Enhancing the Opportunities for Adults with Autism to Find Jobs Using a Job-Matching Algorithm

Joseph T. Bills
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation
Bills, Joseph T., "Enhancing the Opportunities for Adults with Autism to Find Jobs Using a Job-Matching Algorithm" (2022). Theses and Dissertations. 9440.
https://scholarsarchive.byu.edu/etd/9440

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Enhancing the Opportunities for Adults with Autism to Find Jobs
Using a Job-Matching Algorithm

Joseph T. Bills

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Yiu-Kai Ng, Chair
Mikle South
Xinru Page

Department of Computer Science
Brigham Young University

Copyright © 2022 Joseph T. Bills
All Rights Reserved
ABSTRACT

Enhancing the Opportunities for Adults with Autism to Find Jobs
Using a Job-Matching Algorithm

Joseph T. Bills
Department of Computer Science, BYU
Master of Science

Adults with autism face many difficulties when finding employment, such as struggling with interviews and needing accommodating environments for sensory issues. However, autistic adults also have unique skills to contribute to the workplace that companies have recently started to seek after, such as close attention to detail and trustworthiness. To work around these difficulties and help companies find the talent they are looking for we have developed a job-matching system. Our system is based around the stable matching of the Gale-Shapley algorithm to match autistic adults with employers after estimating how both adults with autism and employers would rank the other group. The system also uses filtering to approximate a stable matching even with a changing pool of users and employers, meaning the results are resistant to change as the result of competition. Such a system would be of benefit to both autistic adults and employers and would advance knowledge in recommendation systems that match two parties.

Keywords: autism, recommendation systems, employment
Table of Contents

List of Figures v

1 Introduction 1

2 Related Work 5

3 Methodology 7

3.1 The Matching Algorithm ........................................ 7

3.1.1 Novel Extensions to the Gale Shapley Algorithm ........ 9

3.1.2 Creating Profiles for Matching ............................ 9

3.1.3 Searching for Matches .................................... 10

3.1.4 Filtering Profiles ........................................ 11

3.2 The Gale-Shapley Algorithm .................................. 15

3.2.1 Generating Ranked Results ................................. 17

3.2.2 Server and Client Specifications ......................... 20

4 Validation 29

4.1 Evaluation Procedure ......................................... 29

4.1.1 The User Study ........................................ 31

4.2 The Simulation ............................................... 34

4.2.1 The Initialization Step of the Simulation ................. 35

4.2.2 The Agent Behavior During the Body of the Simulation 37

5 Conclusion and Future Work 40
List of Figures

3.1 Examples of unstable matching and stable matching . . . . . . . . . . . . . . 8
3.2 An example of the result of filtering with $n = 2$ and $m = 1$ . . . . . . . . . 14
3.3 Applying the Gale-Shapley algorithm . . . . . . . . . . . . . . . . . . . . . 16
3.4 A visual representation of matching interest vectors of users and employers . 18
3.5 A new set of candidates to be considered for matching . . . . . . . . . . . . . 19
3.6 A graphical representation of a record . . . . . . . . . . . . . . . . . . . . . 20
3.7 The layout of the client-server architecture of the proposed system . . . . . 21
3.8 An overview of the client app . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.9 The login/register page . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
3.10 A list of matches generated for a user . . . . . . . . . . . . . . . . . . . . . 24

4.1 A page of ten questions displayed to the user. The user study had the user go
through ten such pages. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
Chapter 1

Introduction

Autistic adults are an underemployed demographic, with a recent report saying that 85% are unemployed [19]. However, this statistic is not due to an inherent lack of ability in autistic adults, as is proven by the fact that some intervention can improve rates of employment. One technique that has proven to be successful in improving rates of employment and retention is job matching [4]. However, resources for job matching are limited. To tackle the problem of underemployment of autistic adults, we are using computer science to provide an automated job-matching system.

Helping autistic adults find employment provides benefits both to the individual and to society. For the individual, gainful employment leads to financial independence, which in turn leads to increased opportunities. Meaningful employment also leads to increased self-esteem and general well-being, even leading to increased cognitive ability [10]. For society, increased independence for autistic adults also results in less social expenditure on them, and their employment increases tax revenue [13]. More importantly, individuals with autism have special skills to contribute to the workplace. These talents are currently not being utilized. It is for this reason that job matching is such an attractive solution [4].

Job matching has proven to be a successful technique in helping adults with autism find employment and remain employed. Companies that have existing programs that utilize job matching include Swedish corporation Samhall and American corporation Daivergent. Samhall’s system includes matching clients’ abilities with employers’ demands, using a system that measures 25 different traits (covering sensory function, intellectual ability, mental ability,
social ability and physical ability) on 3 levels (limited, good, and high ability on the client side corresponding to low, medium, and high requirements on the employer side) [20]. Daivergent states they use AI to match vetted candidates with jobs that they extracted from descriptions using machine learning [3]. Unfortunately, there is little public information about how these corporations provide their matching beyond what details they choose to share with the public, both limiting their services to their clientele and restricting potential research contributions from studying their systems.

To make job-matching more widely available, we have developed a job-matching system that enables both users (autistic adults looking for work) and employers to autonomously create profiles and then be automatically matched. Like in Samhall’s system, this matching is done by quantifying the skills of employees and the demands of work on multiple axes. The goal is to a pick a match that minimizes the discrepancy between the user’s skills and the employer’s demands as this will minimize the number of skills the user would need to develop and the accommodations that the employer would have to make. In addition to measuring the job tasks required, demands for the application process, such as required interview skills, are also included as part of the work demands.

The system that we have developed takes into account not only the skills of the users but also their interests. This not only respects the desires of the individual but also leads to much greater productivity [17]. As we assume that the user’s preference is based principally on their interests while an employer’s preference is based principally on the ability of a worker to perform a task effectively according to their skills, interest aspects are measured separately from skill aspects. The fact that the user’s preference and the employer’s preference can differ significantly necessitates a system for finding a compromise between the two. From the interest aspects an automated ranking of employers is generated for each user, and from the skill aspects an automated ranking of users is generated. These different rankings are combined into a single match using the Gale-Shapley algorithm [7], which finds a stable match.
Stable means that no two participants may both have a higher ranked match with each other than who they were already paired with in the stable match.

Our job-matching system differs from existing systems in that it is open to use for anyone who wishes, i.e., no manual vetting is required. It is also open source, so it may be built upon for further research.

