Idea Description

(cool project name: Mentor Match)

Black college students mostly in STEM fields have limited access to mentors who are able to guide them in their transition from college life to navigating life in the working industry. The goal of the Mentor Match application is to expand the access black college students have to mentors outside their small niche.

According to the Harvard Business Review, the evidence is irrefutable: “people who have strong mentors accrue a host of professional benefits, including more rapid advancement, higher salaries, greater organizational commitment, stronger identity, and higher satisfaction with both job and career.” But how can a college student at a small historically black college meet a mentor outside of his or her small local niche

That’s where MentorMatch *(or some other name)* comes in. MentorMatch is an online program that matches college seniors who major in computer science at historically black colleges with experienced software engineers at top tech giants. The program also guides the mentor and mentee relationship.

Connecting Mentors and Mentees

Connecting potential mentors and mentees: (the details of *how* is still TBD) How to make suggestions (perhaps using ML) based on a set of questions that is answered when this feature is enabled or turned on for the user (open to other suggestions as well)

Introduction

The mentors and the mentees can be represented by a vector of features (this could come directly from the questionnaire, and other available metadata). The goal then is to come up with some sort of satisfaction/happiness index from the resulting matches which we should plan to maximize for our allocation system. This formulation seems to have the flavor of a constrained optimization problem where we have to maximize an objective function given some constraints. Here, the constraints could be how many maximum mentees can a mentor take and that each mentee should be paired up with someone.

Current Challenges

The biggest challenge is that we would most likely not have labeled data (satisfaction/happiness scores) from the mentor mentee relationships that we could have potentially leveraged to train some sort of supervised model. In the absence of such ground truth data, we would have to infer these happiness scores. One way would be to look at the cosine similarity scores between the vector of questionnaire features between the mentor and the mentee. The higher this score the better is the compatibility between the couple in terms of their answers to the questionnaire.

Problem Statement

The mentors and mentees who agree to this feature answer a set of questions which are then used to make allocations. In making these allocations, we plan to maximize the overall satisfaction between the matches while adhering to domain specific constraints such as maximum number of mentees a mentor can take, etc. The interpretation of such a method would be that we are doing our best to balance the happiness from matches with the specific constraints that need to be met.

