Modeling Skill Combination Patterns for Deeper Knowledge Tracing

Yun Huang[1], Julio Guerra[2,3], Peter Brusilovsky[1,2]

[1]Intelligent Systems Program, University of Pittsburgh

[2]School of Information Sciences, University of Pittsburgh

[3]Instituto de Informática, Universidad Austral de Chile

Learner Knowledge Model in Complex Tasks

- Many KCs per problem
- KCs interdependence
- Difficulty change due to special KC combinations
- Mastery: fluently apply skills in combinations

```
public class Tester {

  public static void main(String[] args) {

    int[][] array = new int[4][4];

    for (int i = 0; i < array.length ; i++)
      for ( int j = 0 ; j < array.length ; j++)
        array[i][j] = i + j;

    int result = array[1+1][1-1] ;
  }
}
```

What is the final value of **result**?

Submit

Related Work

- Mostly rely on expert engineering
  → Time-consuming and corpus-dependent
- Automatic approaches: extract concepts as KC
  → Assume KC independence: inaccurate
  → Doesn't capture skills (strategies) well
- Overlooking KC and student modeling in programming
  → Doesn't support transparent recommendation well
  → Doesn't give a good sense of students' expertise level

Proposed Approach: data-driven model

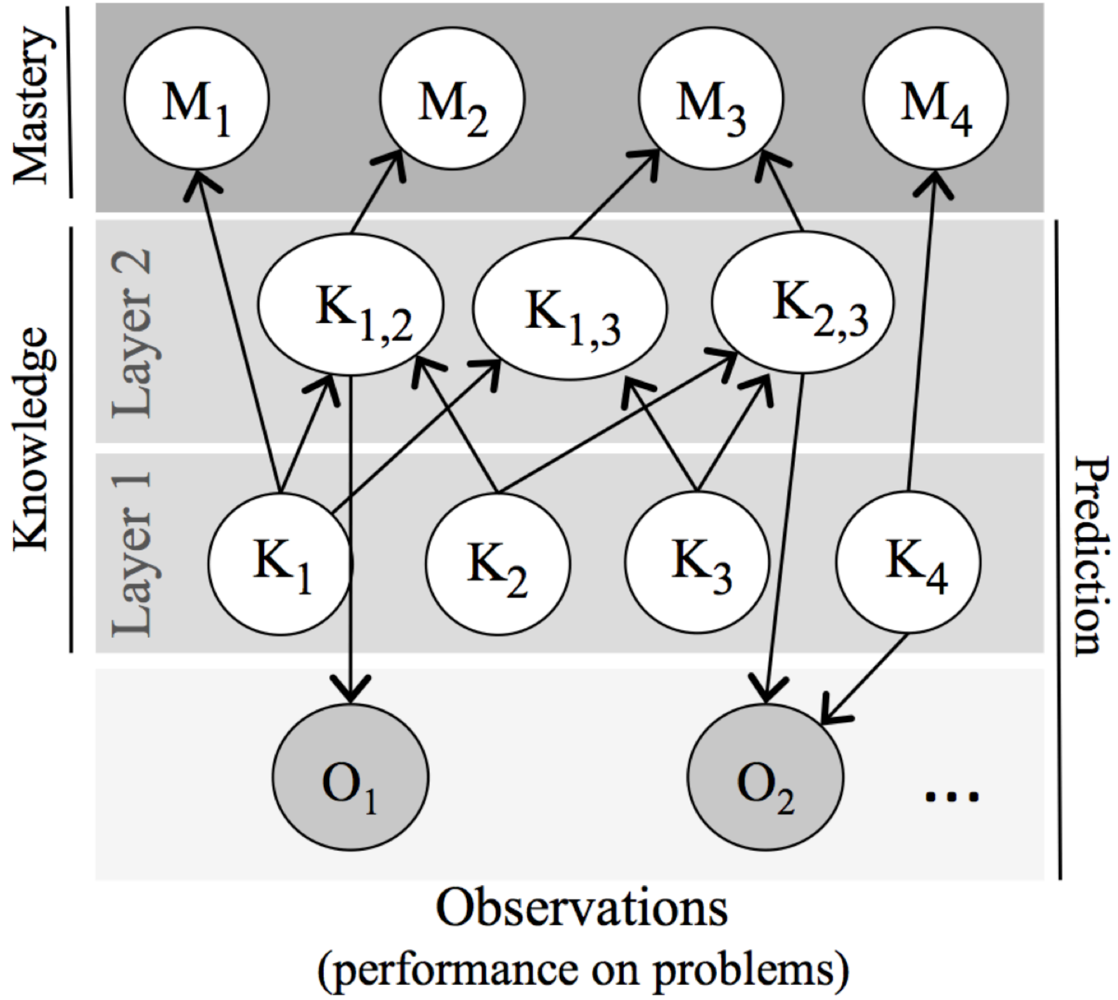
Conjunctive Knowledge Modeling with Hierarchical Skill Combinations (CKM-HSC)

