Making Personalized Education Scalable
The Challenge: ‘Scalable Personalized Education’
Ideal Education is 1-on-1

Thorough Evaluation

Personalized Feedback
What is a MUST for Scalable Personalized Education?

- **My perspective** = Personalization + Education
  - **Education**: Lecturer for 15 years in Computer Science, Director of a data mining study program
  - **Personalization**: Founder at Correlor – Social Big Data, main investor: JVP, clients: big telcos

- So what is the “BIG DATA” in education? **Assignments**

- In STEM (Science, Technology, Engineering, Math) assignments are a big and integral part of learning

- The big majority is ‘**Open-Ended**’, which means...

- **...Evaluation & Personalization are Not Scalable**
Evaluation of Open-Ended Assignments

- Currently: **Manual** and **Specialized**

- **OECD:** Takes >20% of educators’ time

- Evaluation of a single assignment: >5 min

- **Serious** evaluation by a full-time educator: 1,328 hours / year = 166 working days / year

- Realistically, in big courses:
  - Assignments are not being evaluated thoroughly
  - Detailed feedback **cannot** be given on individual basis
MOOCs (Massive Open Online Courses)
- 400+ universities, 2500 courses
- 18M students (11.7M in STEM)
- Millions of dollars in revenues
- 7,200 students in a course
- 90% dropouts

“Evaluation and feedback in MOOCs are virtually impossible”

“It is The Barrier to the MOOC revolution”

Prof. Gil Weinberg, Georgia Tech
1st accredited MOOC for M.Sc. in CS

“Multiple-Choice Questions and Auto-Grading are not the answer”

Times 2014 – Higher Education
How Can We Solve the Problem?

Average Number of Different Solutions

Number of Students

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8
How Can We Solve the Problem?

<table>
<thead>
<tr>
<th>Number of Students</th>
<th>Average Number of Different Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

The Solution

( ) Sense
<table>
<thead>
<tr>
<th>Number of Students</th>
<th>Average Number of Different Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

**How Can We Solve the Problem?**

The graph shows the relationship between the number of students and the average number of different solutions. The data increases linearly as the number of students increases.
How Can We Solve the Problem?

The Solution

<table>
<thead>
<tr>
<th>Number of Students</th>
<th>Average Number of Different Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
How Can We Solve the Problem?

How can machines detect these “solution types” the same way humans can?
Personalized Medicine based on DNA Analysis
Harnessing DNA Analysis Technologies

Submitted Assignments
- Code Analysis
- Text Mining

Pseudo-DNA Sequences
- Genomics

Similarity Patterns
- Clustering

Human Intelligence
- Memory Leak = 80
- Inefficient Algorithm = 80
- Perfect = 100
- Writing is Not Elegant = 95
- Solution is Not Robust = 60

Evaluation & Feedback

Educator

Explanations

Solution Types
Similarity Patterns as Building Blocks

1. A B C D E F G H I J K L M
2. G H I J A B C D E F G H I J K L M
3. K L M X D E F 1 2 3 4 A B C D E F G H I J K L M
4.

Genomics

A B C D E F G H I J K L M
Similarity Patterns as Building Blocks

1. “Wild type” / Centroid
2. Different order
3. Mutations: Insertion, Deletion, Switch
4. Outliers detection