In addition to the previously mentioned benefits for individuals with autism and for society, our job-matching program for matching autistic adults with employers can also benefit people who interact with those involved in the program. Anecdotal reports from companies with programs that bring in autistic workers show that having such workers changes the attitudes of their coworkers [21]. It opens the minds of these co-workers to not only being accepting of a more diverse population, but also to considering different problem-solving strategies. The fact that autistic employees become financially independent also decreases financial strain on relatives and their generally increased well-being reflects positively on everyone they interact with.

Because the system is agnostic to whether or not a user actually has a diagnosis of autism, it can also be used to assist individuals who do not have a diagnosis but face similar obstacles to finding employment as autistic adults. People with other disorders might confront similar challenges as autistic adults in finding employment; these disorders include sensory processing disorder, social (pragmatic) communication disorder, language disorder, obsessive compulsive personality disorder, attention deficit disorder, and social anxiety disorder. While the system as presented here is designed with autism in mind, it could be expanded to include questions relating to even more disorders. As the system is designed to be competitive, anyone could potentially sign up for it, so it could potentially serve as an alternative means of finding employment for society in general.

In terms of intellectual merit, our job-matching system advances both technology for assisting autistic adults in finding jobs and the knowledge of matching systems in general. The fact that users act autonomously in this system means that the labor costs associated
with current job matching systems can be reduced. Positions are also extremely limited in existing job-matching systems that are geared towards autistic adults, so this system acts as a potential starting point for increasing access to more employment opportunities for many more autistic adults. It can potentially be extended to serve other populations as well.

The central algorithm is an extension of the Gale-Shapley algorithm so it may be applied in cases where the basic algorithm cannot be applied. While it is not the first extension of the Gale-Shapley algorithm, it is the first to use the algorithm to generate a ranking rather than a single match. While such a ranking is not useful or even meaningful to all applications of the Gale-Shapley algorithm, it works well with the assumptions made in this problem of matching autistic adults with potential employers. The idea of generating a ranking makes sense in this context as it is based on the assumption that the information the model has is mostly accurate information, but the model may have incomplete information relating to the preference of the user making his choice. Users then complete the missing information by selecting their preferred employer from the ranking. We can assume that getting this information after the ranking is done does not violate the integrity of the results because the algorithm is strategy proof from the perspective of the users [5]. This means it is in the best interest of the users to accurately give their interests as far as they can. This algorithm could also potentially be applied to other problems where the objective is to approximate a stable solution in a two-sided market. Technology that could use such problems includes applications ranging from dating apps to tools analyzing various financial markets. This novel ranking idea may help fill in information that was missed during other parts of an automated process when tackling these problems.
Chapter 2

Related Work

In addition to the Samhall and Daivergent systems, numerous websites exist which attempt to recommend jobs to autistic adults and adults with other disabilities, but like Samhall and Daivergent none of them divulge the full details of how they perform matches. One of the most notable is Mentra, which was envisioned by Jhillika Kumar while studying at Georgia Tech [16]. It starts by selecting areas that a given autistic adult is interested in and asks the reviewer to volunteer skills in those areas. HireAutism is a job-matching site run by the Organization for Autism Research [11]. Like our system and Mentra they also ask for skills and interests. The website Inclusively is geared towards hiring disabled people in general, and asks users what accommodations they request [12].

While numerous companies use their own job matching algorithms, academic research on the subject is limited. The available research exists as a specific application in the broader field of recommendation algorithms in information retrieval and extends established practices in the field. One of the oldest proposed job recommendation algorithms is CASPER [18], which focuses on clustering users based on their activity while reviewing jobs so collaborative filtering may apply. A later system by Malinowski et al. [15] instead used content-based filtering based on profiles manually entered by users. Much of the existing research on job matching relates to processing data from resumes and other sources so it can be used, with Resumatcher by Guo et al. [9] in particular seeking to match similar profiles based on extracting data from unstructured resumes and job descriptions. Others focus on new
methods for comparing the data so similar profiles may be matched, like the self-reinforcing model by Koh and Chew [14] or the collective learning approach by C. Cing [2].

One difference between job matching and other forms of recommendation is that multiple users cannot be assigned to the same job. Existing approaches, however, have neglected this issue by attempting to automate the recruitment stage in the hiring process [2], which aims to recommend to many interested job seekers instead of selecting one. Unfortunately, this process has failed to adequately serve the autistic population, so we seek to work around it. In addition to being specifically tailored to serving the autistic population, our system differs from previous approaches in that it attempts to approximate a one-to-one matching between users and employers so that those who perform poorly in the existing system will still have unique jobs suggested to them. Instead of looking at the similarity between profiles in a single vector space, it looks at the similarity in two different vectors spaces that cover aptitude and interest separately and uses this information to define a stable matching.

Even though research on algorithms for matching autistic adults with employers is minimal, there is substantial research on the subject of autistic employment in general. Of particular interest is the work of Dreaver et al. [4], who looked into tackling the problem from an employer’s perspective. In addition to suggesting matching, they emphasized the importance of external supports and of employers understanding autism [4]. Most research focuses on the perspective of autistic adults, with numerous studies supporting the efficiency of Behavior Skills Training (BST), especially when combined with prompting and audio cues. Grob et al. [8] managed to achieve a 100% success rate at teaching skills using BST combined with prompting, while Burke et al. [1] found that BST combined with audio cues is six times as effective as BST by itself.
Chapter 3

Methodology

In this chapter, we detail the design of our job-matching algorithm and discuss the novelty and uniqueness of our algorithm in matching autistic adults with potential employers based on their interests, job skills, and anticipated working environment.

3.1 The Matching Algorithm

While this is not the first matching algorithm to be applied to helping autistic adults find employment [3, 20], it is the first where all the implementation details are publicly available, and the matching algorithm used is unique in terms of its ranked job recommendations. It is based on the Gale-Shapley algorithm [7], but it is augmented with original features. A matching algorithm can fulfill different criteria, with the Gale-Shapley algorithm finding the single stable matching which is optimal for one of the two parties that it is matching [7]. A matching is defined as a one-to-one map between two parties, with the pairing between a user and an employer called a match. Furthermore, as defined earlier, a matching is stable if no two participants may both have a higher ranked match with each other than whoever they were already paired with in the match (see Figure 3.1 for examples of an unstable and stable matching for the same data set). The reason we are choosing to find this stable matching rather than fulfill a different criterion is because our matches are non-binding, with either party being free to accept or reject the match, since the user still needs to apply for the job afterwards and it’s still up to the employer’s discretion to accept the application. If a stable matching is accurate, then both the user and employer should have no reason not to accept
Figure 3.1: The red bold lines in Figure 3.1(a) show an example of an *unstable matching* for a given set of rankings. It is unstable, since Carol and Fred would rather match with each other (dashed line) than their given match (Eve and Bob, respectively), whereas Figure 3.1(b) is a *stable match* for the same data set.

