Presentation of Three Metaphoric Testing papers

Presented by : Ji Gao

[1]Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Overview

1. 1. Properties of Machine Learning Applications for Use in Metamorphic Testing
2. 2. Testing and Validating Machine Learning Classifiers by Metamorphic Testing
3. 3. A survey on metamorphic testing

Outline

Properties of Machine Learning Applications for Use in Metamorphic Testing

Abstract: It is challenging to test machine learning (ML) applications, which are intended to learn properties of data sets where the correct answers are not already known. In the absence of a test oracle, one approach to testing these applications is to use metamorphic testing, in which properties of the application are exploited to define transformation functions on the input, such that the new output will be unchanged or can easily be predicted based on the original output; if the output is not as expected, then a defect must exist in the application. Here, we seek to enumerate and classify the metamorphic properties of some machine learning algorithms, and demonstrate how these can be applied to reveal defects in the applications of interest. In addition to the results of our testing, we present a set of properties that can be used to define these metamorphic relationships so that metamorphic testing can be used as a general approach to testing machine learning applications.

Goal of this paper

Properties of Machine Learning Applications for Use in Metamorphic Testing

Goals according to the paper:

- Verify the quality of machine learning classifier is hard due to the oracle problem.
- Use metamorphic testing to test machine learning models

Contribution: a set of properties that can be used to define metamorphic relationships

Method: **Metamorphic testing** – Alleviate the oracle problem with specially generated inputs.
On training phase.

Metamorphic relations:

- Add/Multiply on all samples shouldn't affect the result.
- Permutative: Order doesn't matter.
- Invertive: Multiply inputs with a negative constant, the result should be reversed
- Inclusive: Predictive pattern when add inputs.
- Exclusive: Predictable when delete inputs.

Experiment

Properties of Machine Learning Applications for Use in Metamorphic Testing

No data, just description.

Applications are found violating several rules.

Case study: Martirank

Properties of Machine Learning Applications for Use in Metamorphic Testing

MartiRank: A supervised ranking learner.
Input: A bunch of devices with several features.
Output: A list of rules to sort the devices
Should hold all the metamorphic relationships.
Failed the test when all data are multiplied by (-1), find a bug according to it.

Outline

Testing and Validating Machine Learning Classifiers by Metamorphic Testing

Abstract: Machine Learning algorithms have provided core functionality to many application domains - such as bioinformatics, computational linguistics, etc. However, it is difficult to detect faults in such applications because often there is no test oracle to verify the correctness of the computed outputs. To help address the software quality, in this paper we present a technique for testing the implementations of machine learning classification algorithms which support such applications. Our approach is based on the technique metamorphic testing, which has been shown to be effective to alleviate the oracle problem. Also presented include a case study on a real-world machine learning application framework, and a discussion of how programmers implementing machine learning algorithms can avoid the common pitfalls discovered in our study. We also conduct mutation analysis and cross-validation, which reveal that our method has high effectiveness in killing mutants, and that observing expected cross-validation result alone is not sufficiently effective to detect faults in a supervised classification program. The effectiveness of metamorphic testing is further confirmed by the detection of real faults in a popular open-source classification program.

Goal of this paper

Testing and Validating Machine Learning Classifiers by Metamorphic Testing

Goals according to the paper:

- Verify the quality of machine learning classifier is hard due to the oracle problem.
- Little work has been done on showing the implementation of machine learning algorithm works correctly.

Our goal: More generally show how the machine learning algorithm works.

Method

Testing and Validating Machine Learning Classifiers by Metamorphic Testing

Method: **Metamorphic testing** – Alleviate the oracle problem with specially generated inputs.
On training phase.

Method

Testing and Validating Machine Learning Classifiers by Metamorphic Testing

Metamorphic relations:

- MR-0: Consistence with affine transformation
- MR-1.1: Permutation of class labels
- MR-1.2: Permutation of the attribute
- MR-2.1: Addition of uninformative attributes
- MR-2.2: Addition of informative attributes
- MR-3.1: Consistence with re-prediction
- MR-3.2: Additional training sample
- MR-4.1: Addition of classes by duplicating samples
- MR-4.2: Addition of classes by re-labeling samples
- MR-5.1: Removal of classes
- MR-5.2: Removal of samples

- Randomly generated data
- Models: k Nearest Neighbor and Naive Bayes Classifier
- Result: Weka implementation of kNN and NBC have chances to violate such metamorphic relations. According to the authors, there are small detailed issues: Categories with 0 samples, Loss of precision, or Choosing labels among the same likelihood.