individual mastery-skills
(e.g., Mastered ForStatement)

skill combinations
(e.g., ForStatement + ArrayElement)

individual skills
(e.g., ForStatement)

Multi-level Bayesian network



Model Construction

Skill combinations (Layer 2) are selected if:

- Skill combination is *much more difficult* than each of its individual skills.
- Skill combination difficulty is *high*.
- *Difficult* problems (items) likely require skill combinations.
- Each problem has a limited number of skill combinations.

Network structure is learned using a greedy search algorithm (we proposed a simplified version using empirical pruning with higher efficiency).

Model Evaluation

- Multifaceted data-driven evaluation framework that includes:
    - Knowledge inference quality:
        - *Mastery Accuracy*: Do students mostly have correct responses on the data after a student model infers mastery?
        - *Mastery Effort*: How many practices does a student need to reach inferred mastery for all required skills on the data?
    - Parameter plausibility: Item Discriminative Index
    - Performance prediction accuracy: RMSE, AUC
- These metrics extend our recent evaluation frameworks LEOPARD [5] and Polygon [7].

Study

1. **Is proposed skill combination incorporated model better than traditional KT models?** Yes!
2. **Is using hierarchy better than independence for incorporating skill combinations?** Yes! (See paper)
3. **Can we improve modeling by adding external knowledge for skill combination extraction?** Yes! (See paper)

