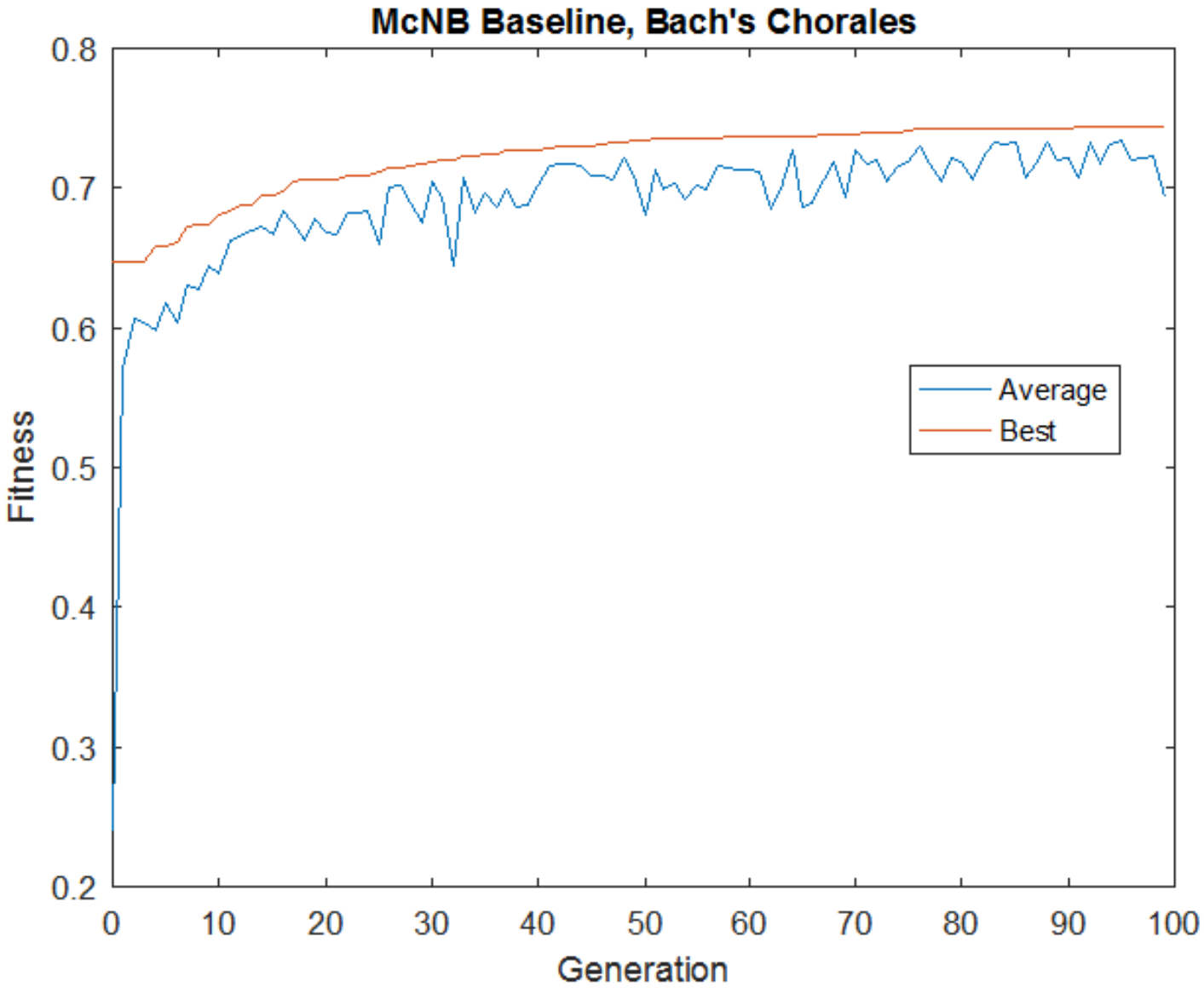
Classifier	Accuracy	$\overline{A}$	MCC	CEN
CTree Baseline	0.709	0.244	0.694	0.18
CTree	0.679	0.270	0.662	0.192
McNB Baseline	0.794	0.662	0.785	0.139
McNB	0.803	0.653	0.794	0.136
Hunter	-	-	-	-
Original Paper	0.75	-	-	-