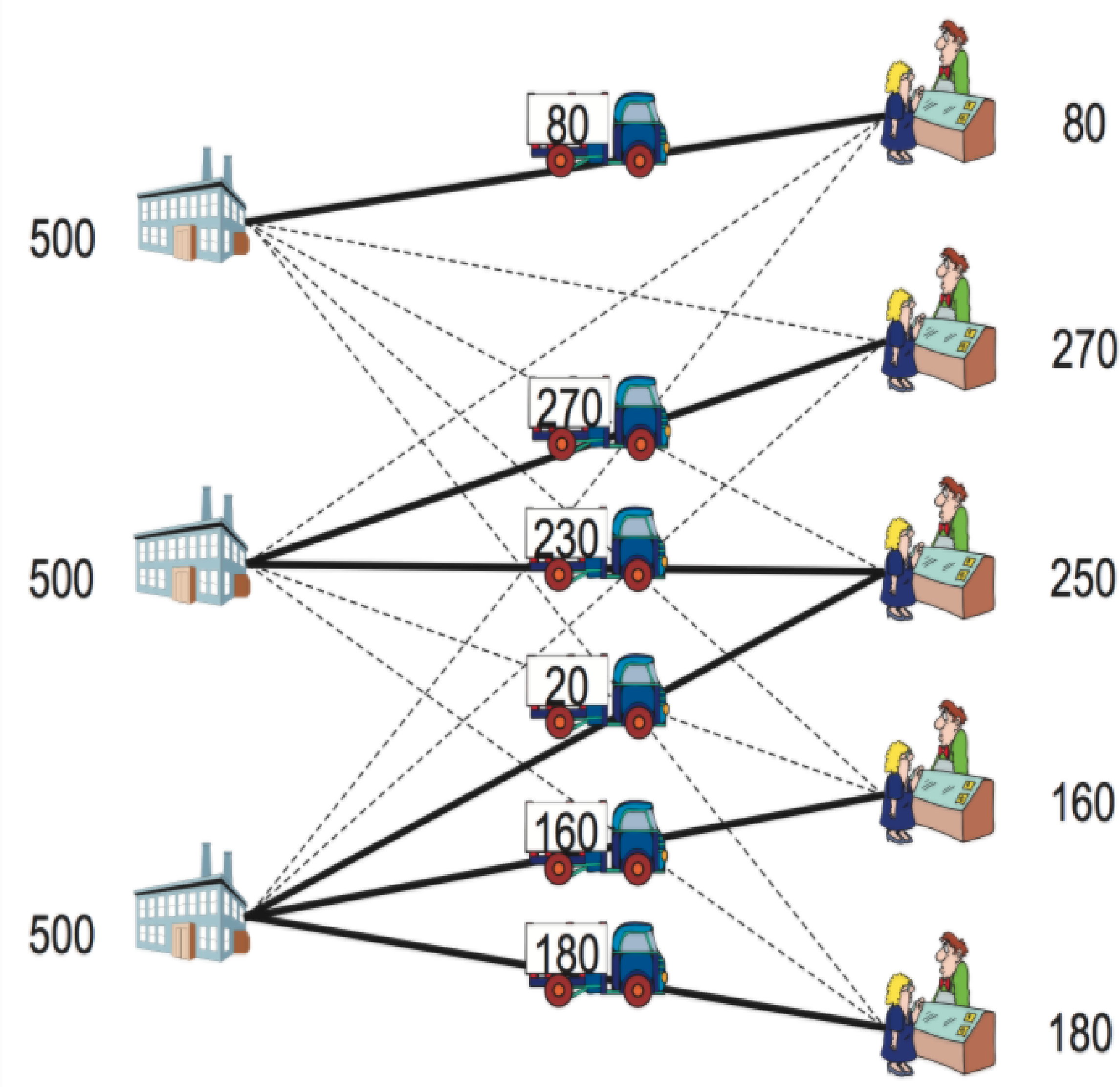
High Level Architecture

We could convert the answers from the questionnaire for everyone into numeric feature vectors. Once we have these scores, we have a m x n (m: number of mentors, n: number of mentees) where each cell can be the cosine similarity for that mentor mentee relationship pair. Then we can formulate this as some sort of assignment problem where we write our objective function and define the constraints. Programming this can be done using the many optimization packages in Python.

Or, we could pose this as a recommendation problem as well, but we would need some historical mentoring data. If we can assume the population of mentees have seeked mentorship in the past and have given us some feedback data on the mentors, we could leverage this information from the crowd to basically give a few recommendations to each mentor or mentee.

Algorithmic details

I think the initiation of this matching algorithm should start with the flavor of mathematical optimization. When we are just starting out, we do not have any ground truth feedback for any allocations as we haven't made any so far. So to start off, we need some sort of an idea around the compatibility of mentors and mentees. For this, we could leverage the responses from the questionnaire. From an algorithmic perspective it doesn't matter what the explicit questions are (assuming due diligence has been done in coining the questions to extract responses for important questions), we can then convert the responses from the questionnaire into numeric vectors which basically are encoded representations of the people. Once we have this information, we can structure this allocation problem as a straight up transportation problem.



Transportation problem

The analogy here is that the stores/warehouses are the mentors (each mentor has some capacity, maybe one can take two mentees, another can take three, etc). The shops can be thought of as the mentees, each mentee has a demand of 1 unit (can even be creative and allow multiple mentors for a mentee).

Then this entire problem can be boiled down mathematically to the following system.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{j \in J} x_{ij} = d_i \qquad \forall i \in I \\
& \sum_{i \in I} x_{ij} \leq M_j \qquad \forall j \in J \\
& x_{ij} \geq 0 \qquad \forall i \in I, j \in J
\end{aligned}
$$

Explanation of notations:

xij - indicator variable for allocation of mentor j to mentee i.
di- demand for the number of mentors by mentee i.
i goes from 1 to n (number of mentees).
j goes from 1 to m (number of mentors).
I, J are sets for the mentees and mentors respectively.
Mj is the capacity for mentor j.
cij is the cost of assigning mentor j to mentee i (cosine similarity can directly give us these costs, the higher the cosine similarity means the better the match, the upper limit of cosine similarity is 1, so if we subtract the similarities from 1 that would give us the degree of incompatibility from the allocations which is what we are trying to minimize).