The match as they would not be able to find a better partner than the one they were matched with who would also reciprocate their choice. If the recommended employer for a user is not a pairing from a stable matching, it may be to the advantage of the user or employer to ignore their matching, defeating the purpose of suggesting that match.

For a given dataset, multiple stable matchings may exist, and it is possible to find the optimal stable matching according to arbitrary objectives [24], but we are choosing to simply use the one found by the Gale-Shapley algorithm. The linear programming algorithm necessary to find other stable matchings is both *harder* to implement and *slower* than the Gale-Shapley algorithm, so there must be a compelling reason to optimize a different objective in order to justify using the more complex algorithm. We consider finding the optimal stable
matching from the perspective of the users to be a good objective for the sake of benefiting the autistic community and using the Gale-Shapley algorithm is sufficient to reach it.

3.1.1 Novel Extensions to the Gale Shapley Algorithm

Our implementation of the Gale-Shapley algorithm is augmented with two novel extensions. This includes (i) a routine for recursively applying the algorithm in order to generate a ranking for a user instead of just a single match, and (ii) a filter so that the algorithm can both run faster and run even when the number of users and employers is not equal. As there may be some inaccuracies in the stable matching due to an inability to perfectly capture all information about a user or employer that may be of interest to the opposite party, a ranking of matches is given rather than just a single match so the user may choose for themselves among the matches. Our method of augmenting the Gale-Shapley algorithm so that it may be used for ranking is unique to this project. This ranking method requires a filtering system, which is another augmentation to that algorithm that is also needed to ensure that the Gale-Shapley algorithm can be applied even with a dynamic set of users and employers. This relaxes another restriction on the Gale-Shapley algorithm, which requires a static set of users.

3.1.2 Creating Profiles for Matching

Before users can be matched with employers, individuals in both parties need to create profiles for themselves within the system. When making a profile, someone first specifies if they are looking for or offering employment, which determines if they are a user or an employer respectively. In either case, the user or employer will be walked through more questions to continue building their profile. Users will be asked questions to figure out both what jobs interest them and what skills they have. Based on their responses a numerical record will be generated with different fields corresponding with different tasks. This numerical record has two parts, an aptitude vector corresponding with their skills and an interest vector corresponding with their interests. Employers meanwhile will be asked questions about the
job they are offering to determine the qualities required to obtain and excel at the job, and what qualities the job has which may be of interest to a user. A numeric record is generated for them as well whose fields directly correspond with those in a user’s record for requirements, the larger the number in the employer’s field means the more of the corresponding skill is demanded from the user. This defines the employer’s \textit{aptitude vector}. For qualities meanwhile, the same number in the user’s corresponding field means it matches a user’s interest in that respect, defining the employer’s \textit{interest vector}. A potential system for guiding users through creating their profiles and storing the corresponding records is detailed in Section 3.2.2, while a potential set of questions and how vectors are calculated from them are also detailed in Section 3.2.2. When employers and users are matched with each other, users will rank employers with similar interest profiles.

3.1.3 Searching for Matches

After a user has created his profile, he has the option to look for jobs by pressing a button. This will trigger the matching algorithm which will then return a list of ranked jobs to be displayed for the user. The list includes the name of the employer offering the job and information about the job. The user can always press the button again to refresh the list of jobs in case activity from other users and employers has changed the results, which just runs the matching algorithm again for whatever user and employer profiles are currently in the system, but there is also a second button to reject all the jobs in the list. Choosing the latter will change the user’s record to include flags that specify that the user is not interested in those jobs and will not include them in further matches, and will also update the user’s interest components of their profile to be further away from the rejected profiles. The specific details in how this is done is described in Section 3.2.1. When running the matching algorithm, users will rank employers with similar interest vectors higher, while employers will rank users with similar aptitude vectors higher.
3.1.4 Filtering Profiles

The matching algorithm is based on the Gale-Shapley algorithm, but it includes some additional steps to ensure that prerequisites for using the Gale-Shapley algorithm are satisfied and so that it can be used to generate a ranking rather than just a single match. First, stable matchings are only defined when the two parties being matched have the same number of participants, so the Gale-Shapley algorithm itself requires that there be the same number of user and employer profiles being matched. To ensure this, a filtering scheme that guarantees an equal number of users and employers is applied so only the user and employer profiles which are predicted to be most likely to impact who the target user is matched with are considered for matching. Specifically, the filtered employers will consist of jobs the target user is most likely to apply for based on being closest to what he is most interested in, and jobs the target user is most likely to succeed at based on his aptitude meeting the employer’s requirements. Meanwhile, the filtered users will be the users the target user is most likely to compete with when going after jobs he is interested in based on having similar interests. The scheme for filtering we use was designed specifically for this project. Our scheme is based on both requesting the $n$ ($n \geq 1$) employer profiles representing the jobs the target user is estimated to be the mostly likely to succeed at, and the $m$ ($m \geq 1$) employer profiles that the target user is predicated to be the most interested that are not already being considered with the previous request. Multiple studies have confirmed that with randomly generated rankings the expected ranking for the final match is $\log(\text{Number of Profiles})$ [22], so we advise setting $n$ and $m$ to be around $\log$ of what is the estimated maximum number of users.

In our implementation, we set $n$ to 100 and $m$ to 50, arbitrary values from a range estimated to be high enough to get accurate results and small enough to run in a reasonable amount of time.

To calculate the $n$ jobs the user is most likely to succeed at, we first calculate the discrepancy between a user and an employer as taking the sum of squared differences in all fields in the employer’s profile where the employer’s requirement exceeds the user’s skill as
denoted by the corresponding field in their profile:

\[
Div_{\text{App}}(U, E) = \sum ReLU(E_{A_i} - U_{A_i})^2
\] (3.1)

where \(ReLU(x) = x\) if \(x \geq 0\), else \(ReLU(x) = 0\), \(E_{A_i}\) is the \(i^{th}\) component of the Employer’s aptitude vector, and \(U_{A_i}\) is the \(i^{th}\) component of the User’s aptitude vector.