Outline

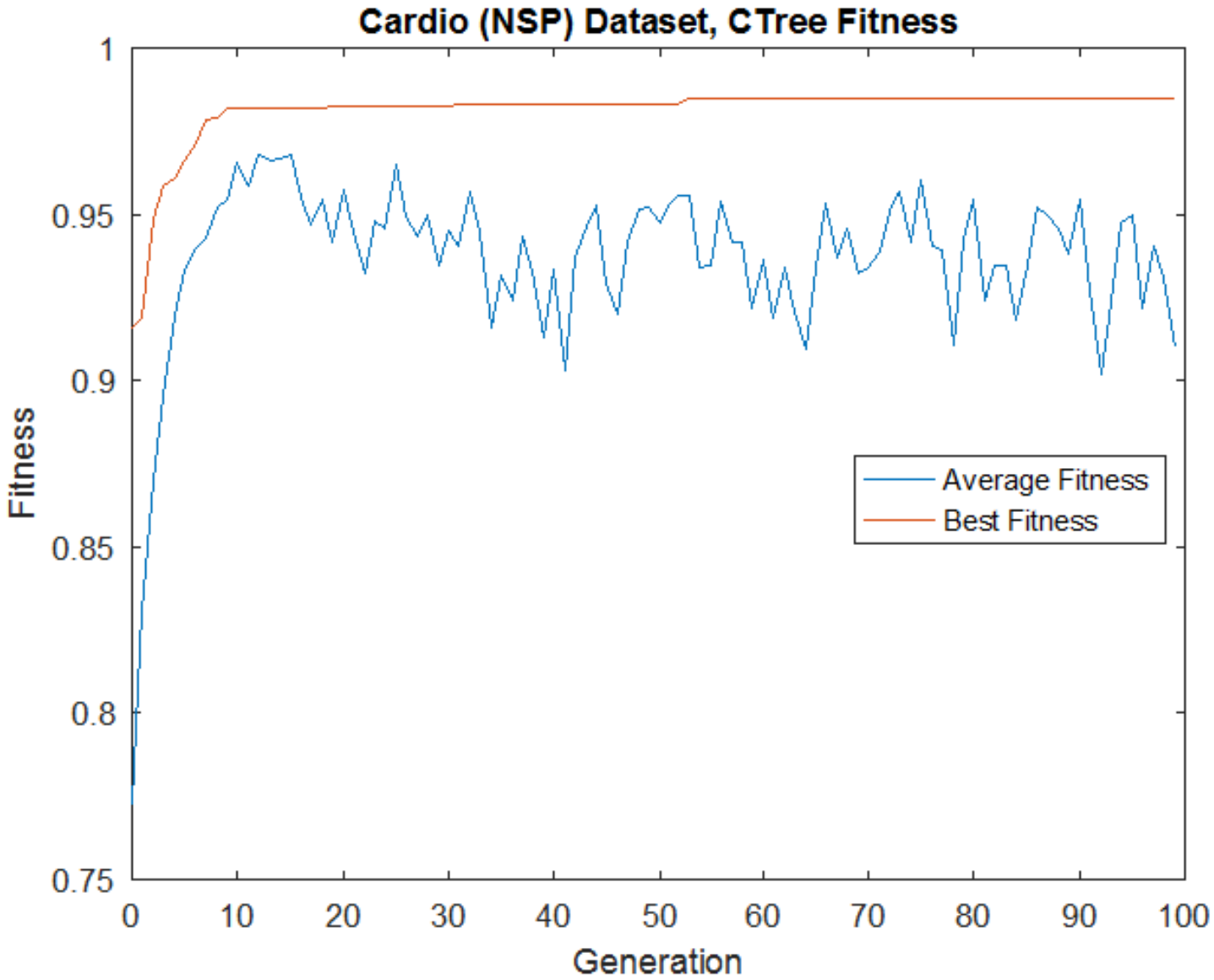
Population Stats

- With evolution, it's not only the fittest that survive.
- Entire population should trend toward greater fitness
- Otherwise, there's a fundamental failure of the GA.