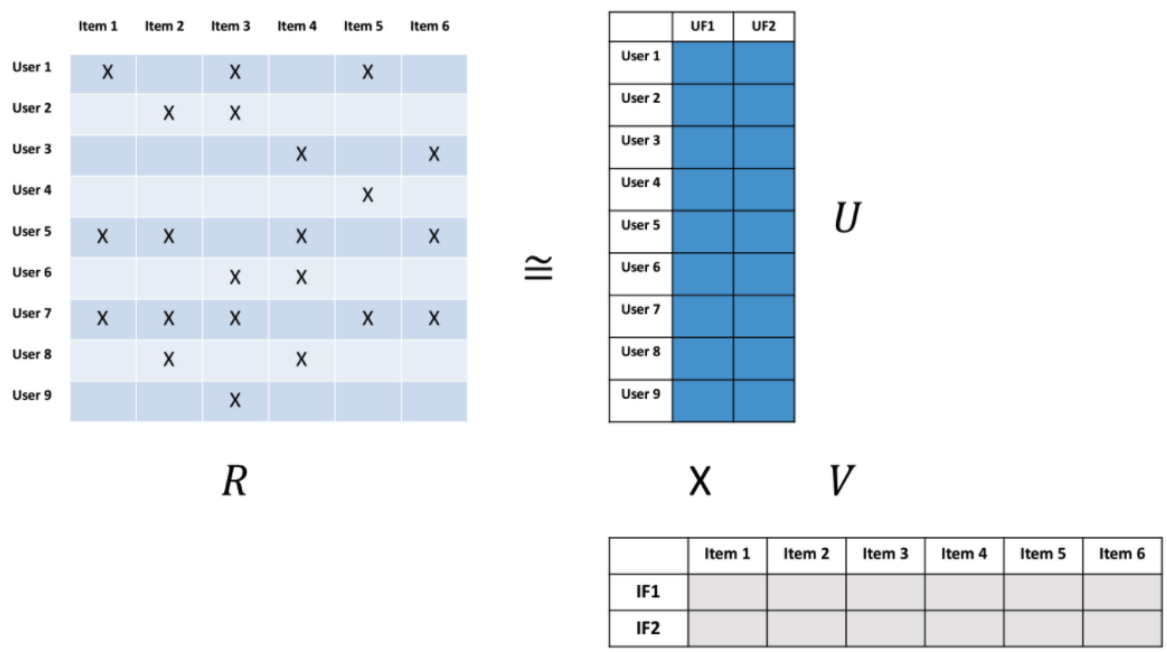
TL;DR -

The mentor mentee allocation problem can be solved using Linear optimization algorithms. Solving the problem this way would mean that we are making sure the abstract allocation system we are constructing not only ensures that the various constraints on the part of the mentors and mentees are met but also the resulting satisfaction/compatibility from the matches is maximized. This is the beauty of optimization, this is what the senior leadership would be interested in because such a system works better on the whole and it would be very hard to match such a system using ad-hoc matching rules because then we would be fine tuning for a niche group and not the overall system.

Recommendation system

This can get our system started and up and running quickly. Once we start to make allocations and get feedback on how successful the relationship was, we can foray into traditional ML by using recommendation systems. The recommendation system could be seen as another approach in addition to the optimization approach. Perhaps, at that time, we could also resort to some sort of A/B testing to decide on which method works the best. However, the recommendation systems are very powerful in embedding the mentors and mentees into common latent vector representations. We could then envision further refinements in the system where we use amazon style sidebars where in addition to our best match we also give options to the mentors and mentees to fine-tune their match themselves.

