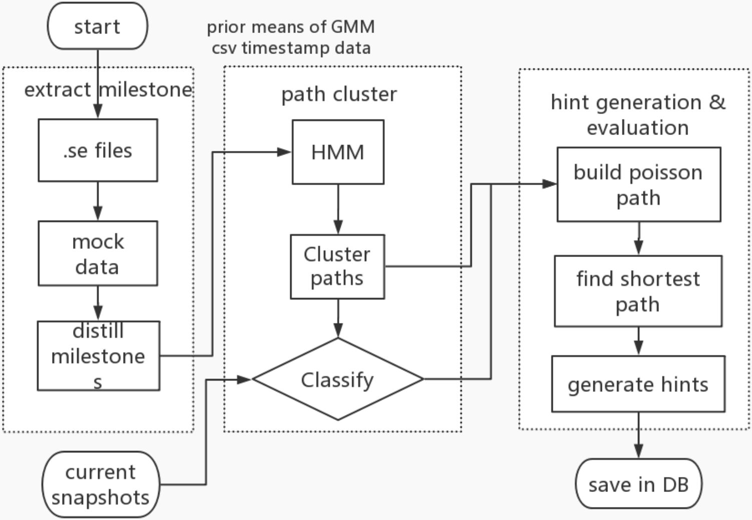
Intelligent Hinting 1.1 in SAGE

Weimeng Luo Junyu Zhang Yi Ding

Programming Systems Lab
Columbia University

May 1, 2018

Introduction



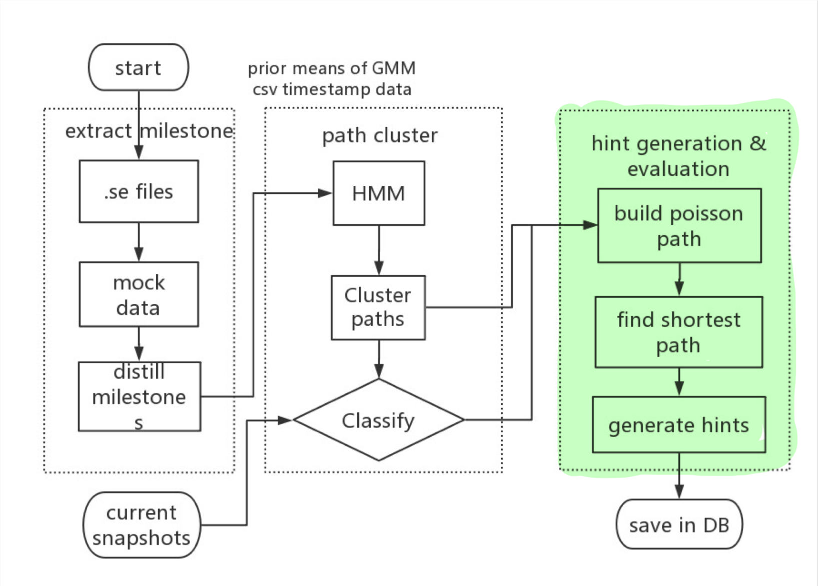
Introduction

Provide Intelligent Hint for Students:`https://youtu.be/YXFs1bOJnz0`
API: /: build graph, /get_hints, /save
'student snapshots' → 'hints'

DEMO

- Cluster Students' Study Path: $'cluster.txt'$
- Build Graph: $'graph_i.dat'$
- Generate Hints: $'hints'$ save in MongoDB

Hint Generation



Build Graph

The node radius and line weight shows how many students come up with this path.

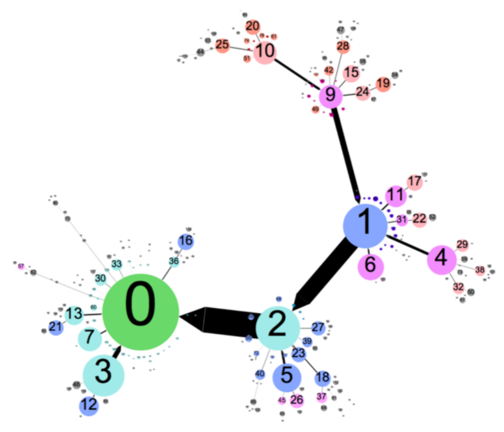


Figure: Transform snapshots of a cluster into graph.[2]

Poisson Path

How to build the learning graph for a type of students?

