What can we Learn from Student Course-Grade Data?

GARY M. WEISS
EDUCATIONAL DATA MINING (EDM) LAB
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
FORDHAM UNIVERSITY, NY, USA

gaweiss@fordham.edu

General Motivation & Goals

- Data science being applied to many domains, but the education domain seems to receive less attention
- Course-grade enrollment data tracked by every university
  - Each record describes one student course enrollment and final grade
- What can we learn from the course enrollment grade data?

What Questions can we Answer using only Course Enrollment Data?

- Grading:
  - Q1: What are the grading patterns (policies) at the department, course, and instructor levels? Do they vary substantially?
- Course Sequencing:
  - Q2: What are the most common sequences of courses that students take?
  - Q3: How does number semesters between courses impact performance?
  - Q4: How does the order of taking courses impact performance?
- Q5: Instructors: How effective is an instructor (based on future student performance)?
- Q6: Student Majors: What major will a student perform well in? Choose?
- Course relationships:
  - Q7: How can we group courses based on *similar student performance (i.e., grades)*?
  - Q8: What can we learn about courses based on *co-enrollment patterns*?

What Questions can we Answer using only Course Enrollment Data?

- Grading:
  - Q1: What are the grading patterns (policies)?
    - At department, course, and instructor levels. Do patterns vary substantially?
- Course Sequencing:
  - Q2: What are the most common sequences of courses that students take?
  - Q3: How does number semesters between courses impact performance?
  - Q4: How does the order of taking courses impact performance?
- Q5: Instructors: How effective is an instructor (based on future grades)?
- Q6: Student Majors: What major will a student perform well in?* Choose?
- Course relationships:
  - Q7: How can we group courses based on *similar student performance (i.e., grades)*?
  - Q8: What can we learn about courses based on *co-enrollment patterns*?

Description*
(Unsupervised
Learning)

Less focused and not easy
to assess utility of the results

** Identifying which major a student will perform well in is the one exception and is a prediction task*

The Data

- Eight years Fordham University undergraduate course data
  - ~10,000 undergraduate students per year
- Each record represents one student in one class
  - Course: name, dept. code, course #, section #, semester, year
  - Instructor: Instructor id (anonymized)
  - Student: Student id (anonymized)
  - Final grade
- What we don't have due to privacy/legal concerns:
  - Student info. (race, gender, parent income, SAT scores, etc.)
  - Instructor info. (rank, race, gender, years of experience, surveys)

Type	Count
Records	446,508
Students	24,691
Unique Courses	2,663
Course Sections	22,608
Student Majors	83
Departments	78

Q1: What Grading Patterns Exist?

- Motivation:
  - Undergraduate grades are important:
    - Provide feedback/motivation to student
    - Used for "admission" to majors, graduate programs, jobs
  - Conflict of interest: grades impact student assessment of instructors
  - We want accurate and fair grades
    - Major deviations in grading may be considered unfair
- Goals:
  - Identify patterns at the department, course, and instructor level.
  - Analyze the patterns– are there big differences? (Answer: yes)
  - Provide an open-source software tool for general use and research
    - https://www.cis.fordham.edu/edmlab/software/grade-analysis-tool

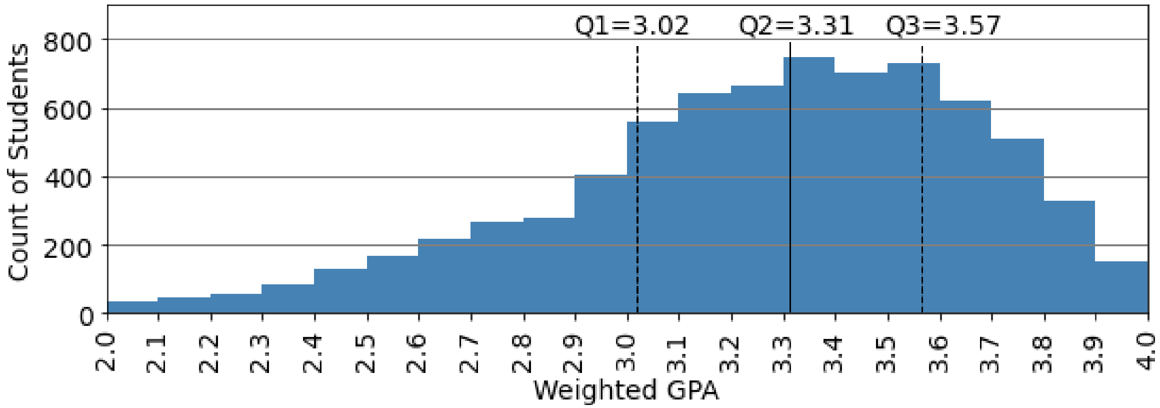
Gary M. Weiss, Luisa A. L. Rosa, Hyun Jeong and Daniel D. Leeds (2023). An Analysis of Grading Patterns in Undergraduate University Courses. *Proceedings of the 2023 IEEE 467h Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, Torino, Italy.

General Grading Trends

Average Grade by Student Year and Course Level

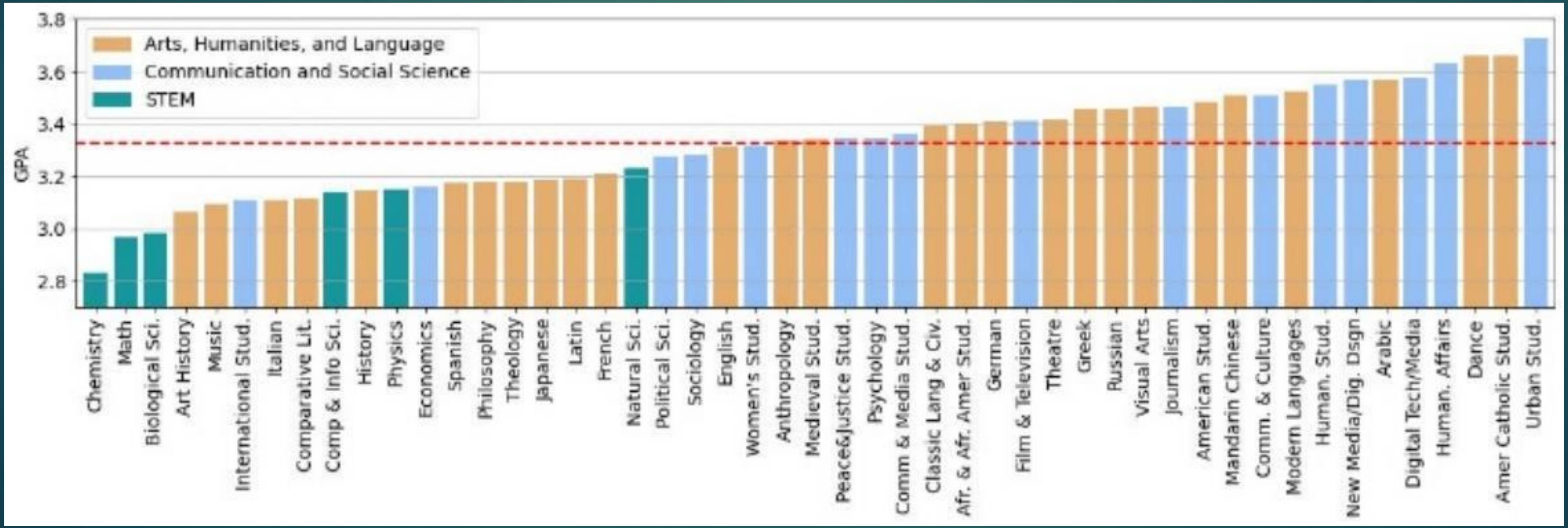
	Course Level				
Student Year	1000	2000	3000	4000	Average
Freshman	3.109	3.276	3.037	3.235	3.122
Sophomore	3.190	3.284	3.217	3.257	3.224
Junior	3.169	3.313	3.281	3.322	3.264
Senior	3.182	3.348	3.333	3.411	3.331
Average	3.137	3.305	3.275	3.389	

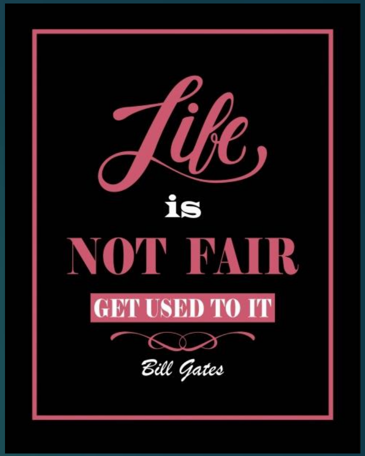
Student Grade Distribution (μ=3.25)



GPA by Department

- Large variation by department
- STEM departments have lowest grades (consistent with research)
- Research: instructors teaching in multiple department adhere to department grading pattern