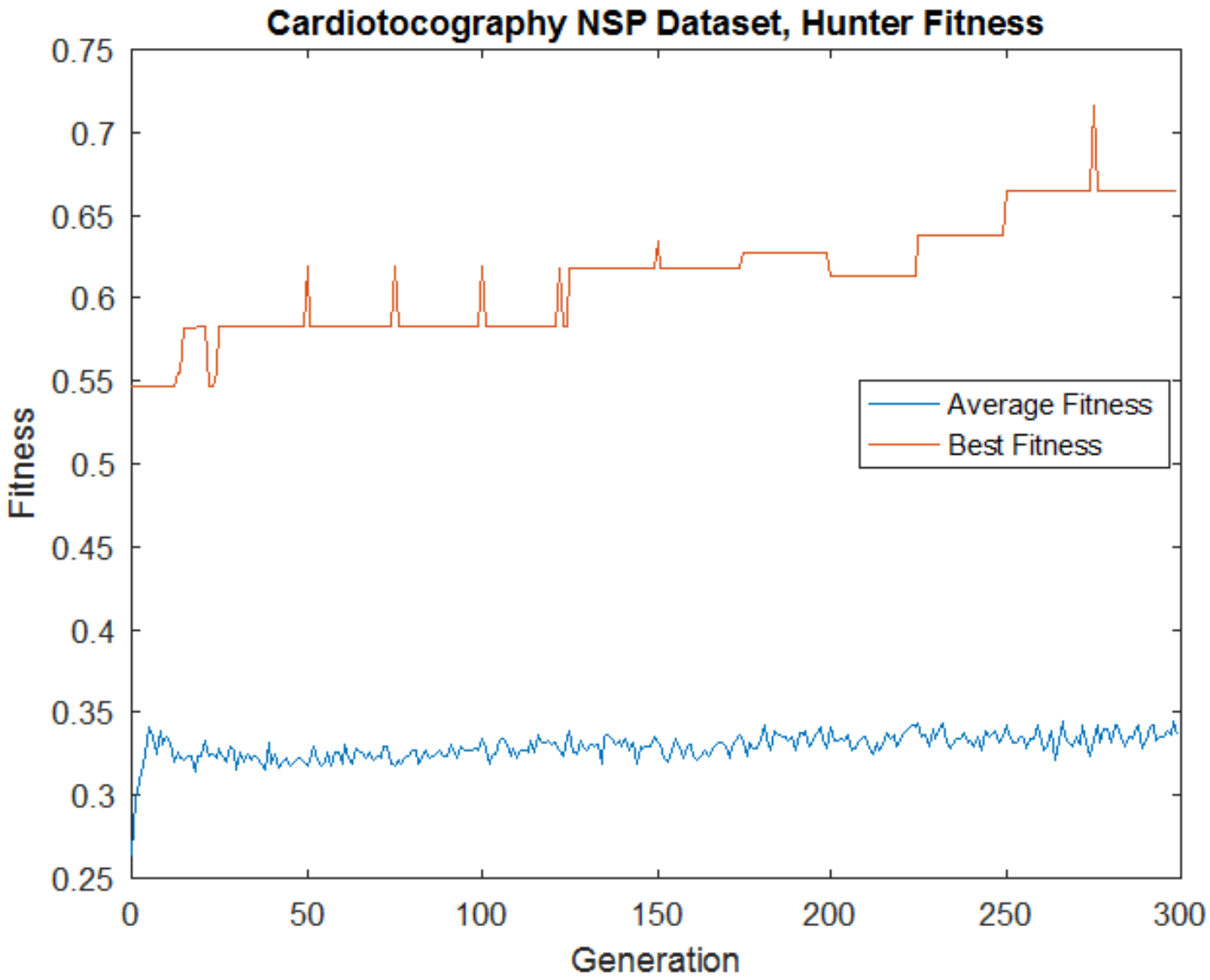
Naïve Bayes, Bach



