Automated Evaluation of Programming Assignments

Nikhila K N
Guide: Sujit Kumar Chakrabarti
Co-Guide: Manish Gupta

IIIT Bangalore

Outline

Outline

Introduction

Automated Evaluation of Programming Assignments (AEPA)

- A method of programming assignments evaluation
- A combined application of testing, static analysis and machine learning techniques

Outline

Motivation

Why automated evaluation?

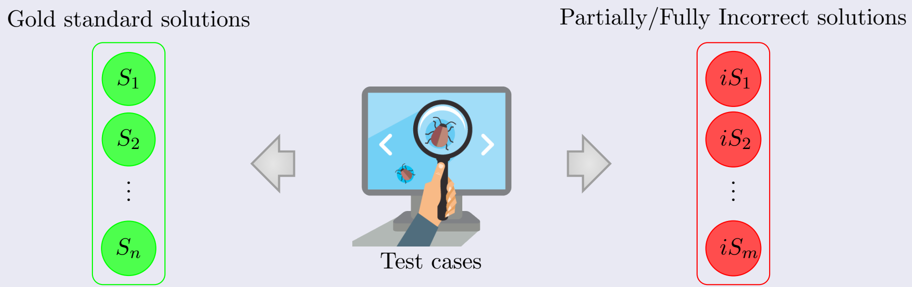
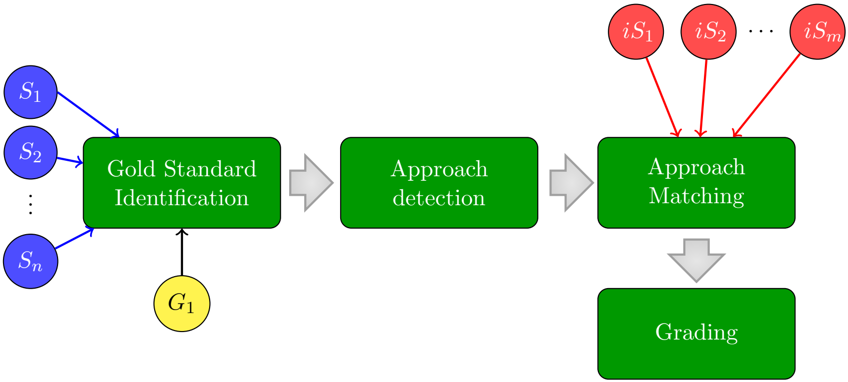


Figure 1: Traditional automated evaluation using test-cases

Outline

Framework


S_1
S_2
S_n
Gold Standard Identification
G_1
Approach detection
iS_1
iS_2
iS_m
Approach Matching
Grading

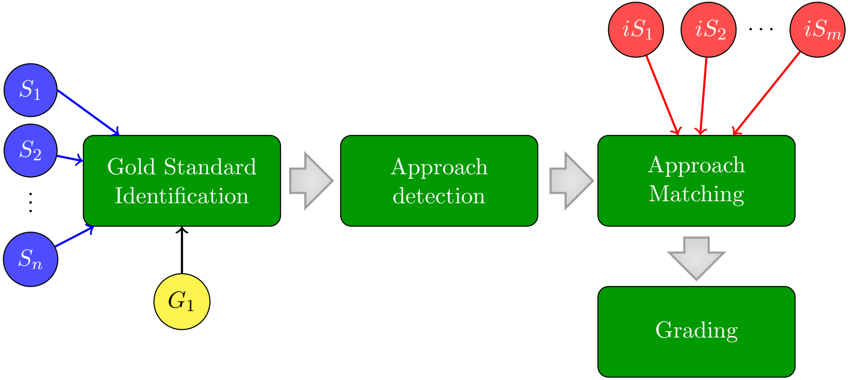
Gold standard Identification

1. Testing based approach is used to identify gold standard solutions

Gold standard Identification

1. Testing based approach is used to identify gold standard solutions
2. Separate the submissions into correct and incorrect using a set of test cases

Framework


iS_1
iS_2
$\cdots$
iS_m
S_1
S_2
$\vdots$
S_n
Gold Standard Identification
Approach detection
Approach Matching
G_1
Grading

Similarity Detection

Types of program similarity [7]

Textual Similarity

```
int sum(int n) {
    int s=0,i;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    return s;
 }
```

```
int sum(int n) {
    int s=0,i;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    return s;
 }
```