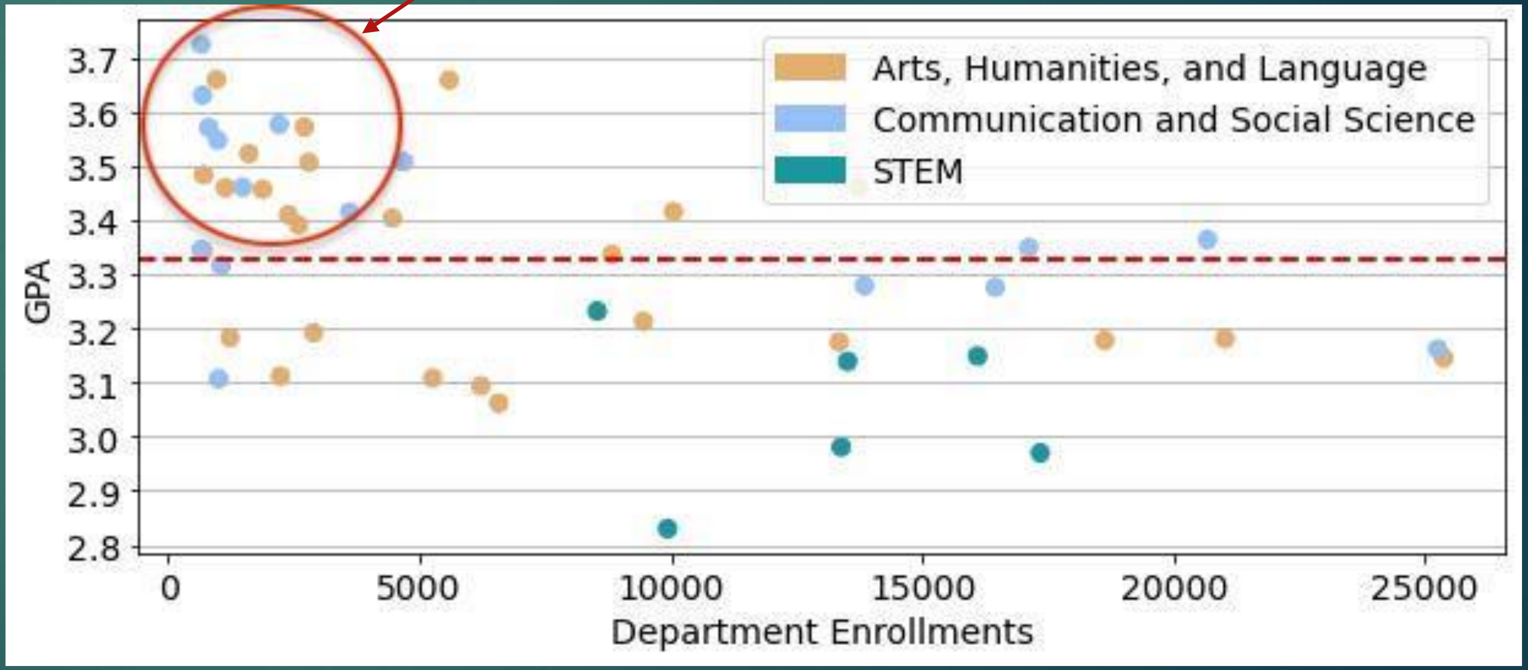




Department GPA vs. Enrollment

- Hypothesis: inverse relationship between dept. enrollment and GPA
- Conclusion: GPA impacted mainly when enrollments are small
  - No big difference between medium and large departments
  - Of 21 highest GPA departments, 19 have enrollments under 5000
  - No department with enrollment over 5000 has GPA > 3.5
  - High GPA → Low Enrollment
    - Low GPA <u>not</u> related to enrollment

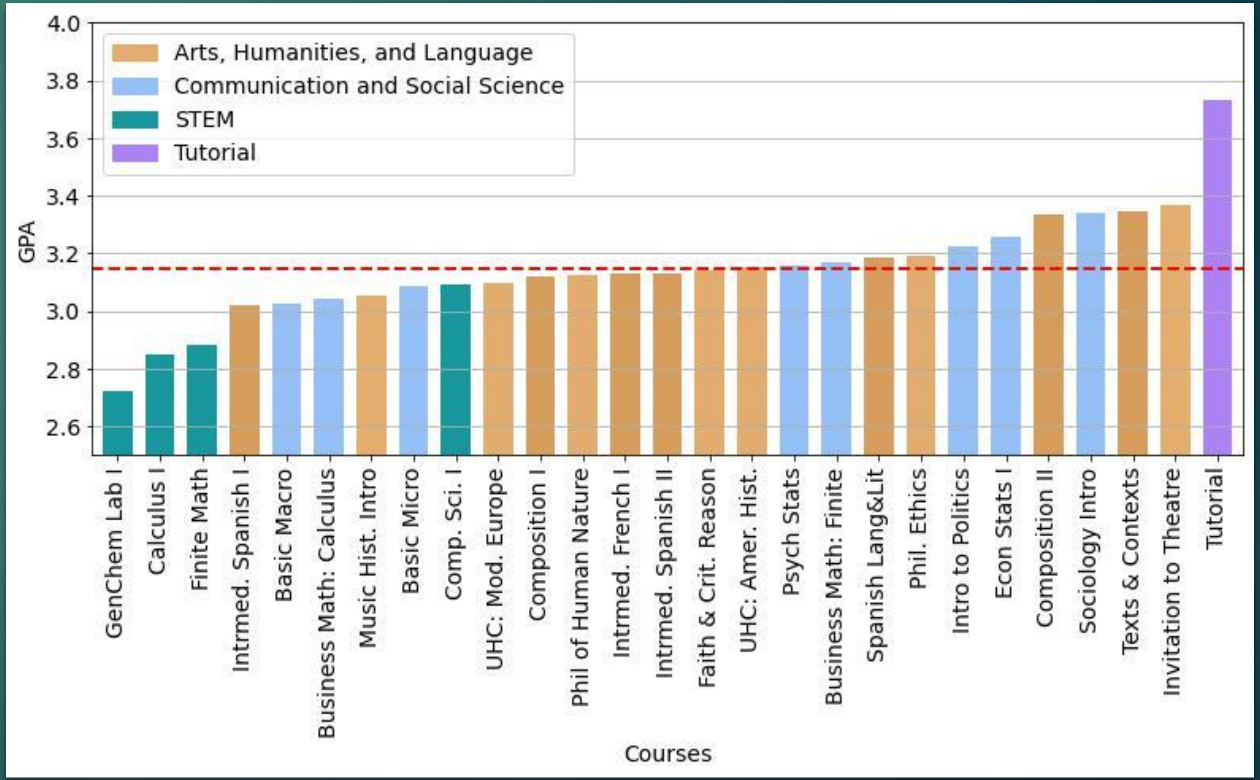
Low enrollment/High GPA



GPA by Course

- Large variation at course level
- GPA pattern follows department
  - Most are lower than department
  - Popular courses have lower grades
- Tutorial courses
  - Insufficient students for course
  - Small sections with high grades
  - Why?
    - Too few to establish distribution
    - Familiarity with students
    - Maybe more effective teaching?

GPA by Course for 27 Courses with at least 70 Sections



Course Grade Distribution Patterns

Average Course Grade Distribution by Cluster (k-means k=4)

	Letter Grade (%)									
Cluster	A	A-	B+	B	B-	C	D	F	GPA	Count
0	**27.7**	**14.6**	13.2	14.2	8.8	16.1	3.1	2.3	3.11	58
1	40.6	23.0	13.6	10.4	5.0	5.6	0.7	1.0	3.49	47
2	**20.6**	**22.9**	20.0	16.0	8.6	9.6	1.2	1.2	3.25	71
3	12.9	12.9	15.6	18.3	13.3	22.6	2.7	1.6	2.91	45

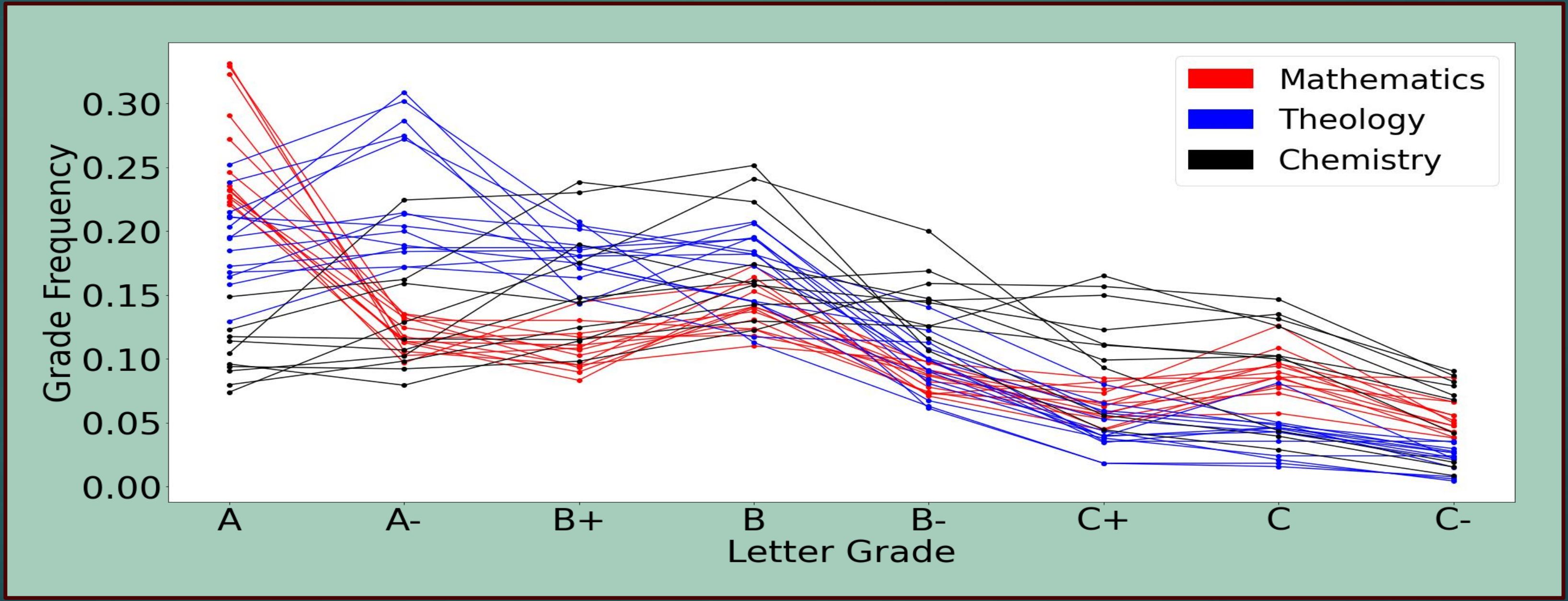
- 221 courses with total enrollments over 300
  - Grade distribution vectors formed for each
  - K-means clustering with k=4 run
  - Cluster 3 only bell-shaped distribution
    - Only traditional sciences (Bio/Chem/Physics)
  - Cluster 0 and 2 similar GPA, different A/A- behavior
    - Math has all high enrollment courses in cluster 0