For example, under a particular scheme a user vector with aptitude components (1.2, 2.3, 4.9) represents their interview skills, noise tolerance, and loyalty, while an employer with aptitude vector (3, 2.5, 2), meaning they put significant emphasis on presentation during interviews, their environment is moderately noisy, and they weakly desire company loyalty. The users fit for the job would be interpreted that they are sufficiently loyal and would only need minor accommodations at most to handle the noise in the environment, but they would need to work on their interview skills significantly to likely succeed at applying to the job. This corresponds with component-wise discrepancy of 3.24, 0.04, and 0, which sums to 3.28 for the total discrepancy.

The discrepancy represents extra labor the user must expend to reach the demands of the job or additional accommodations the employer must make to be accessible to the user, so a higher discrepancy means a user is a worse fit for the job. We can then take those \(n\) employers with minimal discrepancy between them and the target user by ranking the profiles in order of increasing discrepancy and keeping only the top \(n\) in the ranking. In the case of a tie where two employers have the same discrepancy from the target user, the first of the two employer profiles to be created is given priority in the ranking. The top \(m\) profiles that the target user is predicted to be the most interested in are calculated in a similar way, but with a different formula for discrepancy. This is done by comparing the fields related to interest instead of those relating to skill and including all fields in the sum, not just those where the employer’s value is greater than the user’s:

\[
Div_{\text{Int}}(U, E) = \sum (E_{I_i} - U_{I_i})^2
\] (3.2)
where $E_{i}$ is the $i$th component of the Employer’s interest vector, and $U_{i}$ is the $i$th component of the User’s interest vector.

Equation 3.2 is equivalent to the Euclidean distance between the interest fields of the profiles as modeled as vectors in a normed space. This discrepancy represents the divergence between a user’s ideal job and the given job. A potential scheme for interest includes consistency of tasks and the social culture of the workplace, with higher values denoting more consistency in tasks and a more prominent social culture. An autistic adult’s interest vector may be $(2.4, 0.9)$ under this scheme, meaning they want strong consistency in their tasks and would prefer not to interact with others while working. An employer’s interest vector may be $(3.1, 2.2)$, suggesting that the work is quite repetitive and that there is a moderate social culture in the workplace. The component-wise discrepancy is 0.49 and 1.69, which sums to 2.18.

If there are less than $n + m$ (150 in our implementation) employer profiles in the system, then all of them will be considered, and these calculations for discrepancy to get the top $n$ and $m$ employer profiles can be skipped. As a result, either $n + m$ employer profiles will be considered after filtering is applied, or no filtering will be applied and all of them will be considered, in which case we will define $E$ as the total number of employer profiles. From this we define $N = \min(n + m, E)$, where $N$ is the number of filtered employer profiles. $N$ will also be the number filtered user profiles to be considered, to ensure the number of users being considered is equal to the number of employers. Next the $N − 1$ user profiles with the most similar interests to the target user are considered so that, together with the target user, $N$ user profiles will be considered, ensuring that the same number of user and employer profiles are considered after filtering. An example of the result of filtering with $n = 2$ and $m = 1$ is shown in Figure 3.2. The reason users with a similar interest are considered is because they are the most likely to compete with the target user for their preferred job and thus affect the results of the Gale-Shapley algorithm. The corresponding discrepancy in interest is calculated in the same way as the $m$ employers the user is most likely to be interested
Figure 3.2: Users (blue) and employers (orange) being considered for matching are enclosed in rectangles. Each column is a sorted list, with the arrow on the left showing if the list is sorted by interest discrepancy (blue/solid) or aptitude discrepancy (orange/dashed). It is assumed that Alice is the target user, so all discrepancy is measured relative to her. In this example, \( n = 2 \), so in Fred and David are enclosed in the orange (dashed) rectangle, and \( m = 1 \), so Eve is enclosed in the blue (solid) rectangle after David and Fred are passed over due to already being considered. Alice herself is the grey (dot and dashed) rectangle, and \( n + m > 1 \) users closest to Alice are in the green (dotted) rectangle. Together, three users and three employers are being considered, so a matching is defined.

in, by calculating the Euclidean distance between their interest records, and keeping those with the lowest scores. To ensure \( N - 1 \) users can always be filtered, \( n + m - 1 \) mock user profiles are included within the system in addition to real user profiles. These mock profiles do not correspond with any individuals and exist only to ensure the algorithm can be applied. They are randomly generated, but from a distribution that matches that of real user profiles, including potential profiles that would correspond with non-autistic individuals in order to simulate wider competition. While these mock profiles may influence the results of algorithm as they simulate competition, they will never be target users and will never compete with actual users when users apply for jobs after being matched. The fact that the filtering never returns more than the requested number of profiles means that the matching that follows will run in constant time relative to number of profiles, improving over the quadratic time of the unconstrained Gale-Shapley algorithm, though the computational time for filtering grows.
linearly with the number profiles. As a result, the overall time complexity for matching a single user is linear.

3.2 The Gale-Shapley Algorithm

The next pre-requisite the Gale-Shapley algorithm needs before it can run is information about how each user being considered will rank each employer profile being considered, and vice versa. For this, we assume that users rank employers by how interested they are in them, while employers rank users by how likely they are estimated to succeed. These ranks are calculated in the same way they were calculated during the filtering process, by sorting the calculated discrepancy so those with the lowest discrepancy are most preferred. With the rankings generated, the Gale-Shapley algorithm can now be applied to find a stable matching.

When the algorithm starts, users are labeled as being not considered matched, but all users will be labeled as being considered matched when it stops. The Gale-Shapley algorithm consists of applying the following loop, called The Gale-Shapley Loop that matches and un-matches users until every user is considered to be matched with an employer. At that point these matches are now considered as the official matches which are returned.

The Gale-Shapley Loop

1. Every user who is not considered to be matched will be considered as a potential match to their current top ranked employer.

2. Each employer will become matched to their top ranked user as being considered as a potential match. The other users who were being considered as potential matches to them will no longer be considered to be matched and will now consider their next top ranked employer as their current top ranked employer.

Figure 3.3 shows a complete run of the Gale-Shapley algorithm on a simple dataset. The first step shows initiation, and every following step alternates between steps one and
two in the Gale-Shapley loop above. Potential matches are blue (asterisk/dotted), matches are green (plain/solid), and rejected matches are red (x-ed/dashed). Note this example does not show any cases of former matches becoming rejected (going from green to red), but such behavior is possible. The user table show how each user ranks each employer and vice versa for the employer table.