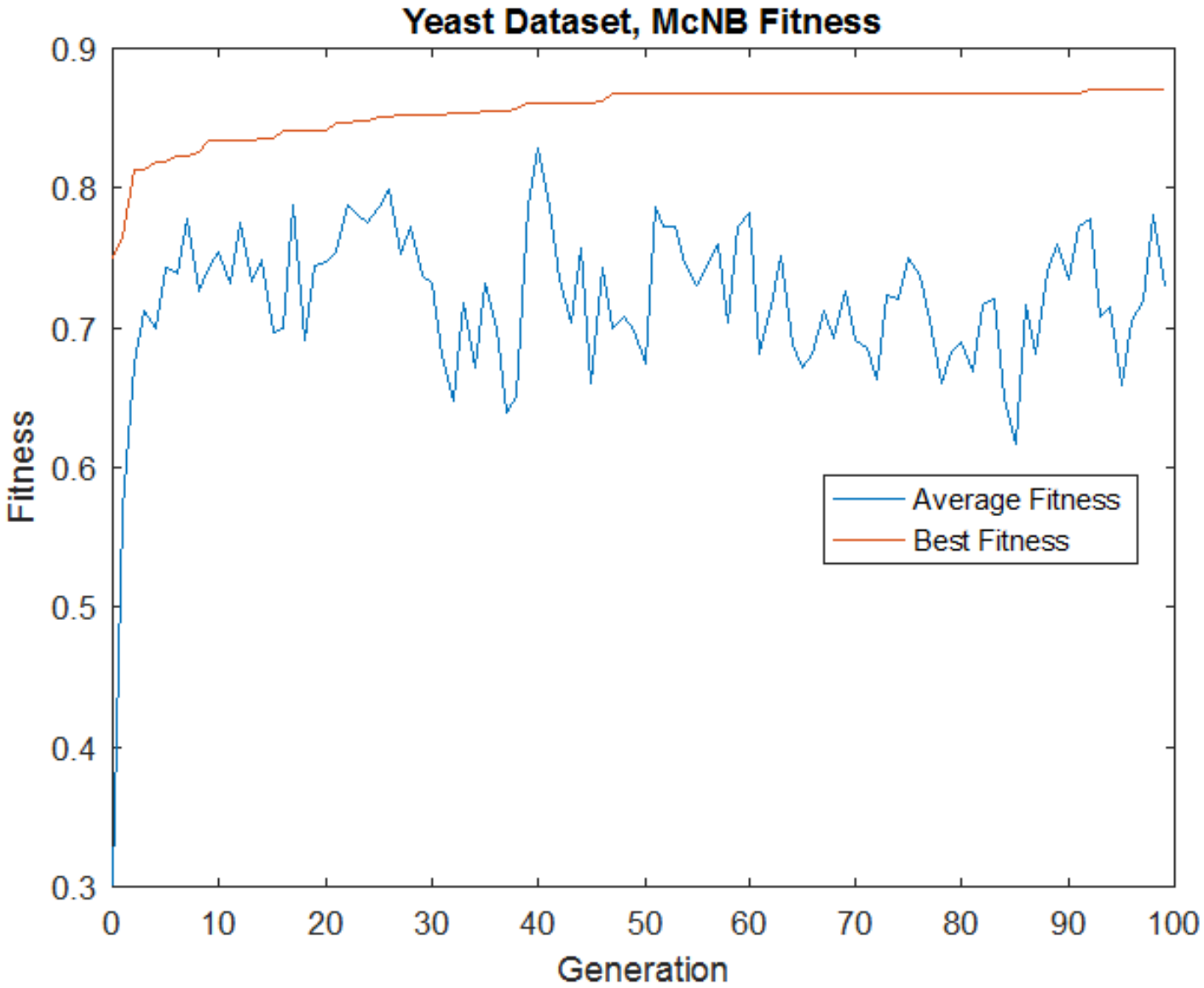
Classification Tree, Cardiotocography (NSP)