Course Cluster Distribution by Department

	Cluster				
Department	**0**	**1**	**2**	**3**	**Total**
Biological Sciences	1	1	5	**7**	14
Chemistry	0	0	2	**8**	10
Communications	2	8	12	1	23
Computer Science	**7**	1	2	3	13
Economics	**8**	1	2	3	14
English	0	0	**5**	0	5
History	1	1	**6**	2	10
Mathematics	**13**	0	0	0	13
Natural Science	4	6	2	4	16
Philosophy	0	0	**3**	0	3
Physics	1	2	0	**6**	9
Psychology	6	8	1	0	15
Spanish	1	0	**4**	1	6
Theology	0	0	**12**	2	14

Individual Course Grade Distributions

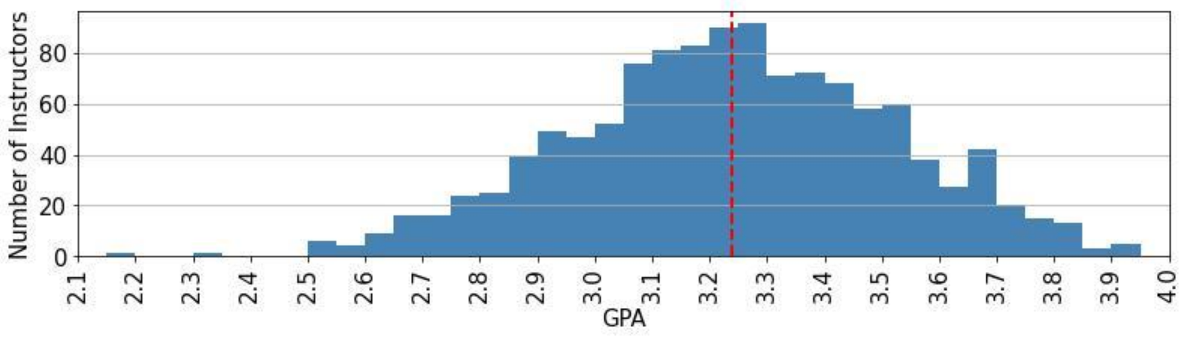
- Grade Distributions shown for each popular course in three departments
- Each line represents one course (parallel coordinates)
- Substantial amount of consistency (not just different averages)
- Look at the distribution of A versus A- in math courses!



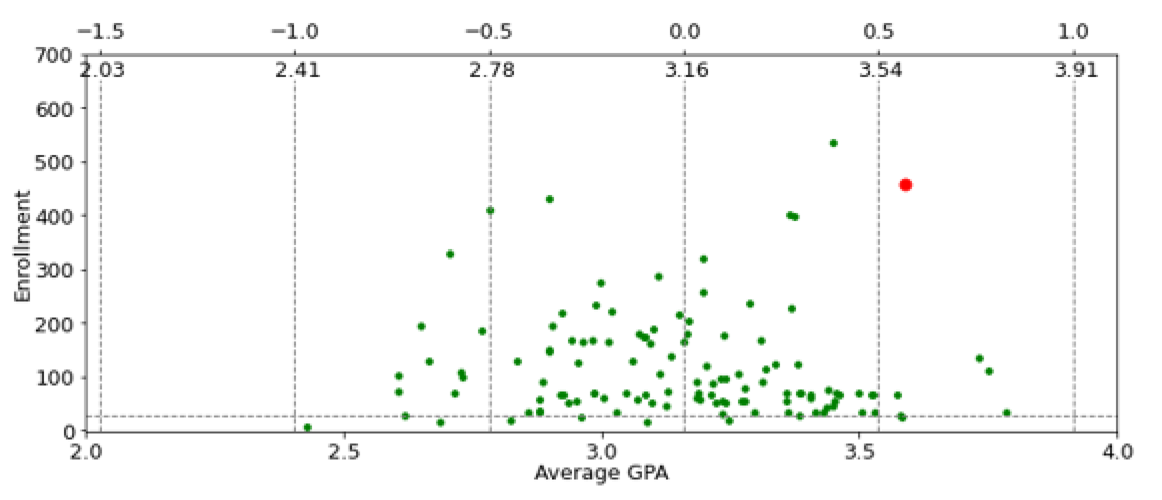
Instructor Grading Patterns

- Substantial spread in instructor grading
- The lower figure shows the distribution of individual instructors for a single course
  - Y-axis shows total enrollment per instructor (can ignore those with few students)
  - One instructor more than 0.5 standard deviations above mean; many below
- Not shown is table of extreme graders
  - One has 2.08 GPA over 248 students
  - One has 3.85 GPA over 195 students

Instructor GPA Distribution (minimum 6 sections)



Instructor GPA Distribution (Faith and Critical Reasoning)



Course Sequencing Questions

- Q2: What are the most common sequences of courses that students take?
  - Better understand curricula and how students take courses
  - Inform course scheduling and advising
- Q3: How does number semesters between courses impact performance?
  - Magnitude of impact may inform us about relationship between courses
  - Improved advising
- Q4: How does the order of taking courses impact performance?
  - Identifying optimal ordering can inform advising
  - Could suggest prerequisites
  - May suggest relationships between courses thought to be unconnected

Q2: What are Most Common Course Sequences?

- Variation of association analysis
  - Finds frequent sequences rather than frequent itemsets (i.e., order matters).
  - Uses Generalized Sequential Pattern (GSP) mining algorithm, an extension to Apriori algorithm that considers order.
  - Raw data transformed so each entry ordered list of courses for one student
- We have a Python-based open source tool that will run GSP on course enrollment data to find frequent sequences
  - https://www.cis.fordham.edu/edmlab/software

Daniel Leeds, Cody Chen, Yijun Zhao, Fiza Metla, James Guest, and Gary Weiss. Generalized Sequential Pattern Mining of Undergraduate Courses. *Proc. of 15th Int. Conference on Educational Data Mining (EDM22)*, Durham, UK, July 24-27, 2022.

Computer Science Frequent 5-Sequences
(minimum support = 50)

Index	Computer Science Frequent 5-Sequence	Support	Index	Computer Science Frequent 5-Sequence	Support
1	Discrete Struct, CS1, CS2, Data Struct, Theory of Comp	50	13	CS1, CS2, Data Struct, Theory of Comp, Data Mining	63
2	CS1, CS2, Data Struct, Databases, Operating Sys	67	14	CS1, CS2, Data Struct, Data Comm and Net, Theory of Comp	55
3	CS1, CS2, Data Struct, Databases, Comp Alg	60	15	CS1, CS2, Databases, Operating Sys, Theory of Comp	80
4	CS1, CS2, Data Struct, Databases, Theory of Comp	73	16	CS1, CS2, Databases, Comp Alg, Theory of Comp	58
5	CS1, CS2, Data Struct, Comp Org, Operating Sys	52	17	CS1, CS2, Comp Org, Data Struct, Operating Sys	54
6	CS1, CS2, Data Struct, Comp Org, Comp Alg	60	18	CS1, CS2, Comp Org, Data Struct, Theory of Comp	52
7	CS1, CS2, Data Struct, Comp Org, Theory of Comp	52	19	CS1, CS2, Comp Org, Operating Sys, Theory of Comp	73
8	CS1CS1, CS2, Data Struct, Operating Sys, Comp Alg	70	20	CS1, CS2, Comp Org, Comp Alg, Theory of Comp	59
9	CS1, CS2, Data Struct, Operating Sys, Theory of Comp	110	21	CS1, CS2, Operating Sys, Theory of Comp, Data Mining	50
10	CS1, CS2, Data Struct, Operating Sys, Data Mining	61	22	CS1, Data Struct, Databases, Operating Sys, Theory of Comp	56
11	CS1, CS2, Data Struct, Comp Alg, Theory of Comp	84	23	CS2, Data Struct, Databases, Operating Sys, Theory of Comp	56
12	CS1, CS2, Data Struct, Comp Alg, Data Comm and Net	55			