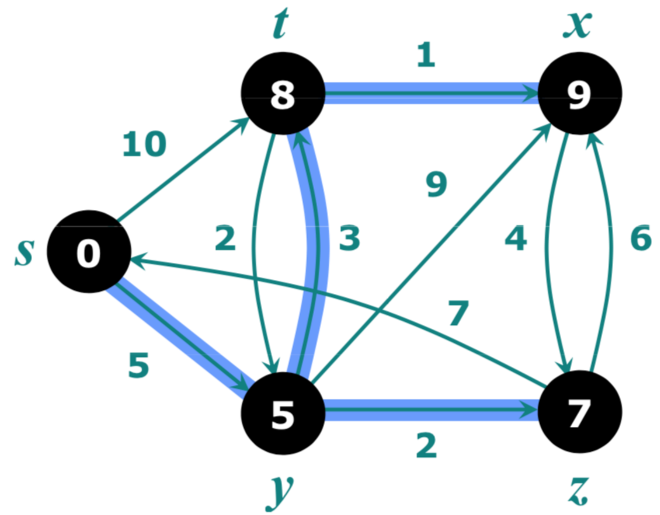
Path Algorithm

- A partial solution is a Poisson process.
- time required to generate a solution $\propto$ # solution from the student's population.

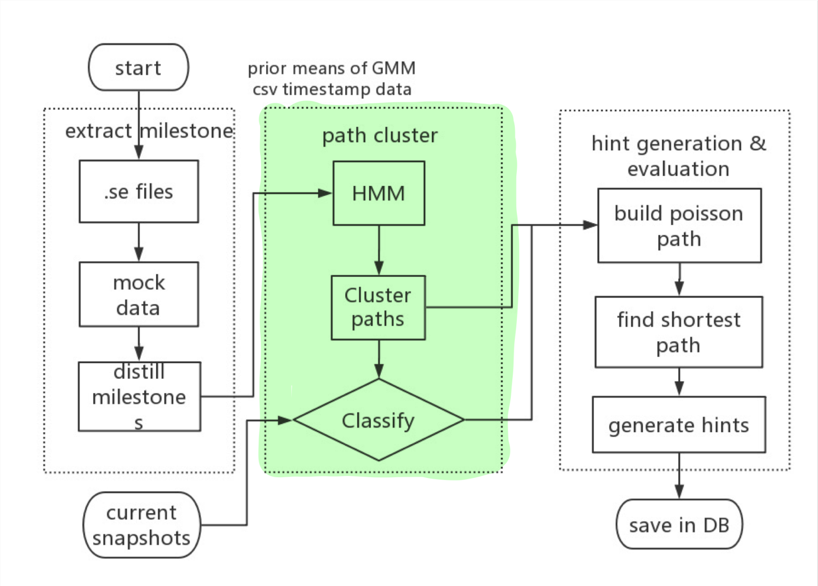
$$\gamma(s) = \operatorname*{argmin}_{p \in Z(s)} \sum_{x \in p} \frac{1}{\lambda}$$

$Z(s)$ are all the paths-to-solution from s and λ_x is the number of times partial solution x is seen in successful student data.

Dijkstra's Algorithm to Find Shortest Path



Hidden Markov Model



Hidden Markov Model

A probabilistic model-based approach to cluster sequences(student path).
Modeling student path as a first-order Markov chain $p(state_{t+1}|state_t)$.

Figure: Example: HMM

- Interpretable student's high-level state
- Find student's learning pattern and cluster them for better hint generation quality

Hidden Markov Model

Students may get trapped in a group of similar snapshots, so we use one "milestone" snapshot to represent those many snapshots and modeling transition process between milestone snapshots.

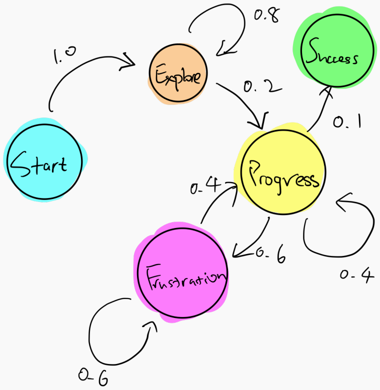


Figure: Example: Hidden state inference of student path

Hidden Markov Model

The number of hidden states K is a hyperparameter, can be determined by cross-validation.
We can calculate transition matrix and emission probability(Gaussian) by dynamic programming.

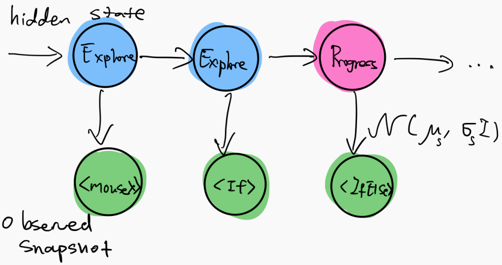


Figure: Example: Hidden state inference of student learning path

Hidden Markov Model

Snapshots are huge, we want a "median" snapshot to represent a group of similar snapshots.
Distill: We use K-Medoids clustering to find "median" snapshots among the whole project, as the prior mean of our hidden Markov model.