Figure 3.3: Applying the Gale-Shapley algorithm
3.2.1 Generating Ranked Results

The employer the target user is matched with is then returned as their top suggested employer. To generate the rest of the ranking, assume that the user was not interested in the most recent employer that was suggested to them. To take into account that information, the fields measuring the user’s interest would be updated to be further away from the fields in that employer’s record. This is done by subtracting a weighted multiple of the employer’s field from the corresponding field in the user’s profile for each field representing interest:

\[ U'_I = U_I - w \times (E_I - U_I) \] (3.3)

where \( U_I \) is the target user’s previous interest vector, \( E \) is the matched employer’s interest profile, \( w \) is the weight given to negative feedback, and \( U'_I \) is the updated interest vector.

As an example, if we used our previous example with the autistic adult’s interest vector being \((2.4, 0.9)\) and the matched employer’s interest vector being \((3.1, 2.2)\), with a weight \( w \) of 0.5, then the difference between the profiles is \((0.7, 1.3)\), so the updated user interest profile will be \((2.4, 0.9) - 0.5 \times (0.7, 1.3) = (2.05, 0.25)\). A visual representation of this change regarding a different example is show in Figure 3.4. The weight in both examples is arbitrary. The value of this weight is determined empirically with the goal of having users chose higher ranked matches.

The employer whom we assumed the user is not interested in would also not be considered when filtering employers. With this in mind, the rest of the matching algorithm can be re-applied with the updated information (updated interest fields and ignoring the last recommended profile), in which case it will generate a new matching and return a new match. Note that if the weight is set to 0 then the previous employer will be flagged without changing the user’s interest, so a new match is still guaranteed. (Figure 3.5 shows a new set of candidates to be considered for matching.) This new match can be returned as the next suggested employer. The process can in theory be repeated until every employer has been
Figure 3.4: The green (dotted) circle encloses the closest users in terms of interest to Alice according to her original interest vector, while the blue (solid) circle encloses the closest employers. Alice’ is at the position of the new vector after moving away from David, her top-ranked match. Alice’ is closer to George than Carol, changing who is considered in the next match. However, Eve remains closer than Henry.

ranked, but it only needs to be applied until the specified number of employers to display to the user have been ranked, at which point they are displayed to the user. This same process is used to update the user’s profile if they reject all their matches, making it so that if \(k\) results are listed at a time, then rejecting the results would cause ranked results \(k + 1\) through \(2k\) according to the original profile vector to be displayed instead; pressing it again would display results \(2k + 1\) through \(3k\), and so on. This process of iteratively applying the Gale-Shapley algorithm on filtered results to create a ranking is novel to our design. The optimal value of \(k\) depends on what is practical to display on the screen to the user while still retaining ease of use. We chose \(k = 5\).

**The Matching Algorithm for (User, Employers, Users)**

1. Define Target User as User

2. Define Potential Employers as Employers and Potential Users as Users excluding the Target User
3. Do $k$ times:

   a. Apply filter on (Potential Employers, Potential Users, Target User) to get $N$ filtered-employers and $N$ filtered-users.

   b. Use Gale-Shapley algorithm on (Filtered Employers, Filtered Users) to get the Stable-Matching map.

   c. Select the match for the Target User from the Stable-Matching map and return the Matched Employer.

   d. Define the $k$-th ranked match as the Matched Employer.

   e. Update Target User’s interest vector as $TargetUser_I - w \times (Match_I - TargetUser_I)$, where $X_I$ denotes the interest vector of the user or employer, while $w$ is the weight

   f. Define Potential Employers as Potential Employers excluding the Matched Employer.

4. Return the ranked matches.
3.2.2 Server and Client Specifications

In this section we detail the design of the server and client architecture of our job-matching system.

Overview of the Server

Records containing user profiles as well as additional information associated with a user are stored on the server. The record for a particular user contains the user’s username, password, a text description detailing whatever the user is offering, a binary flag specifying if they are an employer or not, a sequence of double precision floats representing the user’s profile vector, and a flag specifying if the user has finished creating their account (see Figure 3.6 for a sample record). It is on this server that the matching algorithm is performed. The server is designed so a user can communicate with it using a specifically designed client program, and it supports six different contexts through which the client can send messages. These contexts are Home, Register, Login, Delete, Update, and Match.
Figure 3.7: The user’s computer (yellow) communicates with the server (blue) over the internet through the client. Red messages are encrypted.

**The Server’s Contexts**

Sending a message to **Home** is used to establish a secure connection with the server. Messages to all other contexts are encrypted as they at the very least include the user’s password and often contain other potentially sensitive information as well. (See Figure 3.7 for the layout of the client-server architecture of the proposed system.)

The **Register** command is used to create an account by sending the username and password for the account that the user wishes to create to the server. As long as the username is not associated with any existing records, a default record will be created containing that username and password. By default, the text description is an empty string, the user is not an employer, the vector is set to random values, and the account is not complete.

The **Login** command retrieves the record corresponding with the username that is sent to the server and the user’s record is forwarded to the client if the password matches what is in the record.
All the other remaining commands will also only be undertaken if the password matches what is in the record for the username that is given.

Delete causes the server to remove the record corresponding with the username that is passed to it. Update replaces the record for the user with one corresponding to a serialized record that is also sent with the request. This is how all values in the record other than username and password are set.

Finally, Match returns the usernames and descriptions of the user’s top ranked matches. Before the user’s Match request may be satisfied, their account must be marked as being completed, and only completed accounts will be considered when running the matching algorithm.

Overview of the Client

The client can be run from an application such as one made using JavaFx which communicates with the server on behalf of the user. This application provides a user interface with buttons and text fields so the user can provide the client the information it needs to send its requests in a format that the user understands. The application also maintains a working model of the user’s intent based on user input. To guide the user through creating and using their profile, the application is divided into several pages that are navigated through by pressing buttons in the user interface (see the configuration in Figure 3.8). Each page defines what the user will see at that point, including text, buttons, and text fields. Some button presses also cause the client to send a request to the server, in which case the application must wait for the server response before it changes pages.

Logging in and Registering

After connecting to the server through Home and establishing a secure connection, the client will prompt the user to either register or login (see Figure 3.9). In either case, the user will be prompted to enter a username and password, and the corresponding request will be sent.
If the response from the server specifies that the user logged in or registered successfully, they will be taken to the next page in the app (see discussions given below).

Creating a Profile

If the user has not yet completed his profile, he will be walked through a series of questions to get all the information required to complete it, with the working record being stored in memory on the app that is modified as questions are answered. This working record will be initiated to the default record. Once the profile is completed, an **Update** request will be sent to the server to replace what was in the server with what the app had in memory.

Using a Profile

If the user has completed his profile, he will have the option to either modify his profile or to find matches. If modifying his profile, he can go through the questions again so he may
change any of his responses and send another Update request or delete his profile. When attempting to delete his profile the user will be given a warning before the Delete request is sent. When finding matches, the client will make a Match request and then display the response to the user (see a sample list of matches in Figure 3.10).