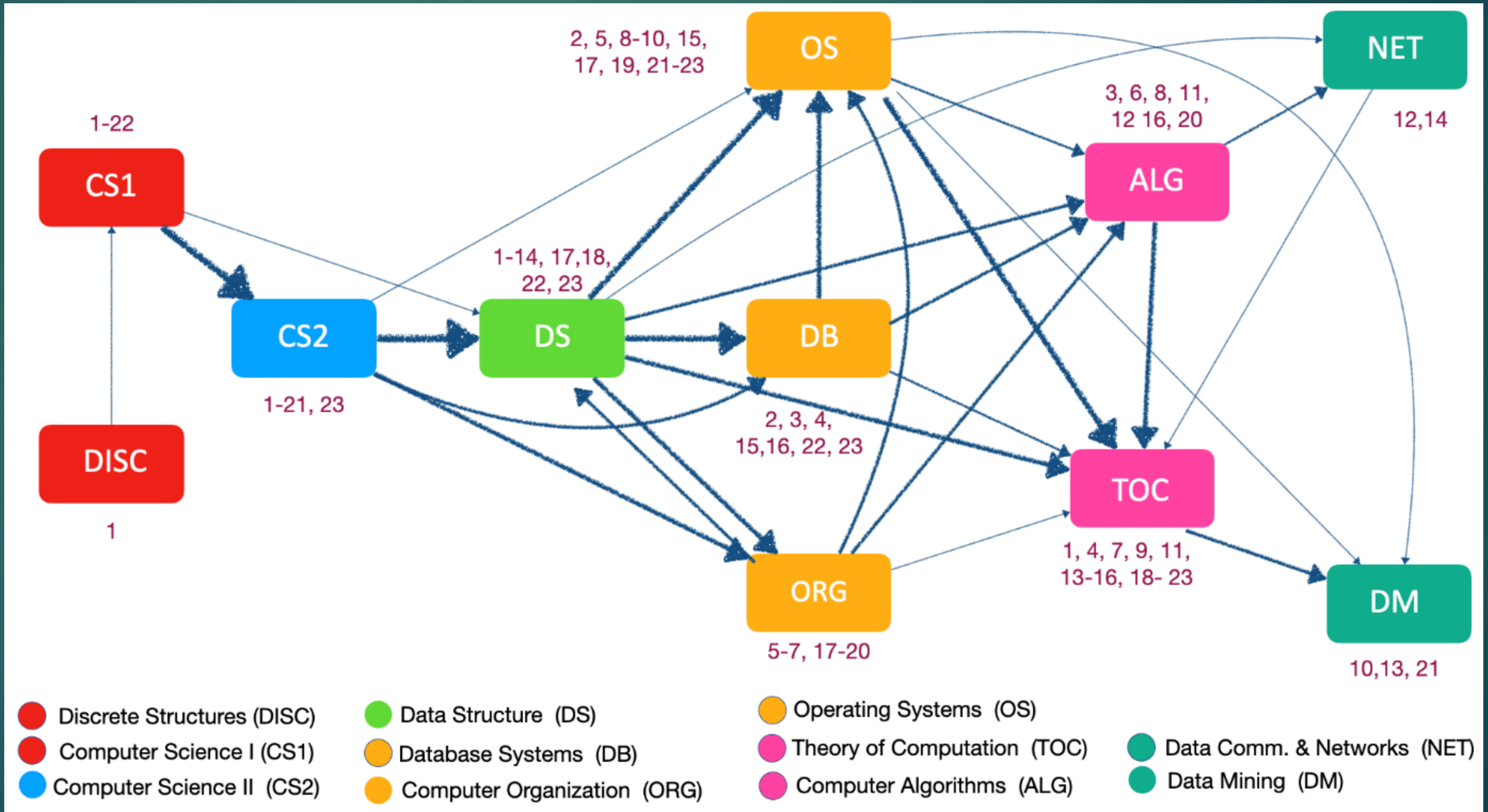
Computer Science Course Sequence Flow
(covers all 5-sequences)

Identifies CS1→CS2 →DS programming sequence

Mostly identifies different course levels (CS1 intro, Algorithms advanced)

Shows Data Mining often taken very late even though no prereqs and could be taken early

Algorithms usually taken before TOC even though similar levels and no prereq relationship. Artifact of scheduling or job prep.



Summary Department Level Results

		Number of k-sequences (k from 2 to 9)							
Department	minsup	2	3	4	5	6	7	8	9
CompSci	50*	56	111	97	21	1			
	100*	61	61	18	1				
Chemistry	50*	9	44	178	152	40	4		
	100*	12	4	1					
Physics	50*	17	14	12	10	5	1		
	100*	10	2						
Biology	50*	22	61	59	23	1			
	100*	14	24	16	5				
Math	50*	40	98	68	6				
	100*	35	32	1					
Psychology	50*	110	193	48					
	100*	72	41	8					
Bio+Chem	50*	38	179	512	592	395	170	38	
	100*	48	115	202	248	174	64	3	
All	500*	322	819	902	599	328	141	40	
	1000*	105	238	148	6				

- Minsup has large impact
- Highly constrained majors have larger sequences
- Adding courses makes large difference
  - See “Bio+Chem”
  - See “All” (high minsup to avoid exponential growth in sequences)

Q3: How does Gap between Courses Impact Performance?

- Methodology
  - We compute, for every pair of courses (A,B), the performance of students taking B after A based on semester gaps between A and B
  - To remove impact of different instructor grading policies we z-normalize grades at section level (Level 1 normalization- L1)
  - To remove impact of differing student abilities in the different partitions, we then z-normalize by student overall GPA (Level 2 normalization- L2)

Gary Weiss, Joseph Denham, and Daniel Leeds. The Impact of Semester Gaps on Student Grades. *Proc. of The 15th Int. Conference on Educational Data Mining (EDM22)*, Durham, UK, July 24-27, 2022.

Computer Science Course Gap Table

CS1→CS2 →Data Struct is key course sequence

We focus on L2 normalization

L2 Diff is difference between performance of gap 1 and 2. Positive: larger gap worse.

Gap 1 means consecutive semesters and gap 2 means extra semester in between

Key result:
An intervening semester between CS1 & CS2 or CS2 & Data Struct leads to worse grades

Course 1	Course 2	L1 Diff	L2 Diff	Corr.	Stdnts	L1 gap 1	L1 gap 2	L2 gap 1	L2 gap 2
CS1	**CS2**	**.250**	**.808**	.568	582	.052	-.198	.078	**-.730**
CS2	**DataStruct**	**.301**	**.357**	.549	351	.072	-.229	.063	**-.293**
Databases	**DataComm**	.044	.346	.526	170	.000	-.044	.452	.107
Databases	**OS**	-.017	.069	.520	187	.146	.163	.188	.118
CS1	**Databases**	.116	.024	.443	274	-.193	-.309	-.087	-.111
CS2	**Databases**	-.029	-.009	.469	238	.069	.098	.304	.313
DiscMath	**CS1**	-.196	-.137	.382	300	-.320	-.125	-.160	-.024
DataStruct	**Algrthms**	-.163	-.346	.526	232	.011	.174	-.131	.215
DataStruct	**OS**	-.168	-.353	.550	226	-.114	.054	-.284	.069
DataStruct	**TOC**	-.143	-.412	.460	251	-.149	-.006	-.402	.010

Spanish Course Gap Table

Spanish 1 → Spanish 2 → Lang & Lit form initial course sequence

Results show that an extra semester between these key courses leads to worse performance.

Course 1	Course 2	L1 Diff	L2 Diff	Corr.	Stdnts	L1 gap 1	L1 gap 2	L2 gap 1	L2 gap 2
Spanish1	**Spanish2**	.686	**.420**	.663	2348	.001	-.685	-.036	-.456
Spanish2	**Lang&Lit**	.471	**.232**	.640	2472	-.023	-.494	-.236	-.468
ApprToLit	**LatinAmerica**	-.111	.151	.592	166	-.092	.019	-.451	-.602
Spanish1	**Lang&Lit**	-.317	-.246	.583	2020	-.395	-.078	-.580	-.334
Lang&Lit	**ApprToLit**	-.113	-.397	.390	496	.009	.122	-.747	-.350

Q4: How Does Order of Courses Impact Performance?

- This study considers, for pairs of courses A and B, whether it is better to take A → B or B → A
  - Examine all course pairs where enough common students take the courses in both orderings
  - Normalize grades at section level to account for different instructor grading policies (e.g., easy graders)
  - Order Benefit (defined below) measures the difference in one order over the reverse order
- New metrics
  - DNG = Difference in normalized grades
    - $DNG_{A:B} = \mu_A(B \to A) - \mu_A(A \to B)$ // Advantage of taking Course A second
    - $DNG_{B:A} = \mu_B(A \to B) - \mu_B(B \to A)$ // Advantage of taking Course B second
    - Both DNGs often positive since usually do best if a course taken second
  - OB = Order Benefit (positive OB preferred order)
    - $OB_{A\to B} = DNG_{B:A} - DNG_{A:B}$ **(Note: $\boldsymbol{OB_{A\to B}}$ = -$\boldsymbol{OB_{B\to A}}$)**
    - Order Benefit measures relative performance of taking B versus A second
    - If $OB_{A\to B}$ > 0 then best to take A → B (if < 0 then take B → A

Tess Gutenbrunner, Daniel Leeds, Spencer Ross, Michael Riad-Zaky, and Gary Weiss, Measuring the Academic Impact of Course Sequencing using Student Grade Data. *Proc. of 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 799-803.