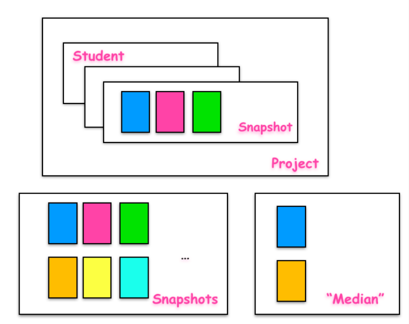


Figure: Example: Milestone calculation. We use Euclidean distance among snapshots.

Clustering

Now we have generated per-student model, then we can calculate (symmetric) pairwise distance between two student's study path.

Model distance

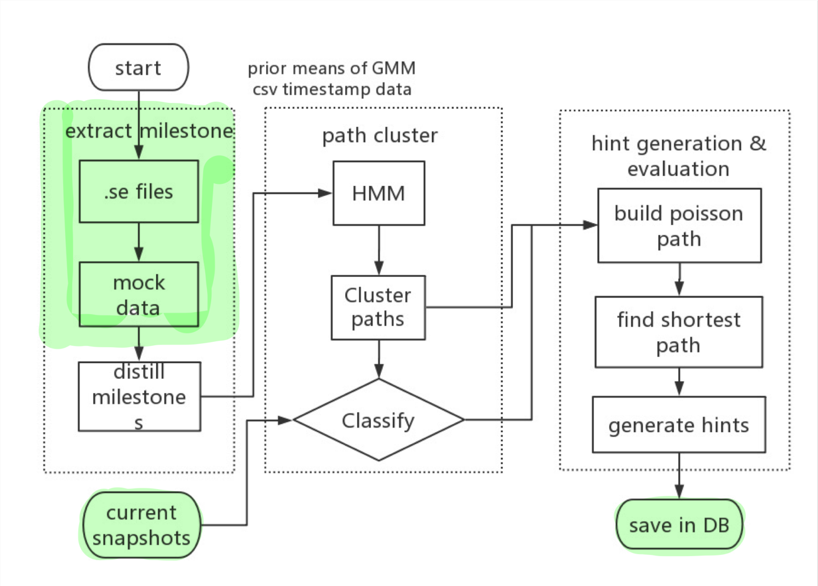
Log-likelihood score for model i and sequence j is $\log L_i(j)$.

$$score_i(j) = \log L_i(j)$$

$$dist(i,j) = \frac{score_i(j) + score_j(i)}{2}$$

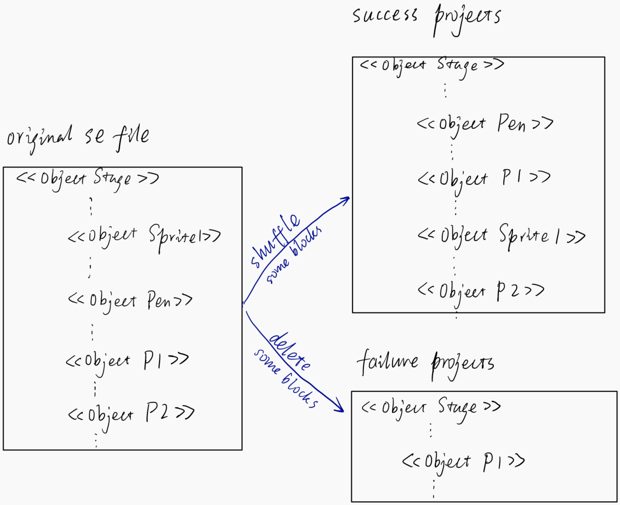
Finally, we use spectral clustering method to cluster all students into K groups.