Note that in the current version of the app, the description is supposed to contain all information necessary for a user to choose an employer, but this may prove to be limited in practice. In future versions of the app employers may design their own pages that contain additional information, keeping the description in the list short. This page could potentially be accessed through the generated list of matches. The employer’s page could also contain direct information about how the employer answered certain questions if they give permission for the question answers to be released.
The Aptitude and Interest Scheme

These are the divisions of interest and aptitude that we used. Note that while the interface requires the specific questions to be defined, it does not specify how the questions relate to the internal interest and aptitude scheme.

Areas of **Interest** include:

1. Realistic
2. Investigative
3. Artistic
4. Social
5. Enterprising
6. Conventional

Areas of **Aptitude** include:

1. Administration and Management
2. Biology
3. Body Coordination
4. Building and Construction
5. Chemistry
6. Clerical
7. Complex Problem Solving
8. Computers and Electronics
9. Customer and Personal Service
10. Economics and Accounting
11. Fine Arts
12. Foreign Language
13. Helping People
14. Instructing
15. Management of Financial Resources
16. Mathematics
17. Mechanical
18. Medicine and Dentistry
19. Memorization
20. Monitoring
21. Negotiation
22. Operations Analysis
23. Personnel and Human Resources
24. Production and Processing
25. Programming
26. Psychology
27. Public Safety and Security
28. Quality Control Analysis
29. Repairing
30. Sales and Marketing
31. Science
32. Sociology and Anthropology
33. Speaking
34. Systems Analysis
35. Teaching and Course Design
36. Telecommunications
37. Therapy and Counseling
38. Time Management
39. Troubleshooting

These questions were taken from CareerOneStop which in turn took them from O*NET Career Exploration Tools and the U.S. Department of Labor, Employment and Training Administration. They are being used under the license found in [23]. The questions have not been modified from the form they were found in, but they are being utilized for a new purpose, so a full validation study is warranted. For their original purposes in career exploration, they have already been validated, and since we have extended the scope of career exploration the use of these questions is justified as the user questions. However, we do not have corresponding questionnaires for employers that could be used in such a matching system with the same empirical validation.

The work of the thesis can be continued to create a set of questions for the employers. Once a set of questions is settled on for employers, the following empirical study can be used to construct an aptitude and interest scheme that will match users as intended. First, a set of users and a set of employers need to be recruited, and they all need to fill out all their questions. Second, each user needs to rank each employer, and visa versa, and the users and employers will be provided with whatever information they need to make as accurate a ranking as a possible. Third, the rankings users provide of employers will be used to construct
a matrix corresponding with interest divergence scores, and the rankings employers provide of users will be used to define a matrix corresponding with aptitude divergence scores. Matrix decomposition of the first matrix will be used to define interest vectors for all of the users and employers, and matrix decomposition of the second matrix will be used to define aptitude vectors. If the matrix decomposition is accurate, this will work since the interest divergence is the same as the square of the dot product of the vectors, which has the same ordinal relationship as the rank values, and the product of the matrices defines all the combinations of dot products. For the aptitude components, it will be accurate if the employer values are always greater than the user values as only in that case is the formula for discrepancy equivalent to the square of the dot products of the vectors. Therefore, that constraint must also be placed on the decomposition. Finally, the interest and aptitude scheme are each defined by using a linear regression to map from the responses users and employers gave to each question to their calculated interest and aptitude vectors. Of all possible schemes for linearly mapping from question responses to interest and aptitude vectors, the one created using this method is best suited for recreating the rankings that the users and employers gave, and if the questions were chosen well, it should hopefully generalize well to other users and employers. If the resulting scheme for users is fairly similar to the original scheme defined by the questionnaires, then it suggests that the original traits chosen are useful for measuring interest and aptitude in jobs. Since those traits generalized well to measuring interest and aptitude in the arbitrary users in the study, there is good reason to believe they would generalize well to additional users.
Chapter 4

Validation

In this chapter, we discuss the performance evaluation of the proposed job-matching system based on the responses of adults with autism in response to an empirical study.

4.1 Evaluation Procedure

To take our first steps towards undergoing a full clinical trial in the future, we performed a feasibility trial for our evaluation. This is done to evaluate if the system is worth continuing to develop to the point where a clinical trial may be done on it in the future, or if revisions should be made beforehand.

There are no universal guidelines for feasibility studies; however, Bowen et al. [6] provided some suggestions for guiding feasibility studies for any sort of medical intervention, including an intervention to increase employment in adults with autism. They recommend conducting a feasibility study in this situation.

Bowen et al. [6] recommend eight different potential focus areas when conducting a feasibility study. Our feasibility study focused on two of these areas, which are acceptability and demand. For this feasibility study, the question to answer is whether the intervention works. For the focus area of acceptance and the question of can it work, the group recommends conducting a focus group to discover how users would use an intervention in practice. Meanwhile, for the focus area of demand they recommend sending a survey to measure interest from the target population. For our feasibility study, we adapted these two principles
to design a single user study that will both discover how users would use the system in practice and discover if they are interested in using a fully functioning version of the system.

In our user study, we focused only on the users’ acceptance of their interface, while leaving the study of employer acceptance for future work. Instead, mock aptitude and interest vectors of companies were used for creating the employers’ profiles. These were based on real job postings, but no rigorous process was used to convert from qualitative job postings to a quantitative profile. Users have been informed that the version of the system they used did not accurately rank the employer profiles, but it did display employer profiles to them in the same manner as the final version of the system would.

While using our system, users were monitored to see if they were able to successfully use the system to generate a list of matches. Users were also monitored to see if they used the information in the list to initiate the application process with any of the employers. Any obstacles to use uncovered during the trial were also recorded. The users were informed that they were not being told to complete any job applications, but merely demonstrate success in initiating a job application, and they were not expected to apply for any of the jobs that were recommended to them. As a hypothetical example to illustrate the process, if an employer accepts applications for the job through email, then composing an email addressed to the address that accepts applications for the job would be counted as a successful initiation of the application process, and the user would not be expected to write or send the email.