CompSci Courses with highest OB

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Computer Alg.	Data Mining	-0.110	0.233	0.343
Data Structures	Computer Organization	-0.073	0.103	0.176
Data Mining	Data Comm. & Netwks.	0.101	0.235	0.134

Math Courses with highest OB

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Discrete Math	Multivar. Calc II	-0.056	0.252	0.308
Multivar. Calc. I	Discrete Math	-0.041	0.249	0.290
Business Finite Math	Finite Math	-0.024	0.145	0.169

CompSci/Math Courses with highest OB

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Structures of CS	Finite Math	-0.002	0.429	0.431
Calculus I	CS I	-0.035	0.338	0.373
Calculus I	CS I Lab	-0.012	0.252	0.264
Calculus I	Structures of CS	-0.010	0.213	0.223

Some of these results are easily justified

- We advise students to take "Data Structures" before "Computer Org," so supports this advice.
- "Business Finite Math" and "Finite Math" cover similar material
  - Assume business version simpler, so we expect best outcome if simpler taken first.
- Nice to see that taking calculus before CS1 programming course is beneficial.
  - Supports decision to require calculus, although hard to explain these results.
  - Calculus not that important to programming especially compared to discrete math.

Q5: How Effective is an Instructor?

- Motivation
  - Critical for tenure and promotion decisions, deciding who should teach what courses, and who needs more training.
  - Current methods rely on student surveys and peer evaluations, both of which are highly subjective and may suffer from gender and racial bias
  - Our goal is to assess effectiveness only based on *future performance*
  - Have observed that at least one "CS 2" instructor said that student grades were largely dependent on the prior instructor for "CS 1"

Gary Weiss, Erik Brown, Michael Riad-Zaky, Ruby Iannone, and Daniel Leeds. Assessing Instructor Effectiveness Based on Future Student Performance. *Proc. of 15th International Conference on Educational Data Mining (EDM22)*, Durham, UK, July 24-27, 2022.

Methodology

- Instructor effectiveness measured between pairs of courses (although can be subsequently aggregated)
- For first course use all sections taught by instructor being evaluated and consider all sections of second course
  - For example, if I teach CS1, then measure performance of my CS1 students when they take CS2 (with any instructor)
- Normalize grades to account for instructor grading (Level 1) and student ability as measured by GPA (Level 2)
- Compute mean instructor benefit between course pairs

Instructor Effectiveness for CS2 (based on Grades in Data Structures)

Instructor ID	Sections Taught	Total # Students		Instructor Benefit	
		CS2	DataStruct	Level 1	Level 2
F212	12	293	158	-0.226	-0.189
F177	4	92	62	0.151	**0.396**
F589	3	56	36	-0.177	-0.228
F653	3	35	33	-0.042	**0.400**
All	32	697	410	-0.145	-0.054

- Data Structures follows CS2 in programming sequence
- Results from Table:
  - Effective instructors: F177 & F653
  - Less effective: F212 & F589
  - T-test between F177 and F212
    - p-value of 0.0003
- Other Results:
  - 4 of top-10 instructors from STEM
  - 8 of bottom-10 from STEM

Q6: What Major will a Student Perform Well In? Choose?

- Motivation
  - Selection of academic major extremely important yet often done with little guidance
  - Poor choice can lead to academic failure or delay in graduation (major change)
- Our approach
  - View this as a recommendation problem and use collaborative filtering
  - Measure similarity based on grades in core courses over first two years
  - We use "average grade in major" in place of product or movie rating
    - Recommend majors student will perform well in (but may not be best for them)
    - Also evaluate how likely the major they choose is in the top-5 recommendations

Samuel Stein, Gary Weiss, Yiwen Chen, and Daniel Leeds. A College Major Recommendation System. *Proc. of the Fourteenth ACM Conference on Recommender Systems (RECSYS 20*), 640-644, September 2020.

Methodology

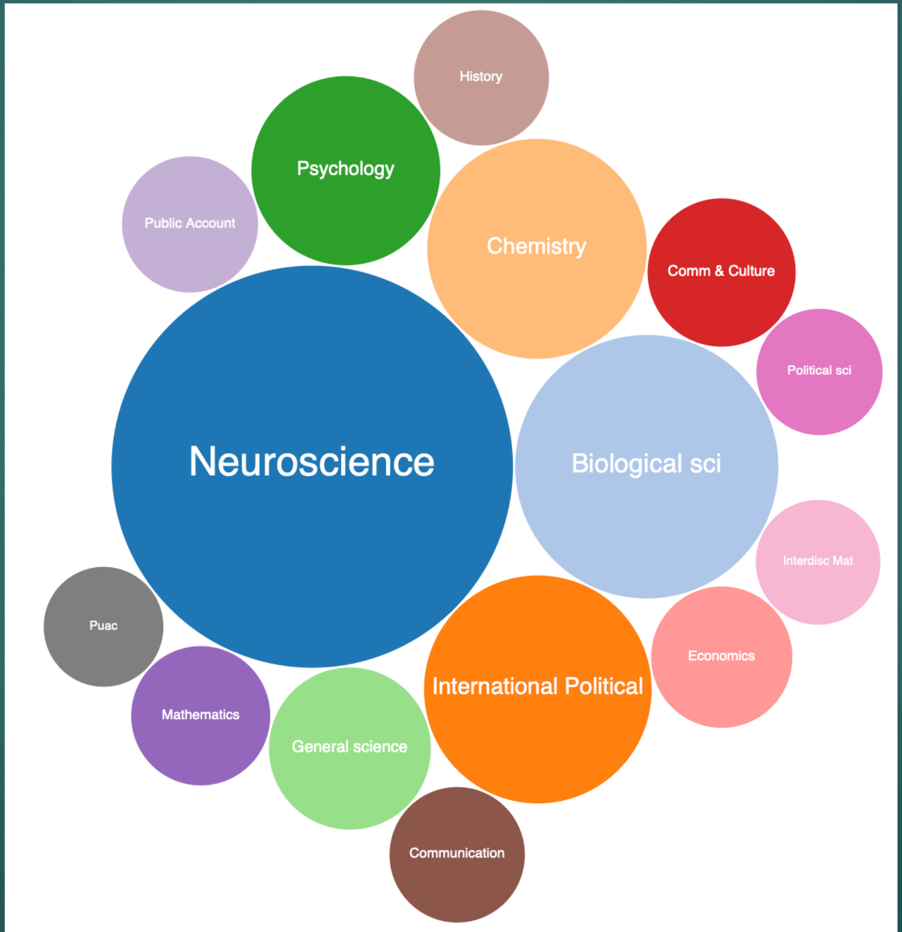
- Use nearest neighbor algorithm to find similar students
  - Use cosine distance as similarity metric
  - Compute similarity based on grades in core courses over first two years
    - Must recommend majors early on so that is why limit time
    - Only use core courses since if students decide on major early, could take courses from major and that would make recommending easy
- Evaluation Metric Components
  - Recommended: % of cases where actual major in the top-5 of recommended majors
  - nGPA > 0: If satisfied then student outperforms average major GPA
    - Accounts for fact that grades vary heavily by department and major

Recommendation Results

- 4 Recommendation Strategies
  - Randomly pick major
  - Pick most common major
  - Pick student's actual major
  - Use recommender system
- Care most about what **is** recommended
  - QOR: recommender system does best
  - Outperforms actual major!
    - Only possible if students perform better in recommended versus actual major
- When major is not recommended by system, students perform worse, which is encouraging

Evaluation Metric	Major Recommendation Strategy			
	Random	Most Common	Actual Major	Recommender System
Recommended & nGPA $\geq$ 0 (QOR)	55%	55%	55%	67%
Recommended & nGPA < 0	45%	45%	45%	33%
Not Recommended & nGPA $\geq$ 0 (QONR)	55%	55%	55%	44%
Not Recommended & nGPA < 0	45%	45%	45%	56%