Similarity Detection

Lexical/Token based

```
int sum(int n) {
    int s=0,i;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    return s;
 }
```

```
int sum(int limit) {
    int sum=0,k;
    for(k=1;k<=limit;k++)
    {
        sum+=k;
    }
   return sum;
 }
```

Similarity Detection

Syntactical/Structural Similarity

```
int sum(int n) {
    int s=0,i;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    return s;
}
```

```
int sum(int limit) {
    int sum=0,k=1;
    while(k<=limit)
    {
        sum+=k;
        k++;
    }
    return sum;
}
```

Similarity Detection

Semantic Similarity

```
int sum(int n) {
    int s=0,i;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    return s;
}
```

```
int sum(int n) {
  if(n==1)
    return 1;
  else
    return n+sum(n-1);
}
```

Similarity Detection

- Calculate the similarity measure of the Gold standard solutions with each other which is then later used for clustering.

Similarity Detection

- Calculate the similarity measure of the Gold standard solutions with each other which is then later used for clustering.
- The key idea is to consider the following program representations
  - Program dependence graph (PDG)

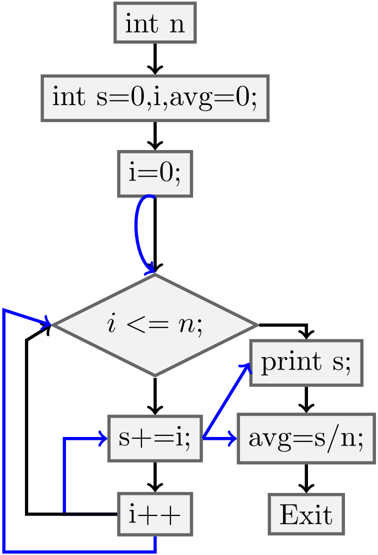
Similarity Detection

- Calculate the similarity measure of the Gold standard solutions with each other which is then later used for clustering.
- The key idea is to consider the following program representations
  - Program dependence graph (PDG)
  - Abstract syntax tree (AST)

Similarity Detection

Program dependence graph (PDG)

```
int average(int n) {
    int s=0,i,avg=0;
    for(i=1;i<=n;i++)
    {
        s+=i;
    }
    print s;
    avg=s/n;
 }
```



Similarity Detection

- γ - Isomorphism

Similarity Detection

- γ - Isomorphism
- Moss (for a Measure Of Software Similarity)

Graph Isomorphism

Formally, the graphs G and G' are isomorphic, then there exists a function $F : V(G) \to V(G')$ such that

i. f is a bijection.

ii. f preserves adjacency of vertices. i.e. if the $(u, v) \in E(G)$ then $(f(u), f(v)) \in E(G')$

Induced Sub-graph & Sub-graph Isomorphism

Let $G = (V, E)$ and $H = (V', E')$. If $H \subseteq G$ and $E' = \{$ all the edges $(u, v) \in E(G) | u, v \in V'\}$, then H is an induced subgraph of G [4]

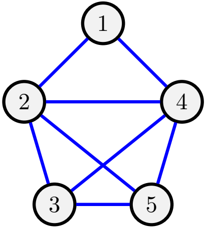


Figure 2: Graph G_1

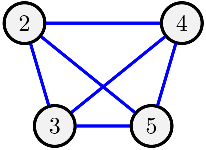


Figure 3: Induced sub graph G_2 of graph G_1

- A graph G is γ-isomorphic to G' if there exists a subgraph $S \subseteq G$ such that S is subgraph isomorphic to G' , and $\|S\| \geq \gamma\|G\|$, $\gamma \in$ (0,1).
- The VF2 algorithm [3] , the Glasgow algorithm [6] and recently the LAD filtering algorithm [5] are used to calculate sub-graph isomorphism.

- Moss is a server system for determining the similarity of programs
- Supports the following languages:
  C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Matlab, VHDL, Verilog, Spice, MIPS assembly, a8086 assembly, a8086 assembly, HCL2.

Approach detection

Why clustering?

- Helps to identify the approach taken by the students

Approach detection

Why clustering?