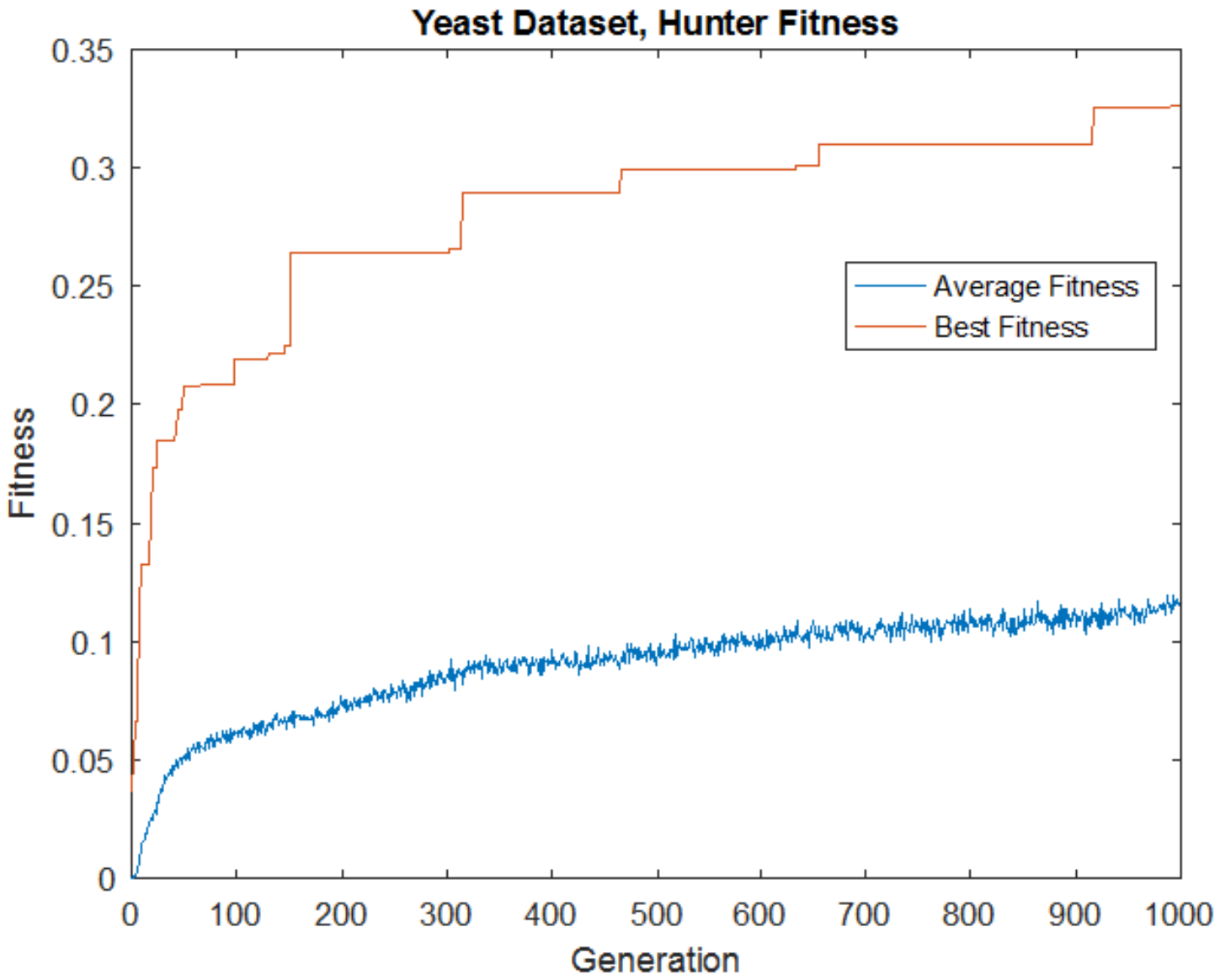
Hunter, Cardiotocography (NSP)



Naïve Bayes, Yeast



Hunter, Yeast



Outline

Conclusions

- Hunters BAD
- Everything else good
- Sleep now

In the beginning, we imagined we were testing something like this:

vs

But it was more like this:

vs

But it was more like this:

vs

Pitfalls

- Breeding Hunters is too CPU intensive, even with smaller populations.
- “Toolbox” not robust enough for Hunters

Hunter Further research

- Refine breeding, perhaps only include cellular crossover at some probability instead of always
- Do more than just boolean decisions (like building distributions or leveraging denoising auto-encoders)
- Increase selective pressure
- Improve

Other Further research

- Refine breeding, perhaps only include cellular crossover at some probability instead of always
- Do more than just boolean decisions (like building distributions or leveraging denoising auto-encoders)

Outline

Bibliography I


Affenzeller, M. and Wagner, S. (2003).
A self-adaptive model for selective pressure handling within the theory of genetic algorithms.
In *International Conference on Computer Aided Systems Theory*, pages 384–393. Springer.