Majors often Recommended with Neuroscience



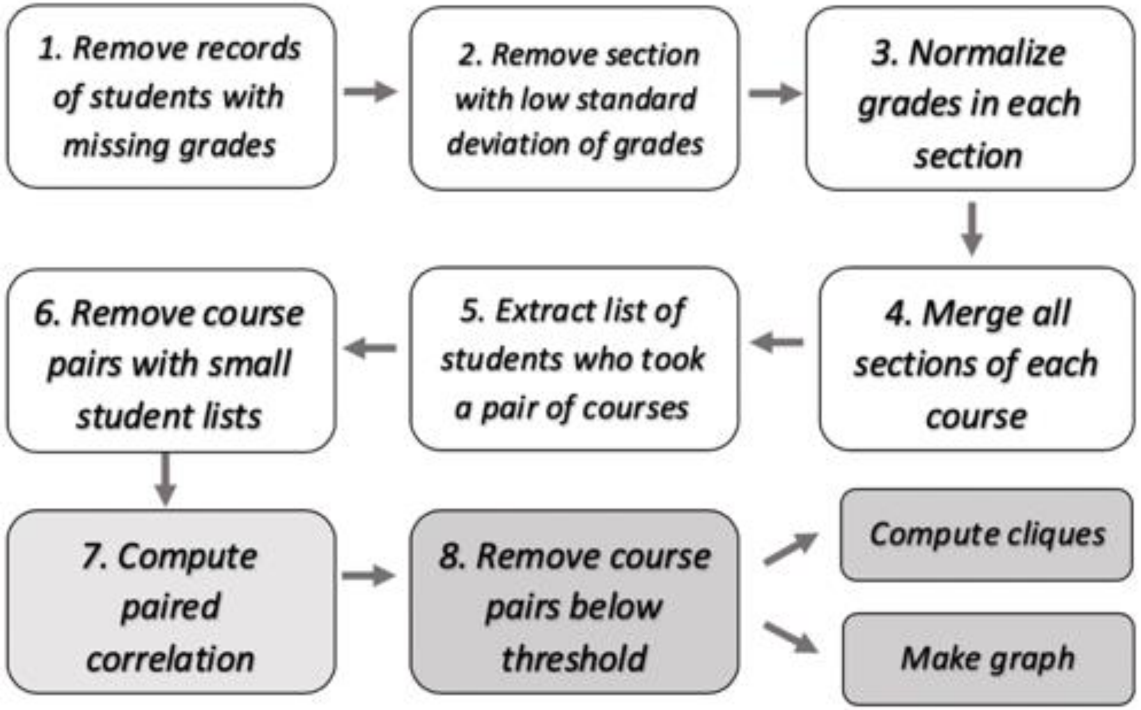
Q7: How Can we Group Courses Based on Similar Student Performance?

- Measuring similarity
  - Two courses are considered similar if the correlation of grades of students taking both courses is high (above threshold)
  - There may be a causal link (doing well in the one impacts the other) or no causal link (the courses may require similar skills)
  - This is a relatively unusual/innovative way of measuring course similarity
- Related useful questions:
  - Are courses within a major more similar than courses in different majors?
  - Can pairwise similarities be used to form meaningful course clusters?
  - Are there high-level patterns that exist/differ between course groupings?

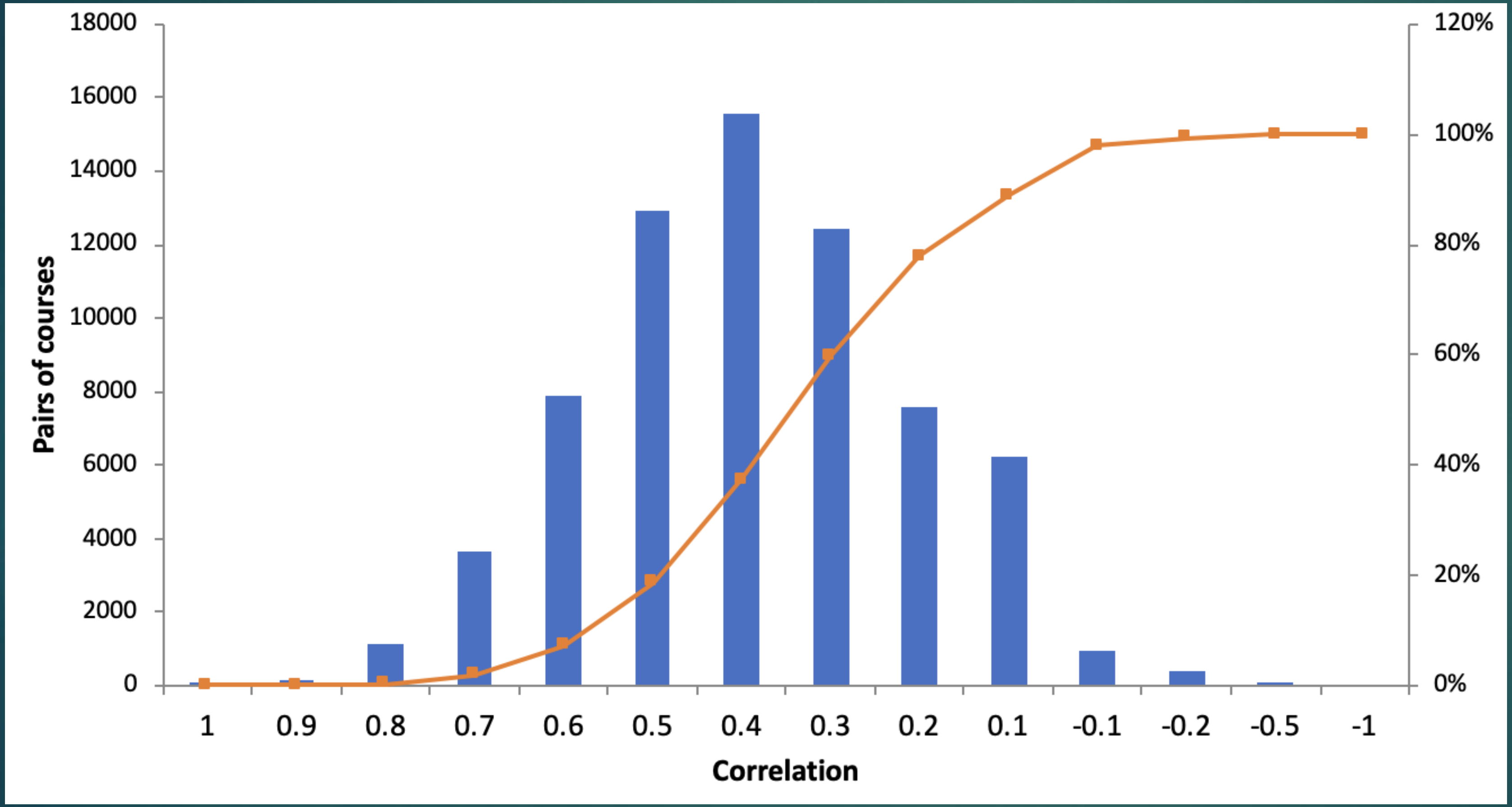
Daniel Leeds, Tianyi Zhang and Gary Weiss, Mining Course Groupings using Academic Performance. *Proc. of 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 804-808.

Methodology

- Normalize grade at section level to account for different instructor grading policies (easy vs. hard).
- Generate course pair dataset (e.g., CS1, CS2), with grades of common students in same position.
- Compute correlation between grade vectors pairs
- Form graph with courses as node
  - Edge between courses if grade correlation above threshold.
- Analyze graph



Distribution of Course-Pair Correlations



- Graph has edge if correlation ≥ 0.5
  - covers 20% of courses
- Strong (weak) students may perform similarly in different classes
  - but still much variation

Course Pairs with High Correlations

Course 1	Course 2	Correlation
Comp Sci 2	Comp Sci 2 Lab	0.95
Gen Phys 1	Computational Neuro.	0.81
Intro Bio 1	Intro Bio 1 Lab	0.79
Web Programming	Bioinformatics	0.79
Learning	Health Psychology	0.78
Gen Chem 2 Lab	Computer Algorithms	0.78
Philos. of Human Nature	Infant & Child Develop.	0.78
Law & Psychology	Cllinical Child Psych.	0.77

- Lectures and their labs heavily correlated
  - This is expected
- Connection between General Physics and Comp. Neuro. interesting
- Chem Lab & Algorithms correlated
  - Both involve following "recipe"?

Network/Graph Analysis & Metrics

- Modularity:
  - Groups of nodes are densely connected to each other
    - High modularity score means more edges than expected by chance
- Betweenness Centrality:
  - How often node appears on shortest paths between random pair of nodes
- Next three figures automatically generated by Gephi (www.gephi.org)
  - Color indicates modularity class
  - Size represents betweenness centrality