Experiment & Integration



Mock data generation

Generate fake projects: success, failure



Mock data generation

Four kinds of learners

- **Extreme movers(1)**
  Move too fast, retry unworkable approach
- **Movers(2)**
  Consistently try new idea, never stop long
- **Stoppers(3)**
  Stop long to appear stuck
- **Tinkers(4)**
  Move forward and then backward consistently

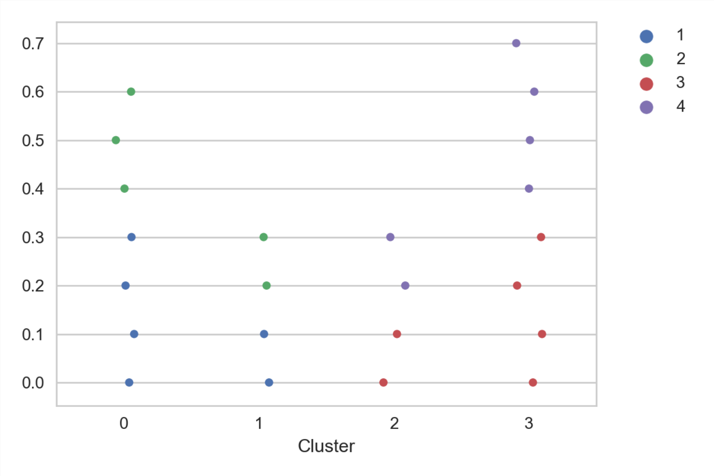


Figure: Cluster result

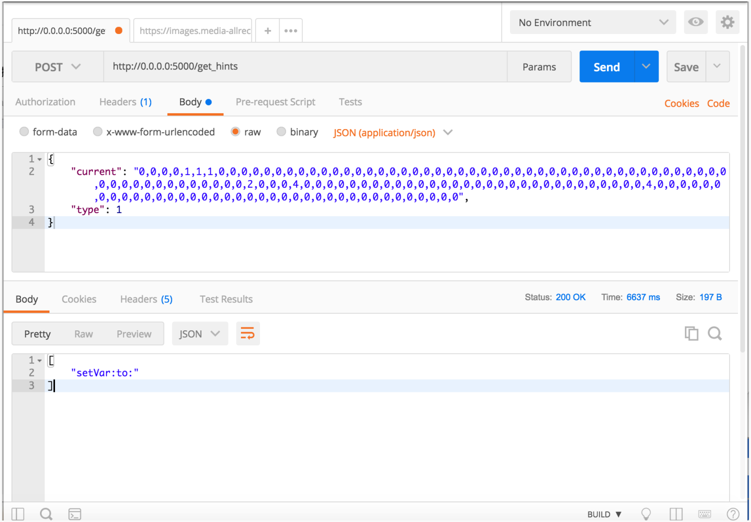
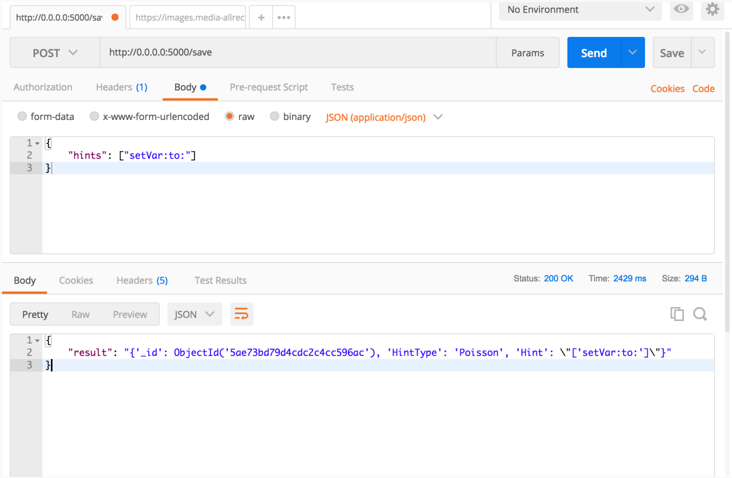


Figure: Generating hints



http://0.0.0.0:5000/sa
https://images.media-allrec
No Environment
POST
http://0.0.0.0:5000/save
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"hints": ["setVar:to:"]
}
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 2429 ms
Size: 294 B
Pretty
Raw
Preview
JSON
{
"result": "{'_id': ObjectId('5ae73bd79d4cdc2c4cc596ac'), 'HintType': 'Poisson', 'Hint': \"['setVar:to:']\"}"
}

Future work

- Use a series of snapshots instead of one.
- Cold-start problem for new incoming students.
- Deep integration into SAGE system.

Reference

D. N. Perkins, Chris Hancock, Renee Hobbs, Fay Martin, and Rebecca Simmons.
Conditions of learning in novice programmers.
Journal of Educational Computing Research, 2(1):37–55, 1986.

Chris Piech, Mehran Sahami, Jonathan Huang, and Leonidas Guibas.
Autonomously generating hints by inferring problem solving policies.
In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, pages 195–204, New York, NY, USA, 2015. ACM.

Chris Piech, Mehran Sahami, Daphne Koller, Steve Cooper, and Paulo Blikstein.
Modeling how students learn to program.
In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, pages 153–160, New York, NY, USA, 2012. ACM.