After using our system, users were asked whether they would be interested in using a working version of our system. If they were uninterested, they would be asked if they would be interested if any changes were made to the system, and if so, they would be asked for suggested changes to the system. By default, we assume the highest entropy null hypotheses where an equal number of users will fall in any two categories of success and failure or acceptance and rejecting, so finding a statistically significant majority would result in rejecting these null hypotheses. If the statistically significant majority of the users in the feasibility study could successfully initiate an application, and a statistically significant majority said they
were interested in using the fully functioning version of the system, then the feasibility study could be considered a success. If the study was unsuccessful, then both notes on obstacles users faced during the study and user feedback would be used to redesign the system, and a second feasibility study would be performed with the necessary modifications.

This project has not only medical implications, but also technical implications. We have already run simulations to confirm the system not only can work but does work from a technical standpoint. In the ideal circumstances of the simulation, it leads to more individuals being hired than for comparable recommendation systems. Additional simulations could also be performed if further testing is needed, such as testing how the system would perform if interest and aptitude are correlated, or if other users around the user are using a different matching system.

While we need to test our solution on end users to see if the designed system is superior to other solutions in practice, we can prove that under certain conditions some aspects of our solution are superior to other solutions. Specifically, we have tested our matching algorithm in a simulated environment, showing that our matching algorithm is leads to better results than comparable algorithms for the case that was simulated. This simulation was coded in Java and ran in the IntelliJ coding environment.

4.1.1 The User Study

Set Up

For our user study, we recruited students from ScenicView Academy, a nationally-recognized nonprofit transitional school for young adults with autism and other learning differences in Provo, Utah, as well as from Brigham Young University. We set our system up on six desktop computers at ScenicView. Each student who was involved had scripts for running the system added to their account at ScenicView. The students at ScenicView were instructed to first run the server script to ensure it was running. They were instructed to next run the client
Figure 4.1: A page of ten questions displayed to the user. The user study had the user go through ten such pages.
script to use the system; they were told to stop once they found a job. They were instructed to then run the third script to take them to the survey page where they would complete a feedback form. The students at Brigham Young University had the system run for them on a computer in the Information Retrieval Lab and took the same survey afterwards. Users in both locations were given a high-level description of the algorithm in a briefing beforehand, and the survey included a link to a paper describing the algorithm in detail that the users could read if they chose to do so.

The version running on these computers included 20 employer profiles that were manually created using public job postings from Hire Autism, Chronically Capable, and Torre. The n employer profiles, representing the jobs the target user is estimated to be the mostly likely to succeed at, is set to two, whereas the m employer profiles, representing the jobs that are predicted to hold the most interest for the user and that are not already included in the n employer profiles, is also set to two. In addition, three mock users were created from data provided by volunteers from Brigham Young University’s Information Retrieval lab. Due to the small number of jobs the number of recommended jobs displayed, k, was reduced to three, and the reject button was not implemented.

**Experimental Results**

Eleven students completed the study and reported their response, eight from ScenicView and three from Brigham Young University. All of the students who reported their response were successful in answering all of the questions in the survey, but only five reported finding a job at the end. Based on the response from one user who stated the URLs didn’t work, we believe this was due to a bug unrelated to either the matching algorithm or the interface. Another user reported that the job pages had expired, which is again independent of the functionality of the system.

A few users praised the simplicity of the design. Eight of the eleven students stated they are interested in using a full version as part of a second study. Ten of the eleven students
said they would recommend the system to someone else, with six of those ten requiring modifications be made first. With the small sample size, ten students recommending the system is the only statistically significant response, which has a p-value of 0.007931 under the one-tailed binomial test. That is less than 0.05, the standard value in psychology, so the null-hypothesis of the results being random can be rejected.

From this we conclude that the idea of the system is appreciated, but a second user study should probably be conducted with an improved interface before a clinical study is conducted. Based on specific feedback, planned changes include adding more questions, introducing the interest and aptitude sections before presenting the questions, randomizing the questions in each section, using a duller color for the background, and adding visual contrast between pages and questions. To resolve the issues with URLs and external pages, we will include job pages inside of the next version of the app.

4.2 The Simulation

In this section, we describe the technical simulation for showing the system works in ideal circumstances.

For the simulation, we created a workable system where agents simulating users and employers continuously enter the system and leave when they are either hired or hire someone. Users in the system request matches and apply to one of the matches suggested to them. Only jobs within the matching system are considered, and jobs will only receive applications from users in the system. After a user applies to the job that an employer is advertising, the employer will decide whether to hold or reject the applicant. Eventually employers decide whether to hire an applicant that they held. A hired user is removed, along with the employer that hired the user, and the count of successful hires is incremented. The simulation ceases after a set number of steps, which was 100 in our simulation, and the number of successful matches is returned. This count is used to compare different matching schemes, which are treated as individual algorithms in this empirical study.
During the simulation process, we used four different matching schemes, denoted “Matched,” “Interest,” “Aptitude,” and “Mixed,” so we could show how the algorithm in our matching scheme compares to the algorithms in similar matching schemes. *Matched* refers to ranking employers using our own matching algorithm, while *Interest* and *Aptitude* rank employers by interest and aptitude divergence, respectively. *Mixed* refers to ranking each employer by the sum of his aptitude and interest divergence when that employer is being recommended to a target user. User and employer values are randomly generated from specific distributions, with independent interest and aptitude distributions. As these different schemes are based on making matches from the same distribution of vector spaces, they are comparable, and we know that differences between these results must be due to the matching algorithm itself rather than due to what data they utilized. Our matching scheme differs both from the comparable schemes and other existing job matching schemes in that the former looks at the competition along two aspects to find stable matchings, while other matching schemes just try to optimize a single aspect, such as discrepancy between employer and employee. For each matching scheme, we ran the simulation 100 times, and calculated the average number of successful matches, as well as the variance of successful matches in the sample, so that *null hypothesis testing* could be performed. Based on these statistics, the null hypothesis that our matching scheme performs equally well or worse to comparable matching algorithms, in terms of average successful matches under the parameters of the simulation, can be rejected, proving that our method is performed better in the case that was simulated.

### 4.2.1 The Initialization Step of the Simulation

At the beginning of each run of the simulation, a series of multivariate Gaussian distributions across the vector space for profiles are defined and are labeled as either used for generating users or for generating employers. These distributions were themselves generated from a uniform distribution for their parameters, i.e., mean and standard deviation, that are based on the intervals that responses to questions are defined over. In our simulation, the vector
space for profiles has five dimensions of interest and five dimensions of aptitude so that each Gaussian distribution has ten parameters to define its mean in each dimension and ten parameters to define its standard of deviation in each dimension. We defined three of these Gaussian distributions for users and three distributions for employers and sampled the mean parameters for these distributions from the uniform distribution on the interval [0, 7] and the standard of deviation parameters from the interval [0, 2]. The uniform distributions correspond with questions that have seven radio buttons, meaning a user can select one of seven options as a response for each question, while allowing for extra variance. These distributions will then be sampled from when defining the true aptitude and interest vectors of the user generated from them, which is separate from the estimated aptitude and interest vectors that whatever matching algorithm is being used refers to. The count of total hires is also initialized to zero at the beginning.