Course Network for Computer Science

Purple: Mainly Programming Courses

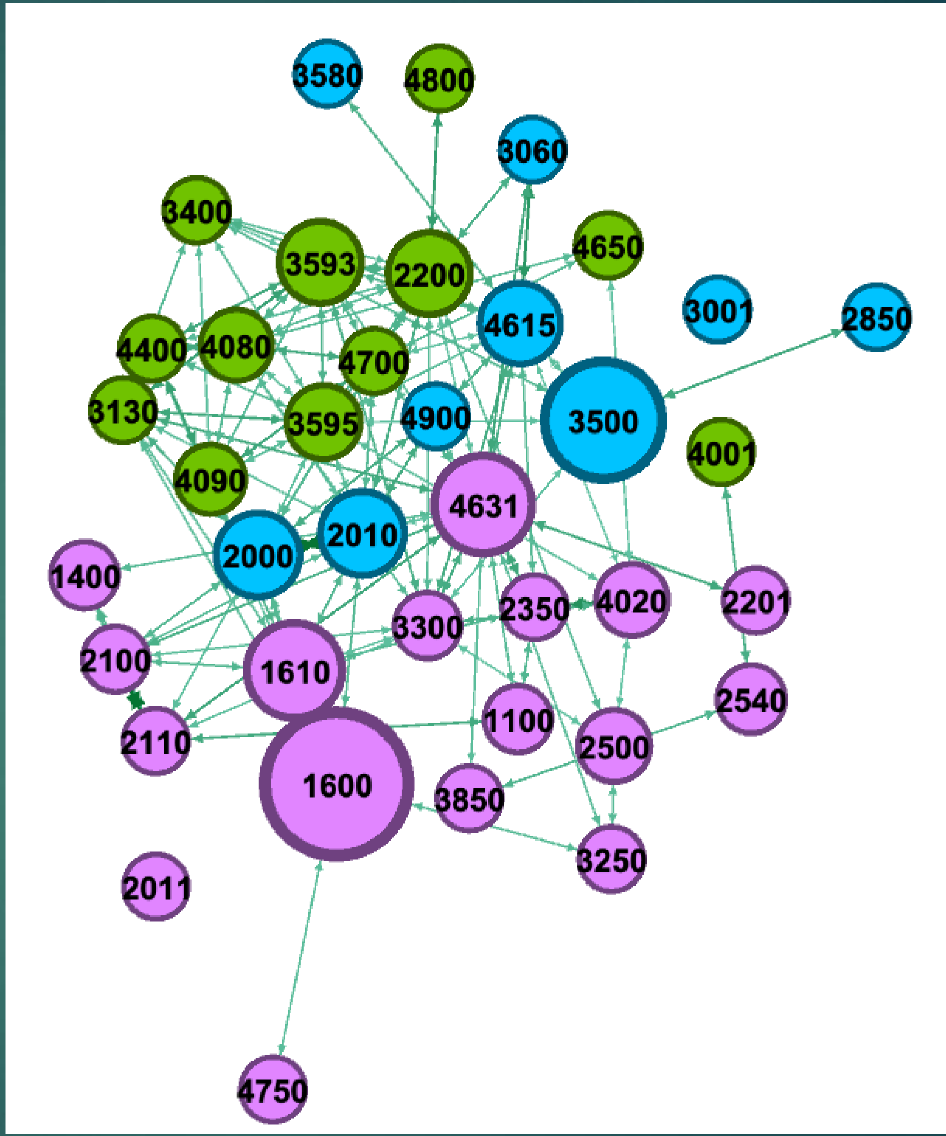
- Data Mining (4631)
- Web Programming (2350)
- Informatics (2500)
- CS I with Lab (1600, 1610)
- CS II with lab (2000, 2010)
- Scientific Computing (4750)

Green: Advanced CS Courses

- Data Structures (2200)
- Algorithms (4080)
- Theory of Computation(4090)
- Operating Systems (3593)
- UNIX programming (3130)

Blue: Info Science& Miscellaneous

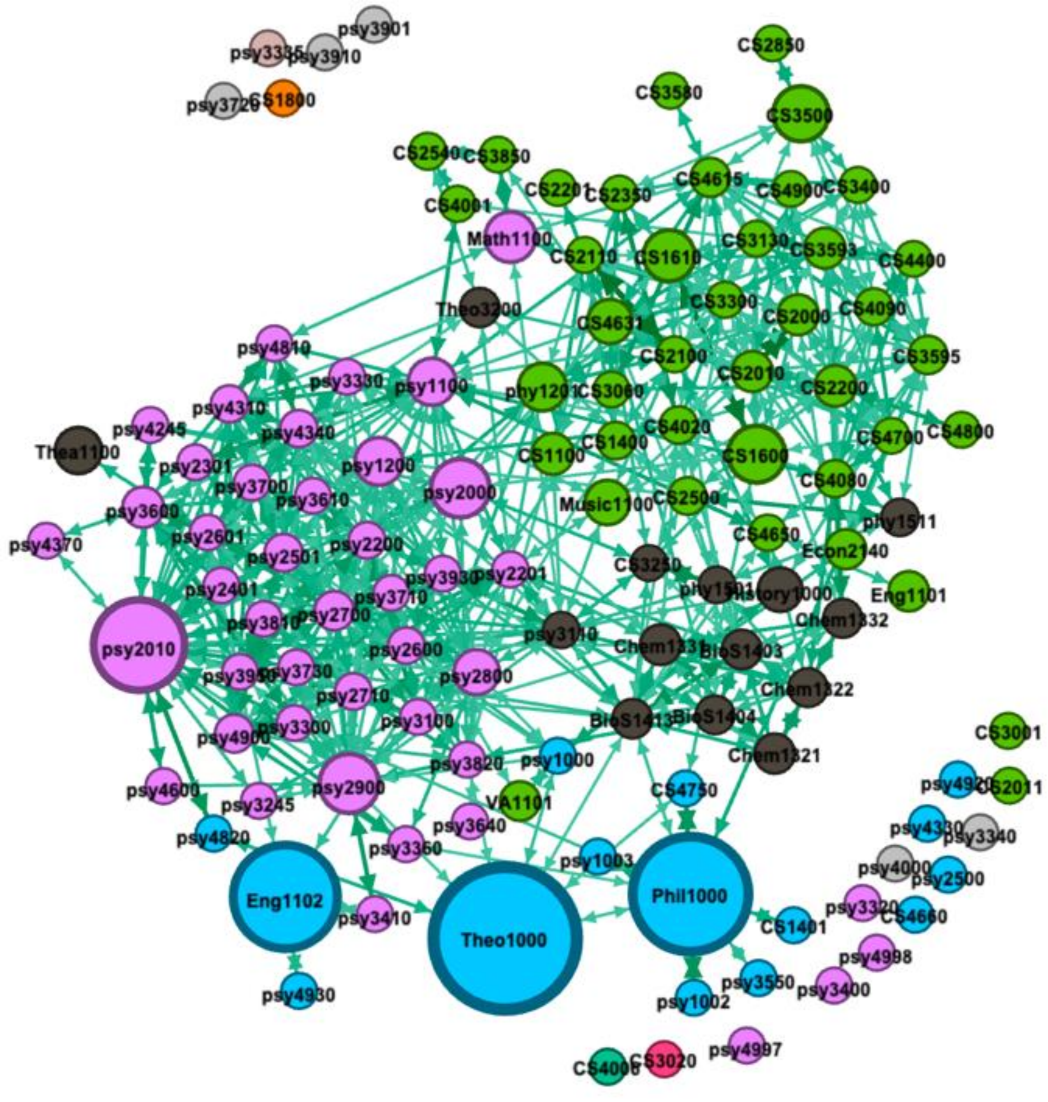
- Computer Data Analysis)2850)
- Database Systems (3500)
- Advanced Database Systems (4515)
- Robots and Film (3001)



- Color indicates modularity class
- Size represents betweenness centrality

Course Network for 3 Departments + Core Curriculum Courses

- Modularity classes:
  - Green (right): Computer Science
  - Purple (left): Psychology
  - Dark grey (under green): Pre-health courses
- Courses within departments better connected than between departments
- First-year core curriculum courses large and hence have high betweenness-centrality
  - English 1102
  - Theology 1000
  - Philosophy 1000



- Color indicates modularity class
- Size represents betweenness centrality

Courses Comprising the Largest Cliques

- Clique:
  - *Fully connected set of nodes (edge between all nodes)*
  - *N-clique: clique with n nodes*
- *Courses within a clique usually follow a common theme*

BIOLOGICAL SCIENCES		
	8-Clique	
Introductory Bio I	General Genetics	Human Anatomy
Introductory Bio II	General Genetics Lab	Human Physiology
Introductory Bio Lab II	Ecology	
CHEMISTRY		
	8-Clique	
General Chem II	Organic Chem Lab II	Biochem Lab I
General Chem Lab II	Physical Chem I	Inorganic Chem
Organic Chem I	Organic Chem II	
COMPUTER SCIENCE		
5-Clique	**5-Clique**	**5-Clique**
Data Mining	Data Mining	CS II
Web Programming	Web Programming	CS II Lab
Data Structures	Data Comm.	Data Structure
Client Server	Client Server	Operating Systems
Computer Org.	Computer Org.	Scientific Comput
PSYCHOLOGY		
	8-Clique	
Child Development	Biopsychology	Research Methods Lab
Learning	Social Psych Lab	Human Sexuality
Aging and Society	Law and Psych	

Category Level Summary Clique Info

Category	courses	sections	records	aveCorr	largest Clique	#clique>5	BetweenC
Sciences and Mathematics	43	670.5	14543.5	0.464	7.5	32.5	0.120
Humanities	71.5	549.5	9103.5	0.410	3	0	0.336
Arts	87	548	8926	0.364	4	0	0.294
Communication and Media Studies	31	91	1681	0.483	2	0	0.124
Social Science	67	665	13903	0.498	5	1	0.148
Modern Languages	15.5	116.5	1426	0.612	4	0	0.244
Others	19.5	258	2916	0.425	4.5	0.5	0.144

- Science & Math have the most large cliques and lowest betweenness centrality
  - Makes sense if most courses are highly related/connected
- Modern Languages has the highest average correlation
  - Courses highly connected/related, as expected (fewer courses limits cliques)

Q8: What Can we Learn from Course Co-enrollment Patterns?

