

Mining Feature Importance: Applying Evolutionary Algorithms within a Web-based Educational System

CITSA 2004, Orlando, July 2004

**Behrouz Minaei-Bidgoli,
Gerd Kortemeyer, William F. Punch**

Michigan State University

Topics

- Problem Overview
- Classification Methods
- Combination of Classifiers
- Weighting the features
- Using GA to choose best set of weights
- Experimental Results
- Feature Importance

Problem Overview

- This research is a part of the latest online educational system developed at Michigan State University (MSU), the *Learning Online Network with Computer-Assisted Personalized Approach (LON-CAPA)*.
- In LON-CAPA, we are involved with two kinds of large data sets:
 - Educational resources: web pages, demonstrations, simulations, individualized problems, quizzes, and examinations.
 - Information about users who create, modify, assess, or use these resources.
- Find *classes* of students. Groups of students use these online resources in a *similar* way.
- *Predict* for any individual student to which class he/she belongs.

Classifiers

- Non-Tree Classifiers (Using MATLAB)
 - Bayesian Classifier
 - 1NN
 - kNN
 - Multi-Layer Perceptron
 - Parzen Window
 - **Combination of Multiple Classifiers (CMC)**
 - ***Genetic Algorithm (GA)***
- Decision Tree-Based Software
 - C5.0 (RuleQuest <<C4.5<<ID3)
 - CART (Salford-systems)
 - QUEST (Univ. of Wisconsin)
 - CRUISE [use an *unbiased* variable selection technique]

Statement of Problem(1)

Our claim is that data mining can help to design better and more intelligent educational web-based environment



Can help instructor to design the course more effectively, detect anomaly



Can help students to use the resources more efficiently

Statement of Problem (2)

Can find *classes* of students. Groups of students use these online resources in a *similar* way

Predict for any individual student to which class he/she belongs

Can help the instructor provide appropriate advising in a timely manner

Data Sets: MSU online courses

Course	# of Students	# of	Size of Activity log	Size of useful data	# of Transactions
		Problems			
ADV 205	609	773	82.5 MB	12.1 MB	424,481
BS 111	372	229	361.2 MB	34.1 MB	1,112,394
BS 111	402	229	367.6 MB	50.2 MB	1,689,656
CE 280	178	196	28.9 MB	3.5 MB	127,779
FI 414	169	68	16.8 MB	2.2 MB	83,715
LBS 271	132	174	119.8 MB	18.7 MB	706,700
LBS 272	102	166	73.9 MB	15.3 MB	585,524
MT 204	27	150	5.2 MB	0.7 MB	23,741
MT 432	62	150	20.0 MB	2.4 MB	90,120
PHY 183	227	184	140.3 MB	21.3 MB	452,342
PHY 183	306	255	210.1 MB	26.8 MB	889,775
PHY 231c	99	247	67.2 MB	14.1 MB	536,691
PHY 232c	83	194	55.1 MB	10.9 MB	412,646
PHY 232	220	259	138.5 MB	19.7 MB	981,568

Extracted Features

1. Total number of attempts
2. Total no. of correct answers (Success rate)
3. Success on the first try
4. Success on the second try
5. Success after 3 to 9 attempts
6. Success after 10 or more attempts
7. Total time until the correct answer
8. Total time spent, regardless of success
9. Participation in online communication

GA Optimizer vs. Classifier

- Apply GA directly as a classifier
- Use GA as an optimization tool for resetting the parameters in other classifiers.
 - Most application of GA in pattern recognition applies GA as an optimizer for some parameters in the classification process.
 - GA is applied to find an optimal set of feature weights that improve classification accuracy.

Individual Representation

- The GA Toolbox supports binary, integer and floating-point chromosome representations.
- `Chrom = crtrp(NIND, FieldDR)` creates a random real-valued matrix of $N \times d$, `NIND` specifies the number of individuals in the population
- `FieldDR = [0 0 0 0 0 0 0; % lower bound
1 1 1 1 1 1]; % upper bound`
- `Chrom =`

0.23	0.17	0.95	0.38	0.06	0.26
0.35	0.09	0.43	0.64	0.20	0.54
0.50	0.10	0.09	0.65	0.68	0.46
0.21	0.29	0.89	0.48	0.63	0.89