- Helps to identify the approach taken by the students
- Isolated vertices in the graph indicate unique approach or incorrect solution even though it passes all the test cases

Approach detection

- Louvian community detection [1]
- IPCA Clustering [2]
- Spetcral clustering [8]
- Physics Simulation based Approach to Node Clustering

- The nodes in the graph as point particles in multi-dimensional space

- The nodes in the graph as point particles in multi-dimensional space
- Our hypothesis is that the forces acting on the the particles will eventually lead similar particles to cluster together

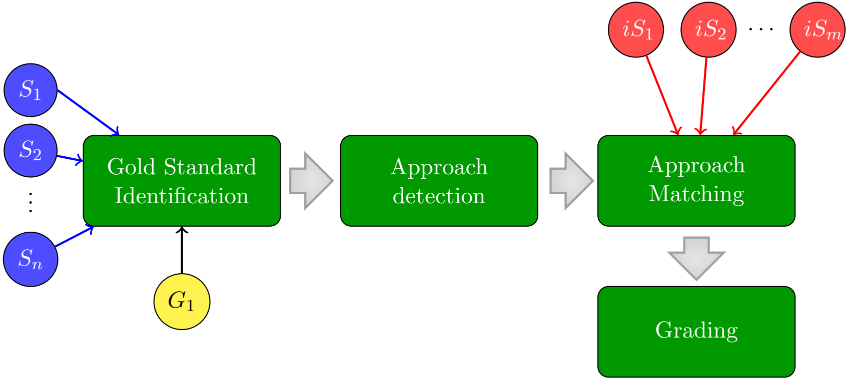
Approach detection

1. Which clustering algorithm works best for identifying approaches?

Approach detection

1. Which clustering algorithm works best for identifying approaches?
2. Is it necessary to carry out clustering on all gold standard solutions to discover all the approaches used in the submitted solutions?

Framework



S_1
S_2
S_n
Gold Standard Identification
G_1
Approach detection
iS_1
iS_2
iS_m
Approach Matching
Grading

Marking

- Find a correlation with manual grading

Marking

- Find a correlation with manual grading
- Studied as a problem of finding which machine learning technique is suitable for finding the best correlation

Marking

Data sets

Pname	Similarity	Clus Rep	Manual grade
S1.c	1	C1.c	4
S2.c	1	C2.c	4
S3.c	0.5	C1.c	2.5
S4.c	1	C3.c	4.5
⋮	⋮	⋮	⋮

Table 1: Structure of data set

Marking

Regression model

- Now, we use a pre-trained regression model R to compute the marks of each member i of I as $M[i] = R[i]$. M is returned as the final result

Marking

Regression model

- Now, we use a pre-trained regression model R to compute the marks of each member i of I as $M[i] = R[i]$. M is returned as the final result

Our experiments investigates the answers to the following questions:

- Which regression modelling works best for training our marking model?

Marking

Regression model

- Now, we use a pre-trained regression model R to compute the marks of each member i of I as $M[i] = R[i]$. M is returned as the final result

Our experiments investigates the answers to the following questions:

- Which regression modelling works best for training our marking model?
- If a regression model is developed using one problem, does it apply to other problems as well?

Marking

Regression model

SVR	C	γ	Accuracy
Linear kernel	1	0.1	35%
rbf kernel	1	0.1	37.5%

Table 2: Experimental result on the simple-array-sum dataset

Outline

Challenges & Future work

1. Data set preparation
2. Efficient similarity calculation
3. Node clustering with large data set
4. Question independent model

Outline

Conclusion

- Black box testing strategy is used for finding Gold standard solutions
- The Node clustering based technique helps to easily identify the approaches taken by the students
- Machine learning based approach is used for grading

References I

Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

Yudong Chen, Sujay Sanghavi, and Huan Xu. "Improved graph clustering". In: *IEEE Transactions on Information Theory* 60.10 (2014), pp. 6440–6455.

Luigi P Cordella et al. "A (sub) graph isomorphism algorithm for matching large graphs". In: *IEEE transactions on pattern analysis and machine intelligence* (2004).

Reinhard Diestel. *Graph Theory, 4th Edition.* Vol. 173. Graduate texts in mathematics. Springer, 2012. ISBN: 978-3-642-14278-9.

References II

Lars Kotthoff, Ciaran Mccreesh, and Christine Solnon. "Portfolios of subgraph isomorphism algorithms". In: *International Conference on Learning and Intelligent Optimization*. Springer. 2016.

Ciaran McCreesh and Patrick Prosser. "A parallel, backjumping subgraph isomorphism algorithm using supplemental graphs". In: *International Conference on Principles and Practice of Constraint Programming*. Springer.

Saul Schleimer, Daniel S. Wilkerson, and Alex Aiken. "Winnowing: Local Algorithms for Document Fingerprinting". In: *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. SIGMOD '03. San Diego, California: ACM, 2003, pp. 76–85. ISBN: 1-58113-634-X. DOI: 10.1145/872757.872770. URL: http://doi.acm.org/10.1145/872757.872770.

References III

Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.