- Use graph theory to learn about courses frequently taken together
  - *Differs from last Q7 since that focused on grade similarity*
  - Identify hub courses that are well connected to other courses
  - Compute network metrics to provide insight into course groupings/subnetworks
    - A traditional task often applied to social networks
- Acquired knowledge can aid in:
  - course planning
  - understanding course structure in different academic departments
  - as-yet-unknown benefits

Gary Weiss, Nam Nguyen, Karla Dominguez and Daniel Leeds. Identifying Hubs in Undergraduate Course Networks Based on Scaled Co-Enrollments. *Proc. of The 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 809-813.

Formation of Network Graph

- Network graph again formed from the course-pair data
  - Nodes represent courses and edges connect nodes if number common students above threshold
- Two types of thresholds
  - <u>Static</u>: edge included if at least 20 common students
  - <u>Dynamic</u>: edge included if co-enrollment *proportion* exceeds threshold and at least 20 common students
  - Dynamic threshold removes many edges associated with popular core curriculum courses.

Network Analysis Metrics

	Metric	Summary Description	Range
Network or Subnetwork Level	Density	Fraction of possible edges present	0 - 1
	Diameter	Maximum distance between any pair of nodes in network	$\mathbf{Z^+}$
	Ave. Clustering Coefficient (ACC)	Fraction of pairs of neighbor nodes connected to each other	0 - 1
Node Level Centrality Metrics	Degree Centrality	Number of edges to node (degree)	$\mathbf{Z^+}$
	Eigenvector centrality	Based on centrality of node's neighbors	≥ 0
	Betweenness centrality	Measure all shortest paths passing through a node	≥ 0

Course Network Results

- Two departments displayed per category
- Network covering "All" courses has much lower density
- Dynamic threshold vs. Static
  - decreases edges, density, and cluster coefficient (AC)
- STEM departments have very high density and ACC
  - Most likely because the knowledge in courses is dependent on other courses
    - Lots of prerequisites

Category/ Department	Nodes	Static Threshold				Dynamic Threshold			
		Edges	Density	Diam.	ACC	Edges	Density	Diam.	ACC
ALL	1763	39968	0.03	4	0.74	24323	0.02	6	0.40
Arts	**41.5**	**239**	**0.32**	**3**	**0.56**	**231**	**0.29**	**2.5**	**0.56**
Dance	54	1236	0.86	3	0.95	1236	0.86	3	0.95
Music	24	87	0.32	3	0.51	73	0.26	2	0.52
Comm and Media Studies	**24**	**25**	**0.20**	**2**	**0.16**	**25**	**0.19**	**2**	**0.16**
Comm and Media Studies	94	862	0.20	3	0.72	828	0.19	4	0.58
New Media & Digital Design	6	8	0.53	2	0.00	8	0.53	2	0.00
Humanities	**81**	**179**	**0.06**	**3**	**0.21**	**104**	**0.04**	**3**	**0.08**
African & African Amer Studies	28	34	0.09	2	0.11	34	0.09	2	0.11
English	167	462	0.03	3	0.59	258	0.02	3	0.12
Modern Languages	**9**	**19**	**0.53**	**2**	**0.49**	**19**	**0.53**	**2**	**0.38**
Greek	4	6	1.00	2	0.00	6	1.00	2	0.00
Spanish	40	118	0.15	3	0.49	98	0.13	2	0.33
STEM	**34**	**295**	**0.47**	**3**	**0.76**	**288**	**0.45**	**3**	**0.75**
Biological Sciences	30	274	0.63	2	0.77	274	0.63	2	0.77
Physics	38	286	0.41	4	0.73	277	0.39	3	0.74
Social Science	**74**	**329**	**0.18**	**2.5**	**0.51**	**285**	**0.16**	**2.5**	**0.44**
Economics	45	325	0.33	2	0.64	270	0.27	2	0.59
Sociology	90	236	0.06	3	0.37	206	0.05	3	0.30

Hub Analysis

- Top static and dynamic hubs based on combined rank that is median of 3 centrality metrics
  - Similar values for static but not dynamic threshold
- Core courses dominate static hubs
  - Not surprising that popular courses taken by most students are always hubs
  - Lower ranked static hubs that are not core courses may be more interesting
- Dynamic hubs may be interesting
  - Biopsychology *frequently* taken with many other courses
    - Makes sense since it is of interest to several communities

Courses	Combined Rank		Centrality Rank		
	Static	Dyn.	Deg.	Btw.	Eig.
Static Threshold: Top Hubs					
Philosophical Ethics	1	45	1	1	2
Faith & Critical Reason	2	76	2	2	1
Philos. of Human Nature	3	75	3	3	3
Composition II	4	78	4	5	4
Banned Books	5	49	5	4	5
Finite Mathematics	7	56	6	7	7
Spanish Lang and Lit	7	29	7	6	8
Dynamic Threshold: Top Hubs					
Biopsychology	31	3	3	20	3
Phys. Sci.: Today's World	30	4	4	2	63
Latin American History	44	5	2	6	5
Intro World Art History	22	5	5	26	2
Intro Phys. Anthropol.	41	6	1	9	6
Intro Cultural Anthropol.	18	6	6	33	1
Films of Moral Struggle	55	8	8	7	54


WRAP IT UP

Limitations and Future Work

- Limitations
  - Little validation, but difficult for descriptive data analysis/data mining
    - Could replicate statistics across subsets of data (like train and test sets)
  - Maybe link to other info sources
    - E.g., validate instructor effectiveness via student surveys
  - We often find that our dataset is not as large as we would like
    - Especially at the department, instructor (small class size), course level
- Future Work
  - Replicate grading study on other universities' data (we have the data)
  - Working to make our grading and sequence analysis tools easy to use
    - Code shareable now and packaged with documentation/tutorials within few weeks
    - https://www.cis.fordham.edu/edmlab/software/

Other Educational Research

- Working with Dr. Yijun Zhao on studying MS in Computer Science and MS in Data Science admission application data
  - Predicting admissions decisions using predictive modeling including textual data (letters of recommendation and resumes)
  - Analyzing letters of recommendation for gender bias and cultural bias based on country of origin
  - If interested, we expect most of this work to be published within the next few weeks

Fordham EDM Lab References

- References numbered by the questions covered in this talk (listed on slide 3)
  - https://www.cis.fordham.edu/edmlab/publications

1. Gary M. Weiss, Luisa A. L. Rosa, Hyun Jeong and Daniel D. Leeds (2023). An Analysis of Grading Patterns in Undergraduate University Courses. *Proceedings of the 2023 IEEE 467h Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, Torino, Italy.
2. Daniel Leeds, Cody Chen, Yijun Zhao, Fiza Metla, James Guest, and Gary Weiss. Generalized Sequential Pattern Mining of Undergraduate Courses**.** *Proc. of 15th Int. Conference on Educational Data Mining (EDM22*), Durham, UK, July 24-27, 2022.
3. Gary Weiss, Joseph Denham, and Daniel Leeds. The Impact of Semester Gaps on Student Grades. *Proc. of The 15th Int. Conference on Educational Data Mining (EDM22)*, Durham, UK, July 24-27, 2022.
4. Tess Gutenbrunner, Daniel Leeds, Spencer Ross, Michael Riad-Zaky, and Gary Weiss, Measuring the Academic Impact of Course Sequencing using Student Grade Data. *Proc. of 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 799-803.
5. Gary Weiss, Erik Brown, Michael Riad-Zaky, Ruby Iannone, and Daniel Leeds. Assessing Instructor Effectiveness Based on Future Student Performance. *Proc. of 15th International Conference on Educational Data Mining (EDM22*), Durham, UK, July 24-27, 2022.
6. Samuel Stein, Gary Weiss, Yiwen Chen, and Daniel Leeds. A College Major Recommendation System. *Proc. of the Fourteenth ACM Conference on Recommender Systems (RECSYS 20*), 640-644, September 2020.
7. Daniel Leeds, Tianyi Zhang and Gary Weiss, Mining Course Groupings using Academic Performance. *Proc. of 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 804-808.
8. Gary Weiss, Nam Nguyen, Karla Dominguez and Daniel Leeds. Identifying Hubs in Undergraduate Course Networks Based on Scaled Co-Enrollments. *Proc. of The 14th Int. Conference on Educational Data Mining (EDM21)*, Paris France, June 29-July 2, 2021, 809-813.

Acknowledgements

- All of this research was done collaboratively with Dr. Daniel Leeds and some with Dr. Yijun Zhao
- Many graduate and undergraduate students were involved in this research. The following appear as co-authors on the papers:
  - Erik Brown, Cody Chen, Yiwen Chen, Joseph Denham, Karla Dominguez, James Guest, Ruby Iannone, Hyun Jeong, Fiza Metla, Nam Nguyen, Michael Riad-Zaky, Luisa Rosa, Spencer Ross, Samuel Stein , Tianyi Zhang

Questions?

For more information:

- You can contact me at: gaweiss@fordham.edu
- My webpage: https://storm.cis.fordham.edu/~gweiss/
- EDM Lab page: https://www.cis.fordham.edu/edmlab/
- These Slides: https://storm.cis.fordham.edu/~gweiss/presentations/Weiss-ICDATA23-invited-talk.pptx (or .pdf)