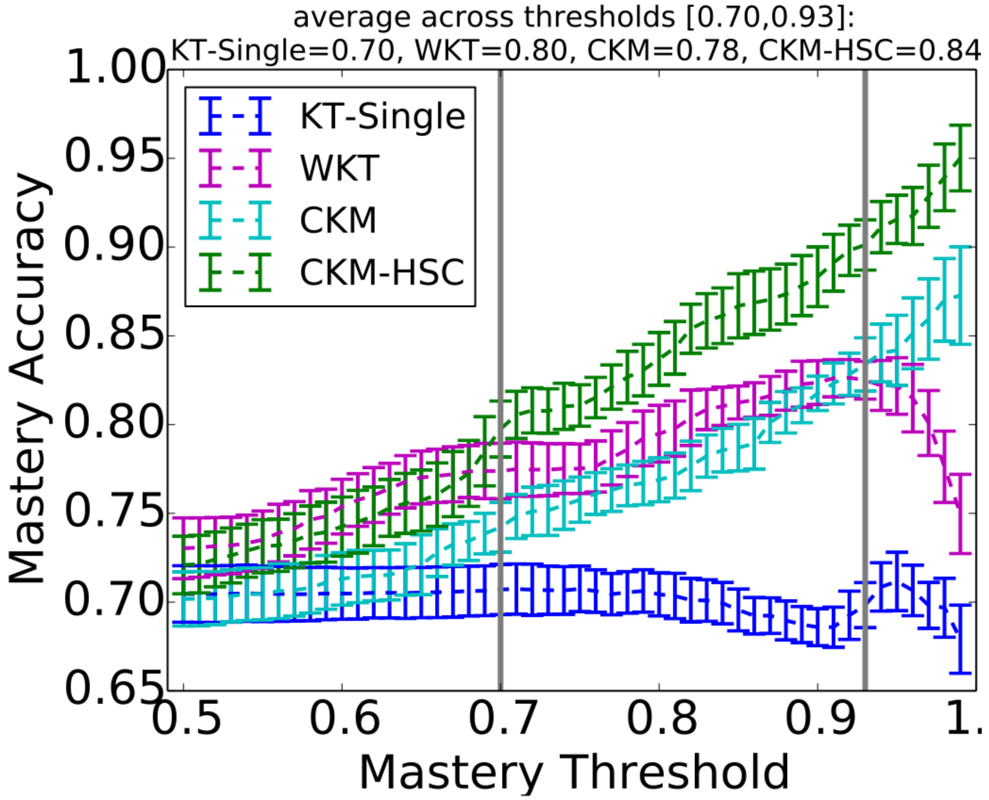
Dataset

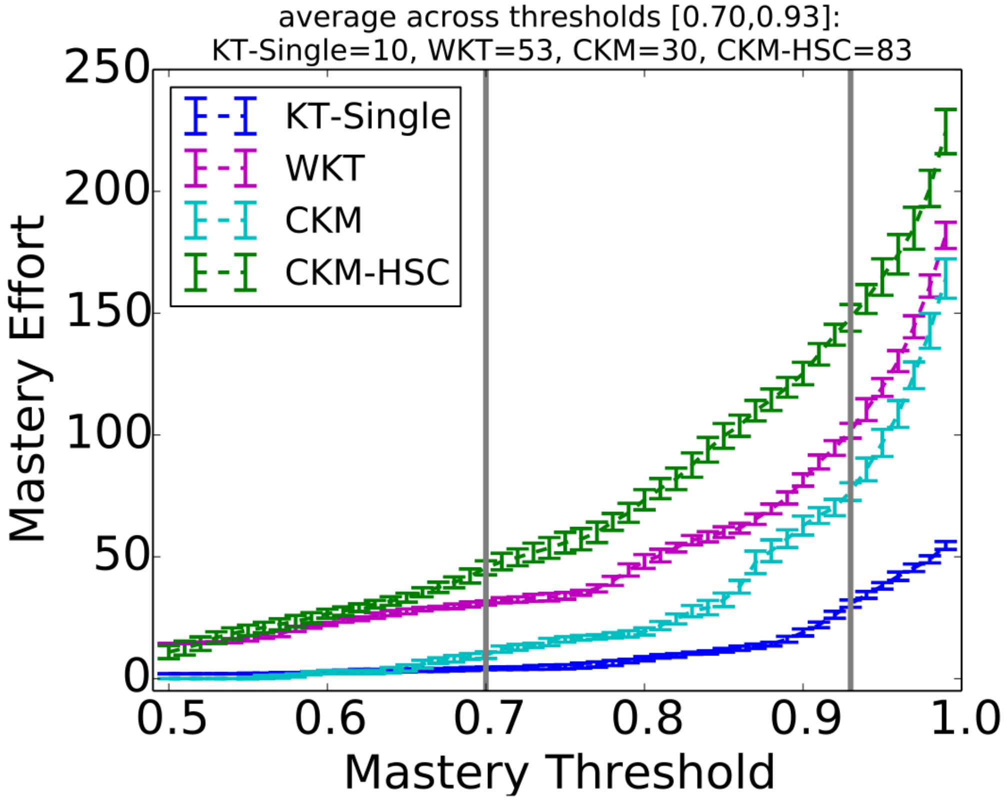
Dataset	#obs.	#items	#skills	avg #skills/item	#users	%correct
SQL	17,197	45	34	5 (from 1 to 10)	366	58%
Java	25,988	45	56	5 (from 1 to 11)	347	67%

Results

Is skill combination model better than traditional KT models?

- **Increases the mastery inference accuracy**
- **Requires students to focus more on skill combinations**