Simple GA

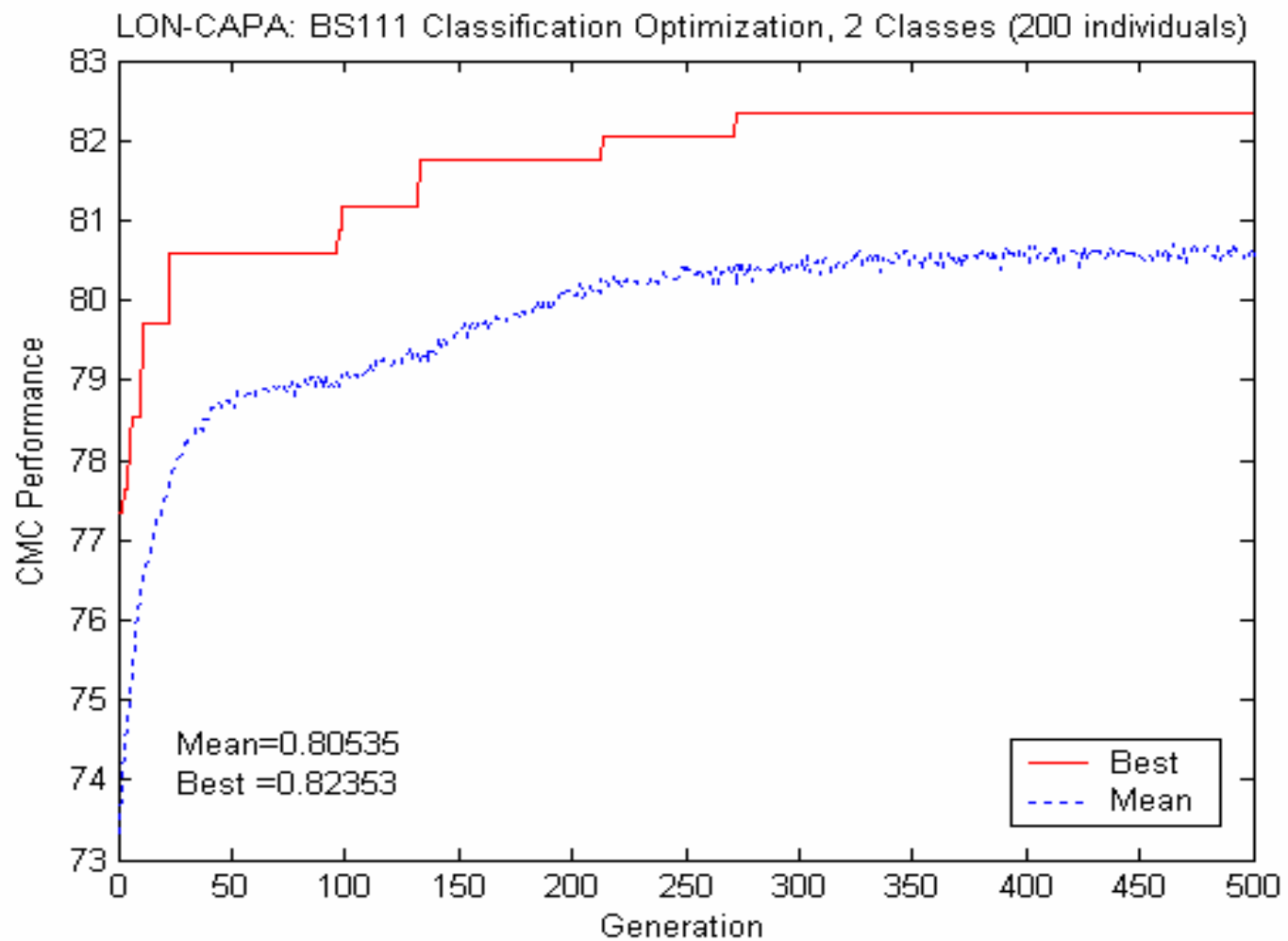
```
% Create population                                Chrom = crtrp(NIND, FieldD);
% Evaluate initial population,                      ObjV = ObjFn2(data, Chrom);
gen = 0;
while gen < MAXGEN,
    % Assign fitness-value to entire population      FitnV = ranking(ObjV);
    % Select individuals for breeding SelCh = select(SEL_F, Chrom, FitnV, GGAP);
    % Recombine selected individuals (crossover)
                                SelCh = recomb(XOV_F, SelCh,XOVR);
    % Perform mutation on offspring
                                SelCh = mutata(MUT_F, SelCh, FieldD, MUTR);
    % Evaluate offspring, call objective function    ObjVSel = ObjFn2(data, SelCh);
    % Reinsert offspring into current population
                                [Chrom ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel);

    gen = gen+1;
    MUTR = MUTR+0.001;
end
```