Below is a visual for how we would make the recommendation system work

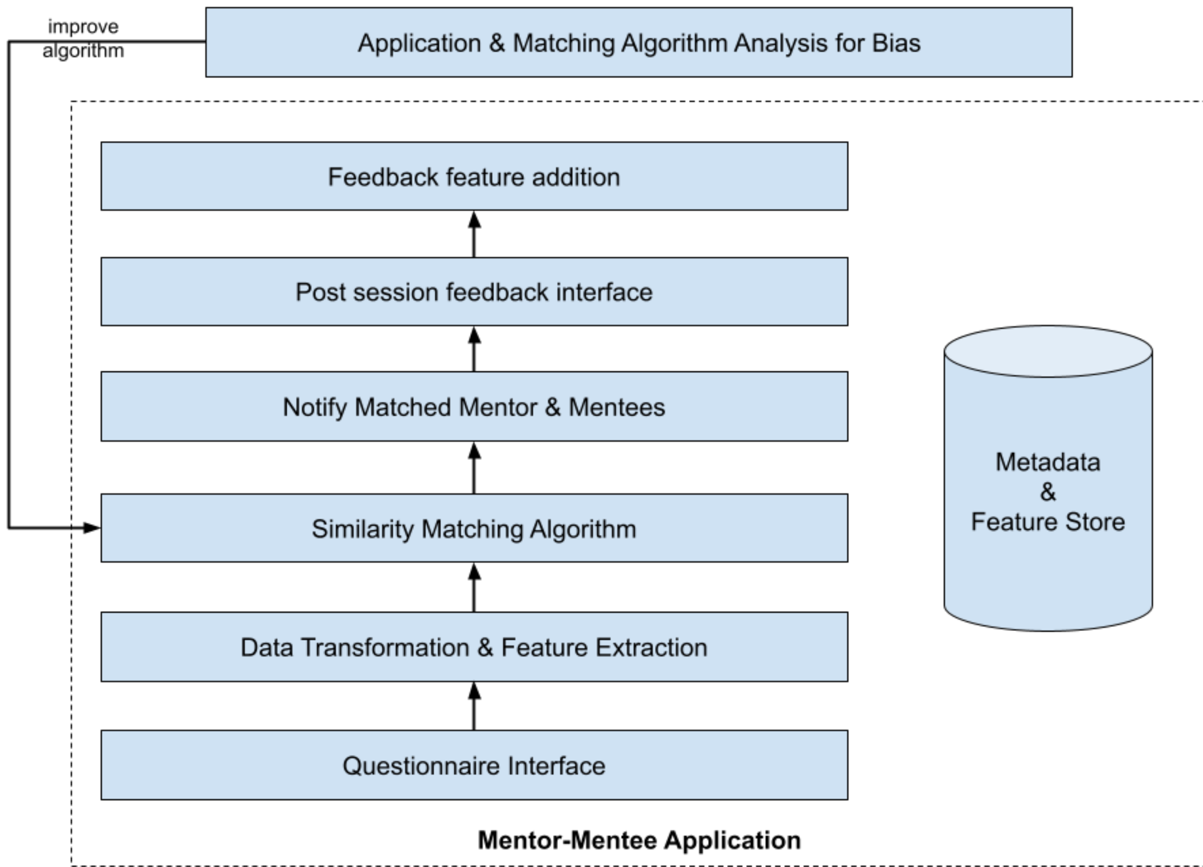


References

1. https://scipbook.readthedocs.io/en/latest/intro.html#transportation-problem
2. https://github.com/Ved-Piyush/Data-Science-Blog/blob/master/Collaborative%20Filtering%20for%20Implicit%20Feedback%20Datasets-Demo.ipynb
3. Hu, Yifan, et al. “Collaborative Filtering for Implicit Feedback Datasets.” 2008 Eighth IEEE International Conference on Data Mining, 2008
4. A Gentle Introduction to Recommender Systems with Implicit Feedback by Jesse Steinweg-Woods

---

High Level Architecture



Questionnaire Interface

- Where is this located? E.g. Chrome extension, LinkedIn, etc.
- Control questions that shed light on allocation? Multiple passes of matches? Bias score?
- Cluster mentors and mentees together → make sure mentees/ mentors of different clusters are assigned to one another
- Inferred gender, inferred race, wealth status

