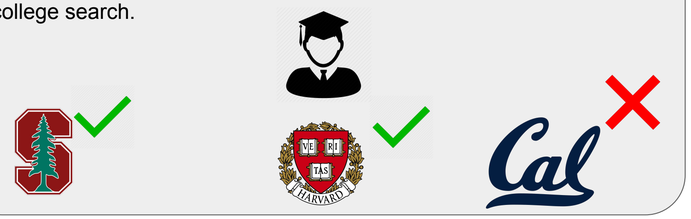
CollegeMatch

Joaquin Borggio, Sagar Maheshwari, Andre Turati

Motivation

Thousands of high school students are faced with the difficult and tedious college application process every year. Because it is so challenging and stressful to know where to start in this process, we wanted to create a system that provides students with a list of N schools that fit their profiles well. Currently, there are resources such as professional counselors or "College Personality Quizzes," however these can be expensive or have internal biases. CollegeMatch serves as an unbiased and free way to start the college search.



Data Acquisition

- Used College Scorecard Data provided by the Department of Education on all universities in the United States.
- Generating users/training data using normal deviates for each feature using DoE dataset
- Split train/dev/test sets using a 70/30 random-to-real split for train and dev, and 100% real users for our test set
- Gathered real student profiles from students via Google Form both at Stanford and at other schools

U.S. DEPARTMENT OF EDUCATION College Scorecard

ϵ - Update Algorithm

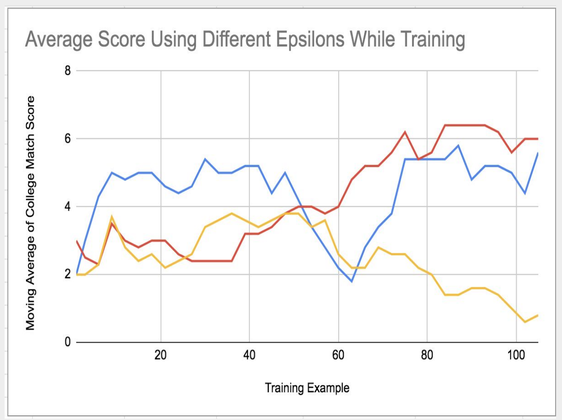
- Challenge: How to perform weight updates with SGD without a ground truth (i.e. how do we define the 10 best school matches)?
- Proposed Solution: ϵ - Update Training
  - Keep track of 2 most important features per college match
    - For positive match, increase the weight of features by ϵ %
    - For negative match, decrease the weight of feature by ϵ %

Methodology

- Combination of both Unsupervised and 'Supervised' Learning
- **Unsupervised:**
  - Weighted K- nearest neighbors approach with many modifications
  - Certain "features" had to be combined such as latitude, longitude, and if a student wanted to be close to home; a school having lower cost of attendance than student's family income is not a bad thing
- **'Supervised':**
  - Each feature has a corresponding weight which we initially tune manually using the dev set to find a better than random starting weight vector for training
  - We trained our algorithm using the ϵ-update algorithm (see bottom-left) using three different values of ϵ: 1%, 3%, and 5%
  - We choose the optimal ϵ value based on performance on test set and then evaluate performance on the test set

- User Input
  - Ex: { "SAT": 1320, "ACT": 31, Location: "Tampa, FL", CTH: "No", Locale: "Suburb", Size: "Medium", Major: "Undecided", Income: 30000, Max_Tuition: 5000}
  - Use user input to find top N school matches (reach and target).

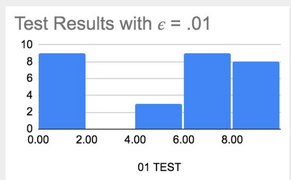
Results and ϵ Comparison

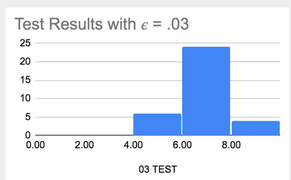


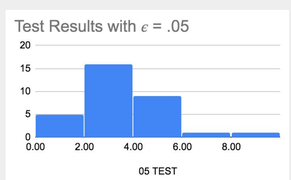
ϵ = .01 ϵ = .03 ϵ = .05

Evaluation metric while learning weights with various epsilons (Training)

Number of Positive Matches on Test Set per ϵ







Analysis

- Our evaluation metric is the number of colleges (out of 10) that are 'positive' matches (i.e. the student would reasonably apply there)
- Overall, the epsilon update of **ϵ = .03** gave optimal matches
- Epsilons
  - ϵ = . 01; gave on average good matches but updated too slowly to see good improvements and sometimes gave poor matchings
  - ϵ = . 03; converged and gave on average 6.4 / 10 positive matches
  - ϵ = . 05; diverged very quickly and gave more negative matches than positive

Test Results (Average Score out of 10)		
ϵ = .01	ϵ = .03	ϵ = .05
5.4	6.4	3.3

Challenges & Future Use

- Challenges:
  - Evaluation metric: unclear "ground truth" when it comes to the best N colleges for a user profile.
  - Generating training data and self-evaluation of these college matches for the ϵ update algorithm
  - ϵ update algorithm not guaranteed to converge
- Future:
  - Hope that students may use this as a starting point for their college journeys!
  - For future iterations of this project, we would like to have more data from students around the country and receive feedback to better evaluate results
  - Incorporate domain expert labeling colleges as good or bad matches

Joaquin & Sagar's Video Description:

https://youtu.be/rV6mE8tDJPE