Results without GA

Data sets	Bayes	1NN	k NN	Parzen	Classification Fusion
				Window	
ADV 205, 03	55.7	69.9	70.7	55.8	78.2
BS 111, 02	54.6	67.8	69.6	57.6	74.9
BS 111, 03	52.6	62.1	55	59.7	71.2
CE 280, 03	66.6	73.6	74.9	65.2	81.4
FI 414, 03	65	76.4	72.3	70.3	82.2
LBS 271, 02	66.9	75.6	73.8	59.6	79.2
LBS 272, 03	72.3	70.4	69.6	65.3	77.6
MT 204, 03	63.4	71.5	68.4	56.4	82.2
MT 432, 03	67.6	77.6	79.1	59.8	84
PHY 183, 02	73.4	76.8	80.3	65	83.9
PHY 183, 03	59.6	66.5	70.4	54.4	76.6
PHY 231c, 03	56.7	74.5	72.6	60.9	80.7
PHY 232c, 03	65.6	71.7	75.6	57.8	81.6
PHY 232, 03	59.9	73.5	71.4	56.3	79.8

Results of using GA



GA Optimization Results

Data sets	Without GA	GA ptimized	Improvement
ADV 205, 03	78.19 ± 1.34	89.11 ± 1.23	10.92 ± 0.94
BS 111, 02	74.93 ± 2.12	87.25 ± 0.93	12.21 ± 1.65
BS 111, 03	71.19 ± 1.34	81.09 ± 2.42	9.82 ± 1.33
CE 280, 03	81.43 ± 2.13	92.61 ± 2.07	11.36 ± 1.41
FI 414, 03	82.24 ± 1.54	91.73 ± 1.21	9.50 ± 1.76
LBS 271, 02	79.23 ± 1.92	90.02 ± 1.65	10.88 ± 0.64
LBS 272, 03	77.56 ± 0.87	87.61 ± 1.03	10.11 ± 0.62
MT 204, 03	82.24 ± 1.65	91.93 ± 2.23	9.96 ± 1.32
MT 432, 03	84.03 ± 2.13	95.21 ± 1.22	11.16 ± 1.28
PHY 183, 02	83.87 ± 1.73	94.09 ± 2.84	10.22 ± 1.92
PHY 183, 03	76.56 ± 1.37	87.14 ± 1.69	9.36 ± 1.14
PHY 231c, 03	80.67 ± 1.32	91.41 ± 2.27	10.74 ± 1.34
PHY 232c, 03	81.55 ± 0.13	92.39 ± 1.58	10.78 ± 1.53
PHY 232, 03	79.77 ± 1.64	88.61 ± 2.45	9.13 ± 2.23
Total Average	78.98 ± 12	90.03 ± 1.30	10.53 ± 56

Features importance

Feature	Importance %
Average Number of Tries	18.9
Total number of Correct Answers	84.7
# of Success at the First Try	24.4
# of Success at the Second Try	26.5
Got Correct with 3-9 Tries	21.2
Got Correct with # of Tries ≥ 10	91.7
Time Spent to Solve the Problems	32.1
Total Time Spent on the Problems	36.5
# of communication	3.6

Conclusion

- Four classifiers used to segregate the students. CMC improves accuracy significantly.
- Weighting the features and using a genetic algorithm to minimize the error rate improves the prediction accuracy by at least 10% in the all cases.
- In the case of the number of features is low, the feature weighting is working better than feature selection.

Contribution

- A new approach to evaluating student usage of web-based instruction
- An approach that is easily adaptable to different types of courses, different population sizes, and different attributes to be analyzed
- Rigorous application of known classifiers as a means of analyzing and comparing use and performance of students who have taken a technical course that was partially/completely administered via the web



Questions

www.lon-capa.org
garage.cse.msu.edu

minaeibi@cse.msu.edu