Data Transformation & Feature Extraction

Training Data Generation

{form {

```
  common_form_fields = ["name", "title", "seniority", "gender", "ethnicity", "interests"]
  mentor_form_fields = ["mentees_capacity"]
  mentee_form_fields = ["ordered_department_of_interest"]
}

fields {
  "seniority"=[
    {
      "name"="junior"
      "percent_of_population"=40
     }
     {
       "name"="senior"
       "percent_of_population"=25
     }
     {
      "name"="principal"
      "percent_of_population"=15
     }
     {
      "name"="manager"
      "percent_of_population"=15
     }
     {
      "name"="director"
      "percent_of_population"=5
     }
  ]

  "title": [
    {
     "name"="engineer"
     "percent_of_population"=40
    }
    {
      "name"="sales"
      "percent_of_population"=25
    }
    {
     "name"="finance"
     "percent_of_population"=15
    }
    {
     "name"="executive"
     "percent_of_population"=15
    }
    {
     "name"="hr"
     "percent_of_population"=5
    }
  ]

  "interests": [
        {
```

```
        "name"="technology"
        "percent_of_population"=30
       }
       {
         "name"="finance"
         "percent_of_population"=25
       }
       {
        "name"="sports"
        "percent_of_population"=15
       }
       {
        "name"="automobiles"
        "percent_of_population"=15
       }
       {
        "name"="gardening"
        "percent_of_population"=5
       }
       {
        "name"="fashion"
        "percent_of_population"=10
       }
     ]
"gender": [
  {
   "name"="male"
   "percent_of_population"=60
  }
  {
    "name"="female"
    "percent_of_population"=40
  }
]

"ethnicity": [
      {
       "name"="Caucasian American"
       "percent_of_population"=60
      }
      {
        "name"="African American"
        "percent_of_population"=25
      }
      {
        "name"="Asian American"
        "percent_of_population"=15} ...}]}
```

Similarity Matching Algorithm

Notify Matched Mentor & Mentees

Post Session Feedback Interface

Label Assignment

Application & Matching Algorithm Analysis for Bias

General Background/ Context

	Mentee	**Mentors**
Target Audience	Students from historically black colleges in their senior year of college with interest in technology	Tech workers who are highly committed to helping the mentees

Outreach Strategy	In-app notifications or email campaign: • Target students from historically black colleges with majors in engineering, mathematics, physics, etc. • Direct students to fill out survey • Make clear the period of the mentorship and the requirements	In-app notifications or email campaign: • Target tech workers of specific companies (eg. LinkedIn, Google, etc.) wi • • th at least X years of experience in the field • Direct mentors to fill out survey (potentially make mentors' survey longer to increase engagement during the program) → Don't need to be in the survey
Commitment Period	Recruit over one to two months Six month commitment for mentorship	
Preferred Selection (in cases where there is an over/undersupply of mentors)	Select mentees who have weaker networks (e.g. LinkedIn connections) and/or closer to graduation	Select mentors who are more senior in cases where there is an oversupply of mentors
Throughout the Mentorship Period ...	• Once mentees are matched, provide an email with the name of mentor and general guidelines of things to consider in a mentorship relationship (e.g. how to be a good mentee, how to prepare for mentorship sessions, etc.) • Before each meeting, provide prompts via notification/ email to both the mentor and mentee on things to think about/ talk about during the mentorship session • After each mentorship session, ask mentors and mentees to provide rating on the match (can use this info to refine the matching algorithm)	

Metrics to Track

- Have a 10% control group among mentees who fill out the survey to track long-term metrics

Product Adoption (by mentors and mentees, college and company):
- Email open rate
- Survey CTR
- Survey completion rate

Fulfilled Matches:
- % of mentees successfully matched

Success of Matches:
- % of mentoring sessions that mentee/ mentor attends
- Average rating of mentoring sessions (if did not attend, automatic rate of 0)

Long-Term Metrics:
- % of mentees who find a full-time job within 6 months of graduation
- Growth in LinkedIn connections of mentee

Script

- Explain the overall concept of our application [Rachael] [35 seconds]
  - mentee/mentor idea
  - Explain the matchmaking process from the perspective of the user
  - Present target audience and outreach strategy
  - Explain what happens after being matched
    - "I wonder how we got matched"
- High Level Architecture [Ajay] [25 seconds]
- How the matchmaking process works [Ved] [30 seconds]
- How we will control for potential bias [Ariena - partial] [15 seconds]
- How we will measure and improve the experience over time [Ariena - partial] [15 seconds]