Alexandre, E., Cuadra, L., Salcedo-Sanz, S., Pastor-SÃąnchez, A., and Casanova-Mateo, C. (2015).
Hybridizing Extreme Learning Machines and Genetic Algorithms to select acoustic features in vehicle classification applications.
Neurocomputing, 152:58–68.

Bibliography II


Ayres-de campos, D., Bernardes, J., Garrido, A., Marques-de sÃą, J., and Pereira-leite, L. (2000).
SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms.
Journal of Maternal-Fetal Medicine, 9(5):311–318.


Back, T. (1994).
Selective pressure in evolutionary algorithms: a characterization of selection mechanisms.
In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 57–62 vol.1.


CEDAR (2002).
CEDAR Databases.

Bibliography III


Chou, J.-S., Cheng, M.-Y., Wu, Y.-W., and Pham, A.-D. (2014).
Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification.
Expert Systems with Applications, 41(8):3955–3964.


Daniele P. Radicioni and Roberto Esposito (2014).
UCI Machine Learning Repository: Bach Choral Harmony Data Set.


Dehuri, S., Patnaik, S., Ghosh, A., and Mall, R. (2008).
Application of elitist multi-objective genetic algorithm for classification rule generation.
Applied Soft Computing, 8(1):477–487.

Bibliography IV


Devos, O., Downey, G., and Duponchel, L. (2014).
Simultaneous data pre-processing and SVM classification model selection based on a parallel genetic algorithm applied to spectroscopic data of olive oils.
Food Chemistry, 148:124–130.


Duan, L., Guo, L., Liu, K., Liu, E. H., and Li, P. (2014).
Characterization and classification of seven Citrus herbs by liquid chromatographyâĂŞquadrupole time-of-flight mass spectrometry and genetic algorithm optimized support vector machines.
Journal of Chromatography A, 1339:118–127.

Bibliography V


Fidelis, M. V., Lopes, H. S., and Freitas, A. A. (2000).
Discovering comprehensible classification rules with a genetic algorithm.
In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 805–810 vol.1.


Hoque, M. S., Mukit, M. A., and Bikas, M. A. N. (2012).
An Implementation of Intrusion Detection System Using Genetic Algorithm.
International Journal of Network Security & Its Applications, 4(2):109–120.
arXiv: 1204.1336.


Hyafil, L. and Rivest, R. L. (1976).
Constructing optimal binary decision trees is NP-complete.
Information processing letters, 5(1):15–17.

Bibliography VI


J. P. Marques de SÃČÂą, J. Bernardes, and D. Ayres de Campos (2010).
UCI Machine Learning Repository: Cardiotocography Data Set.


Jurman, G., Riccadonna, S., and Furlanello, C. (2012).
A Comparison of MCC and CEN Error Measures in Multi-Class Prediction.
PLOS ONE, 7(8):e41882.


Kharma, N., Kowaliw, T., Clement, E., Jensen, C., Youssef, A., and Yao, J. (2004).
PROJECT CellNet: EVOLVING AN AUTONOMOUS PATTERN RECOGNIZER.
International Journal of Pattern Recognition and Artificial Intelligence, 18(06):1039–1056.

Bibliography VII


Kowaliw, T., Kharma, N., Jensen, C., Moghnieh, H., and Yao, J. (2004).
CellNet Co-Ev: Evolving Better Pattern Recognizers Using Competitive Co-evolution.
In *Genetic and Evolutionary Computation âĂŞ GECCO 2004*, pages 1090–1101. Springer, Berlin, Heidelberg.
DOI: 10.1007/978-3-540-24855-2_119.


Kozeny, V. (2015).
Genetic algorithms for credit scoring: Alternative fitness function performance comparison.
Expert Systems with Applications, 42(6):2998–3004.