Future Work

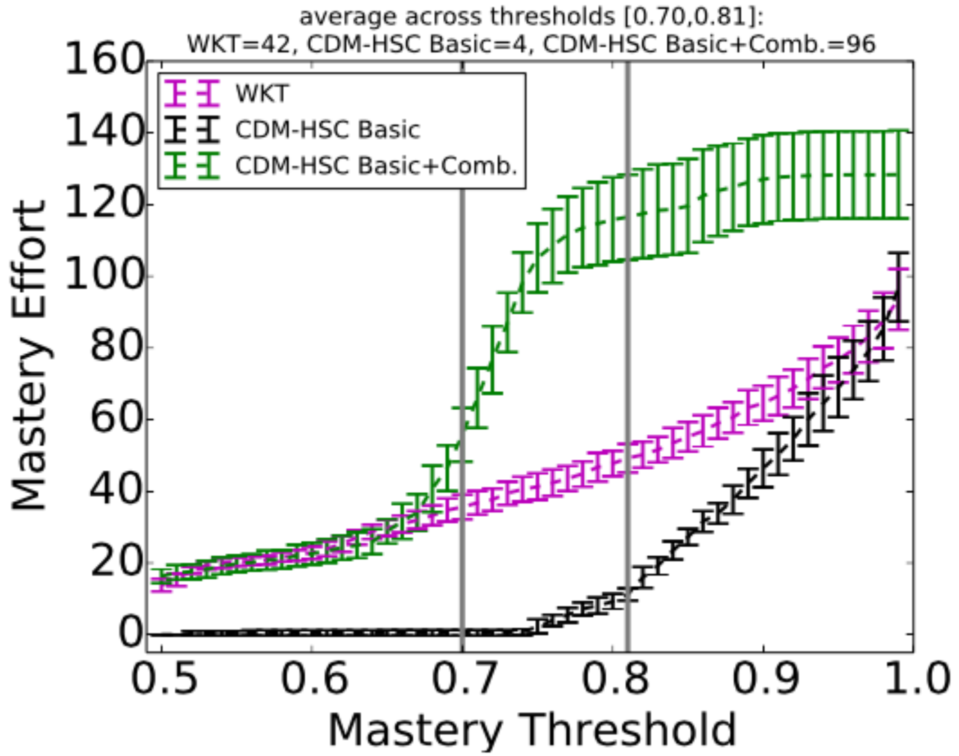
- consider more complex skill combinations
- explore more efficient implementation tool and new techniques for learning the structure
- collect more suitable datasets
- open student models and remediation

Acknowledgement

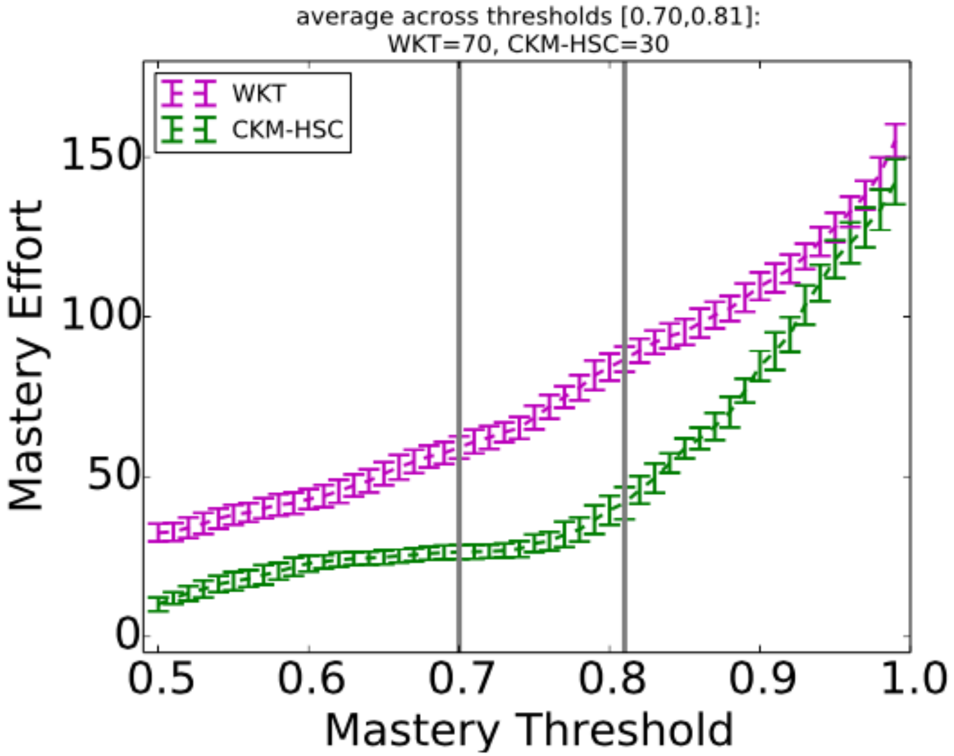
- Advanced Distributed Learning Initiative (http://www.adlnet.gov/) and the National Science Foundation Cyber-Human Systems (CHS) Program under Grant IIS-1525186.
- National Science Foundation support for UMAP students

Thank you very much!