Table 1
Result of testing *k* NN and NBC.

MR	*k*NN		NBC	
	NP	VP	NP	VP
0	Y	0	Y	7.4%
1.1		15.9%	Y	0.3%
1.2	Y	0	Y	0
2.1	Y	0	Y	0.6%
2.2		4.1%	Y	0
3.1	Y	0		0
3.2	Y	0	Y	0
4.1		25.3%	Y	0
4.2	Y	0		3.9%
5.1		5.9%	Y	5.6%
5.2		2.8%		2.8%

Experiment II

Testing and Validating Machine Learning Classifiers by Metamorphic Testing

Make Mutation on the code - create errors in the code
Some mutants (not all) failed the metamorphic tests.

Table 6
Effectiveness of metamorphic relations for *k* NN.

	Metamorphic relation										
Mutant	0	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2
Original	0	0	0	0	0	0	0	0	0	0	0
v1	0	1.7	7.7	27.0	5.7	0	0	0	0	6.7	3.7
v2	0	0	0	0	0	0	0	0	0	4.7	3.0
v3	0	0	0	0	42.7	0	0	0	0	4.7	3.7
v5	0	2.3	0	0	0	0	0	2.3	0	6.7	3.0
v6	0	11.3	0	26.3	37.0	0	0	0	0	2.0	0
v7	0	9.7	0	0	1.7	0	0	0	0	4.0	1.7
v9	0	9.0	0	0	3.3	8.3	0	41.7	0	5.0	2.0
v10	0	1.3	22.7	34.3	94.7	0	0	0	0	4.3	5.3
v12	0	10.3	0	0	1.7	0	0	0	0	4.0	1.7
v13	0	0.3	22.3	0	0	0	0	0	0	3.0	3.0
v15	0	0	16.3	0	13.7	0	0	0	0	3.3	2.3
v16	0	10.0	0	26.3	37.0	0	0	0	0	2.0	0
v17	0	13.7	0	0	0	0	0	0	0	0	0
v18	0	11.0	0	0	0	0	0	0	0	0	0
v19	0	9.3	0	26.3	37.0	0	0	0	0	2.0	0
v20	0	0	0	0	43.7	0	0	0	0	2.3	1.7
v21	0	1	0	42.7	24.0	0.7	0	0	0	3.7	4.0
v22	0	68.3	0	0	0	0	0	0	0	0	0
v24	0	62.7	0	0	0	0	0	0	0	0	0
Total	0	15	4	6	12	2	0	2	0	15	16

Conclusion: Metamorphic testing is a good supplement of traditional test.

Mutation on the code:

- Some mutants achieved good accuracy on cross-validation test.
- Some mutants (not all) failed the metamorphic tests.

Conclusion: Metamorphic testing is a good supplement of traditional test.

Goal achieved

Buggy code can be found with metamorphic testing?

Outline

A survey on metamorphic testing

Abstract: A test oracle determines whether a test execution reveals a fault, often by comparing the observed program output to the expected output. This is not always practical, for example when a programs input-output relation is complex and difficult to capture formally. Metamorphic testing provides an alternative, where correctness is not determined by checking an individual concrete output, but by applying a transformation to a test input and observing how the program output morphs into a different one as a result. Since the introduction of such metamorphic relations in 1998, many contributions on metamorphic testing have been made, and the technique has seen successful applications in a variety of domains, ranging from web services to computer graphics. This article provides a comprehensive survey on metamorphic testing: It summarises the research results and application areas, and analyses common practice in empirical studies of metamorphic testing as well as the main open challenges.

A survey on metamorphic testing

- Metamorphic testing: Check relations among different executions.
- Steps of metamorphic testing:
  1. Construction of metamorphic relations.
  2. Generation of source test cases.
  3. Execution of metamorphic test cases.

- Find good metamorphic relations: definition, formal description, effective
- Automatically construct metamorphic relations
- Generate a good source test set

Applications

A survey on metamorphic testing

- Web service
- Computer graphics
- Embedded systems
- Simulation and modelling
- Machine learning
- ...

Challenges

- Define good metamorphic relations
- Generate metamorphic relations
- Combine metamorphic relations
- Generate source test cases automatically
- Prioritize metamorphic relations