Bibliography VIII


Li, L., Zhang, G., Nie, J., Niu, Y., and Yao, A. (2012).
The Application of Genetic Algorithm to Intrusion Detection in MP2p Network.
In *Advances in Swarm Intelligence*, pages 390–397. Springer, Berlin, Heidelberg.
DOI: 10.1007/978-3-642-30976-2_47.


Lichman, M. (2013).
UCI Machine Learning Repository.
University of California, Irvine, School of Information and Computer Sciences.


Maaten, L. v. d. and Hinton, G. (2008).
Visualizing Data using t-SNE.
Journal of Machine Learning Research, 9(Nov):2579–2605.

Bibliography IX


Marchetti, M., Onorati, F., Matteucci, M., Mainardi, L., Piccione, F., Silvoni, S., and Priftis, K. (2013).
Improving the Efficacy of ERP-Based BCIs Using Different Modalities of Covert Visuospatial Attention and a Genetic Algorithm-Based Classifier.
PLOS ONE, 8(1):e53946.


Nakai, K. and Kanehisa, M. (1992).
A knowledge base for predicting protein localization sites in eukaryotic cells.
Genomics, 14(4):897–911.


Ocak, H. (2013).
A Medical Decision Support System Based on Support Vector Machines and the Genetic Algorithm for the Evaluation of Fetal Well-Being.
Journal of Medical Systems, 37(2):9913.

Bibliography X


Paul Horton (1996).
UCI Machine Learning Repository: Yeast Data Set.


Radicioni, D. P. and Esposito, R. (2010).
BREVE: An HMPerceptron-Based Chord Recognition System.
In RaÅŻ, Z. W. and Wieczorkowska, A. A., editors, *Advances in Music Information Retrieval*, number 274 in Studies in Computational Intelligence, pages 143–164. Springer Berlin Heidelberg.
DOI: 10.1007/978-3-642-11674-2_7.


Ruiz, E., Albareda-Sambola, M., FernÃąndez, E., and Resende, M. G. C. (2015).
A biased random-key genetic algorithm for the capacitated minimum spanning tree problem.
Computers & Operations Research, 57:95–108.

Bibliography XI


Salari, N., Shohaimi, S., Najafi, F., Nallappan, M., and Karishnarajah, I. (2014).
A Novel Hybrid Classification Model of Genetic Algorithms, Modified k-Nearest Neighbor and Developed Backpropagation Neural Network.
PLOS ONE, 9(11):e112987.


Schuman, C. D., Birdwell, J. D., and Dean, M. E. (2014).
Spatiotemporal classification using neuroscience-inspired dynamic architectures.
Procedia Computer Science, 41:89–97.


Srikanth, R., George, R., Warsi, N., Prabhu, D., Petry, F. E., and Buckles, B. P. (1995).
A variable-length genetic algorithm for clustering and classification.
Pattern Recognition Letters, 16(8):789–800.

Bibliography XII


Stefanie Scheid (2004).
Introduction to Kernel Smoothing | Kernel (Operating System) | Probability Density Function.


Tseng, M.-H., Chen, S.-J., Hwang, G.-H., and Shen, M.-Y. (2008).
A genetic algorithm rule-based approach for land-cover classification.
ISPRS Journal of Photogrammetry and Remote Sensing, 63(2):202–212.


Uysal, A. K. and Gunal, S. (2014).
Text classification using genetic algorithm oriented latent semantic features.
Expert Systems with Applications, 41(13):5938–5947.

Bibliography XIII


Wei, J.-M., Yuan, X.-J., Hu, Q.-H., and Wang, S.-Q. (2010).
A novel measure for evaluating classifiers.
Expert Systems with Applications, 37(5):3799–3809.


Wu, J., Long, J., and Liu, M. (2015).
Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm.
Neurocomputing, 148:136–142.


Zhang, J. (2003).
Evolution by gene duplication: an update.
Trends in Ecology & Evolution, 18(6):292–298.

Bibliography XIV

Zhang, Z., McDonnell, K. T., Zadok, E., and Mueller, K. (2015).

Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map.
IEEE Transactions on Visualization and Computer Graphics, 21(2):289–303.