- **more reasonably distributes students' efforts**: requiring students to focus more on skill combinations

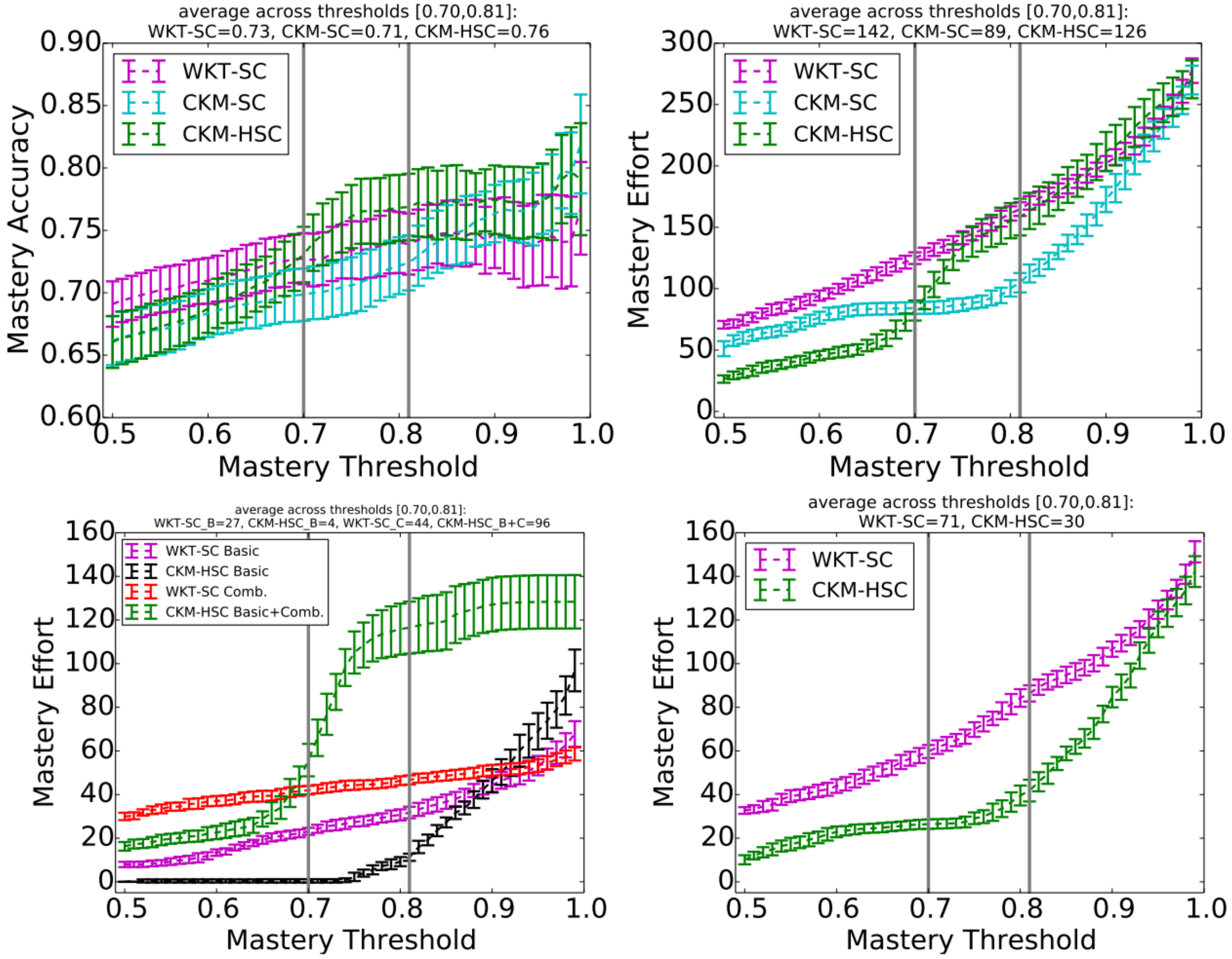


(a) Skills requiring knowing combined skills



(b) Skills not requiring knowing combined skills

Is using hierarchy better than independence for incorporating skill combinations?



(a) Skills requiring knowing combined skills

(b) Skills not requiring knowing combined skills

Can we improve modeling by adding external knowledge for skill combination extraction?

Table 3: Comparison among CKM-HSC and alternatives adding external knowledge on Java (average across 10 folds with 95% CI; mastery metrics computed on [0.7, 0.93]).

Models	M.Acc	M.Effort	IDI	RMSE	AUC
CKM-HSC	.84	83	.46(.01)	.45(.01)	.73(.01)
CKM-HSC-P	.82	70	.45(.01)	.45(.01)	.73 (.01)
CKM-HSC-P-E	.82	59	.42(.01)	.45(.01)	.73 (.01)