A user’s estimated vectors are generated by the simulation to create the profile of the user. This is done by first sampling a value for each component for a Gaussian distribution whose mean is the corresponding value in the true vector, and whose standard deviation is a set parameter. The value is clamped to the range of the questionnaire so that it is set to the maximum or minimum value in the range if it exceeds them, with our range being [0, 7]. This is the value we used for the corresponding component in the estimated vector.

After the distributions are defined, but before the body of the simulation begins, the mock users are generated from the user distributions. As specified in our matching algorithm, there are $n + m - 1$ mock users, which was 199 in our implementation as we set our matching algorithm parameters $n$ and $m$ to 100 for this simulation. When each mock user is generated, one of the user distributions is selected with equal probability, and the mock user’s true vector is sampled from the distribution. The mock users then create their profiles before the body of the simulation beings.

The body of the simulation begins by queuing a step event to an event loop that is then activated. The event loop is a queue that will execute events in the order that they
were queued. The step event is requeued every time it is executed, allowing a step of the simulation to be defined as all events that are executed between two step events. All agent actions are queued into this event loop to ensure they are simulated as occurring during the same step in time. The step event is executed 100 times in total to simulate the 100 steps, after which the simulation stops, and the number of successful matches is returned.

4.2.2 The Agent Behavior During the Body of the Simulation

Every time the step event is executed, new users or employers will be generated from the same geometric distribution to simulate users continuously entering the system. This geometric distribution is defined by its stopping probability, which is 0.3 in our simulation. 0.3 is the probability after each user or employer is generated that the step event will end. Each time after a user or an employer is generated, an event for making their profiles will be queued. Once a user has created a profile, the user will queue an event to request matches. Out of the list of given matches, the user applies for the one that (s)he likes the most in terms of interest as based on the true vector’s values. The user will then wait until (s)he has received the rejected or hired confirmation. If rejected, the user will request matches again by queuing another match event and the application process will repeat. A constraint is that a user will not apply to the same job twice. If a user has applied to every job, (s)he will wait before refreshing the job list by queuing another match event to be executed next step.

In addition to the *aptitude vector*, an employer will have a threshold for each component of the vector, and these thresholds denote the requirements for a job. After an employer creates his/her profile, the employer will queue a wait event and stick around until (s)he has users in his/her application queue. If an employer has applications in his/her queue, (s)he will test the first user in the queue according to his/her threshold and remove him/her from the application queue. If a user is tested and the user does not meet the threshold in some component, meaning the threshold for that component is greater than the value in the user’s true aptitude vector, the user will be rejected. If the employer rejects a user while not holding
any users, then the employer will lower the threshold to be between the previous value and the value of the rejected user’s component according to a linear function, simulating the employer becoming more willing to accommodate as they are unable to find someone who can fulfill the requirements. In our simulation, the adjustment weight was set at 0.5, meaning the new threshold was set halfway between the old threshold and the user’s component. If the tested user meets all the thresholds, (s)he will be held if no user is currently being held. Once an employer has held an applicant, (s)he will wait a fixed number of steps before hiring the applicant. In our simulation, the wait lasts ten steps. If a tested user both has a lower discrepancy than the held candidate and is accepted based on the thresholds, then (s)he will be held and the previously held user will be rejected. However, if a tested user has a lower discrepancy than the held candidate but is rejected due to not meeting the threshold in some component, the employer will lower the threshold. This behavior simulates the employer moving to accommodate users that are generally better for the position, but who are not currently being accommodated. Once a user is hired, both the user and the employer are removed from the system and will no longer be considered by the matching algorithm.

Results

Given below are the results for the simulations that we ran:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Matched</th>
<th>Interest</th>
<th>Aptitude</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>69.72</td>
<td>48.99</td>
<td>42.1</td>
<td>48.47</td>
</tr>
<tr>
<td>Variance</td>
<td>130.53</td>
<td>82.68</td>
<td>84.20</td>
<td>96.82</td>
</tr>
<tr>
<td>Z-score for Matched being greater</td>
<td>N/A</td>
<td>4.49</td>
<td>5.96</td>
<td>4.46</td>
</tr>
<tr>
<td>Probability of z-score under null hypothesis</td>
<td>N/A</td>
<td>.000004</td>
<td>2.0E-9</td>
<td>.000004</td>
</tr>
</tbody>
</table>

Psychology uses a p-value of 5, meaning that if the probability of the z-score under null hypothesis is less than 0.05, then the results are statistically significant, and the null
hypothesis should be rejected. In this case, all the probabilities are well below 0.05, and thus we can safely conclude that our method is more effective than comparable methods for the given parameters.

**Summary**

While the simulation was only run for the given parameters, these parameters were chosen arbitrarily within a simulation scheme designed to emulate the behavior of real-world agents. The fact that our job-matching algorithm works better for the certain set of parameters that were tested indicates that this approach may be better in general.
Chapter 5

Conclusion and Future Work

The problem of developing job-matching systems for autistic adults is still ongoing, but progress is being made. From our study we found that autistic adults are interested in this sort of matching program, and that this program has the potential to increase the number of autistic adults who are hired. We expect that existing projects working on helping autistic find employment may consider using our algorithm in order to improve their own results. Our matching program is promising for improving the general welfare of autistic adults as well as increasing productivity.

We plan on continuing to refine our system in the future. Since no users reported issues with the number of questions, we plan on adding questions specifically related to autism to future versions of the system and seeing if the design is still feasible. These questions will be taken from various empirically verified questionnaires. We also plan on adding general questions related to the logistics of employment such as those related to location and work time. In addition to adding more questions, we will improve the general quality of the interface before moving onto a full clinical study. We will retain the focus on simplicity, but might add back buttons and potentially add the client to a website.

Improvements may also be made to the algorithm. One potential improvement is modifying the discrepancy formula so the metric is specific to the areas being compared rather than using a single formula for all areas. This change will help the discrepancy formula better measure distance in non-Euclidean space, such as the surface of the sphere, so that it
can model things such as geographic coordinates. Weighting can also be applied to each area that can be adjusted based on the preferences of both the user and employer.

With this in mind, there are numerous opportunities for future research extending from the research work in this thesis. We expect that the current findings can be put to good use.
References