Genomics
```cpp
#include <iostream>
#include <math.h>
using namespace std;

double squared_distance(double m, double x1, double y1)
{
    double distance;
    distance = pow((y1-m*x1), 2);
    return distance;
}

double find_slope(double x1, double y1, double x2, double y2)
{
    double d1, d2, d3, average;
    double m = 0;
    d1 = squared_distance(0, x1, y1);
    d2 = squared_distance(0, x2, y2);
    d3 = squared_distance(0, x3, y3);
    average = (d1+d2+d3) / 3;

    for (slop = 0.1; slop <= 5; slop += 0.1)
    {
        double distance1, distance2, distance3;
        distance1 = squared_distance(slope, x1, y1);
        distance2 = squared_distance(slope, x2, y2);
        distance3 = squared_distance(slope, x3, y3);
        double average_lope = (distance1 + distance2 + distance3) / 3;

        if (average_lope < average)
        {
            average = average_lope;
            m = slope;
        }
    }
    return m;
}
```
```
Tailored feedback for: Wrong Hierarchy

- **Group Name**: Wrong Hierarchy
- **Feedback**: The design of function hierarchy is not optimal. Several pieces of code are written more twice or more, which may result inconsistency and bugs.
- **Grade**: 80
- **Next Assignment**: Recursion-C
- **Recommended Content**: https://www.youtube.com/watch?v=pTbJ6fUMw
Tailored feedback for: Low Robustness

- **Group Name**: Low Robustness
- **Feedback**: Your algorithm produces the correct result in more than 99% of the times. HOWEVER, there are some extreme cases of particular inputs, which will result your algorithm to get it wrong. Can you tell when and why?
- **Grade**: 90
- **Next Assignment**: Recursion-B
- **Recommended Content**: https://www.youtube.com/watch?v=pTbj6fUMw
<table>
<thead>
<tr>
<th>Student name</th>
<th>Student ID</th>
<th>Grade</th>
<th>Feedback</th>
<th>Next Assignment</th>
<th>Recommended Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam Altman</td>
<td>2392</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
<tr>
<td>Denise Reiss</td>
<td>9045</td>
<td>75</td>
<td>Well done!</td>
<td>Recursion-1</td>
<td><a href="https://www.youtube.com/watch?v=JaoPjWvYHls">https://www.youtube.com/watch?v=JaoPjWvYHls</a></td>
</tr>
<tr>
<td>Jim Breyer</td>
<td>2105</td>
<td>80</td>
<td>Your main algorithm used two nested loops, one that conducts a regular sort and another that alters it according to prime numbers. However, it is possible to combine the two operations in a SINGLE nested loop.</td>
<td>Recursion-3</td>
<td><a href="https://www.youtube.com/watch?v=35JG05EcYmA">https://www.youtube.com/watch?v=35JG05EcYmA</a></td>
</tr>
<tr>
<td>Charles Marsala</td>
<td>5172</td>
<td>100</td>
<td>Well done!</td>
<td>Recursion-1</td>
<td><a href="https://www.youtube.com/watch?v=JaoPjWvYHls">https://www.youtube.com/watch?v=JaoPjWvYHls</a></td>
</tr>
<tr>
<td>Chris Kelly</td>
<td>5332</td>
<td>80</td>
<td>Your main algorithm used two nested loops, one that conducts a regular sort and another that alters it according to prime numbers. However, it is possible to combine the two operations in a SINGLE nested loop.</td>
<td>Recursion-3</td>
<td><a href="https://www.youtube.com/watch?v=35JG05EcYmA">https://www.youtube.com/watch?v=35JG05EcYmA</a></td>
</tr>
<tr>
<td>Bill Rechert</td>
<td>1570</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
<tr>
<td>Stuart Gannes</td>
<td>8456</td>
<td>80</td>
<td>Your main algorithm used two nested loops, one that conducts a regular sort and another that alters it according to prime numbers. However, it is possible to combine the two operations in a SINGLE nested loop.</td>
<td>Recursion-3</td>
<td><a href="https://www.youtube.com/watch?v=35JG05EcYmA">https://www.youtube.com/watch?v=35JG05EcYmA</a></td>
</tr>
<tr>
<td>Joshua Reeves</td>
<td>5566</td>
<td>100</td>
<td>Well done!</td>
<td>Recursion-1</td>
<td><a href="https://www.youtube.com/watch?v=JaoPjWvYHls">https://www.youtube.com/watch?v=JaoPjWvYHls</a></td>
</tr>
<tr>
<td>Ed Baker</td>
<td>4387</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
<tr>
<td>Denise Aptekar</td>
<td>5491</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
<tr>
<td>Kiran Vaka</td>
<td>9449</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
<tr>
<td>Drue Katsoka</td>
<td>8982</td>
<td>65</td>
<td>Your algorithm produces the correct result in more than 99% of the times. HOWEVER, there are some extreme cases of particular inputs, which will result your algorithm to get it wrong. Can you tell when and why?</td>
<td>Recursion-2</td>
<td><a href="https://www.youtube.com/watch?v=JaoPjWvYHls">https://www.youtube.com/watch?v=JaoPjWvYHls</a></td>
</tr>
<tr>
<td>Takeki Mori</td>
<td>1912</td>
<td>75</td>
<td>In cases where memory allocation is dynamic vs. static, it is critical that the number of ‘new’ commands will be equal to the number of ‘delete’ commands. Not necessarily equal in writing, but equal in calling!</td>
<td>Recursion-4</td>
<td><a href="https://www.youtube.com/watch?v=pTb9fUMw50">https://www.youtube.com/watch?v=pTb9fUMw50</a></td>
</tr>
</tbody>
</table>
Suddenly, So Much is Possible with... Groups

Analytics for Faster & Easier Evaluation

More Detailed and Targeted Feedback

Re-Submission with Re-Feedback

Recommended Next-Best-Assignment

Recommended Complementary Partners

\[\text{( ) Sense Platform}\]
Implementation in Israeli Academia

- Achieved successful academic results on a beta product at Bar-Ilan University, the largest university in Israel
  
- 2 programming courses of 150 students (total)
  
- 1 data mining course of 40 students
  
- 1 medicine course (pilot in free text) of 50 students
  
- 3 professors
  
- 5 teaching assistants

More time to focus on (video) feedback
Implementation in Israeli Academia

The results led the Rector’s academic director to ask for product implementation on a full institutional level.

Average of ~5% increase in 5 competence KPIs

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Grade in former years</th>
<th>Grade this year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 2035</td>
<td>95 Baseline Grade</td>
<td>Grade this year:</td>
</tr>
<tr>
<td>Assignment 2042</td>
<td>92 Baseline Grade</td>
<td>Grade this year:</td>
</tr>
<tr>
<td>Assignment 2044</td>
<td>92 Baseline Grade</td>
<td>Grade this year:</td>
</tr>
</tbody>
</table>
Prof. David Malan: "We could put in just automated check but we didn’t want to. We want to do the same thing I would have done in a 30 students class - understanding solution design, not only output correctness".

Daven Farnham (in charge of assessments): “We are taking only 200 online students for the $2,050 track...as the limiting factor is hiring teaching assistants for grading".
Why Now?

- The recent LMS revolution has brought the abundance of electronic submissions: >85% in STEM courses

- Universities create new revenues via online platforms: Coursera, EdX, Lynda, 2U, Udemy, Udacity... $Bs in investments

- e-Learning and e-Tutoring are proliferating: eTeacher, Codecademy, P2P University... $Bs in investments
Will machine learning ever replace the human educator?

No.

But educators who use machine learning will replace the ones who don’t.