RICE UNIVERSITY

A Clustering Algorithm for University Admissions

by

Naomi Beth Reed

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Richard A. Tapia, University Professor
Chair
Computational and Applied Mathematics

Mark Embree, Assistant Professor
Computational and Applied Mathematics

Dennis D. Cox
Professor
Statistics

Rudy Guerra, Professor
Statistics

Brad Peercy, Postdoctoral Associate
Computational and Applied Mathematics

HOUSTON, TEXAS

AUGUST 2006
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 1441850
Copyright 2007 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346
ABSTRACT

A Clustering Algorithm for University Admissions:

Exploring Applications Quest

by

Naomi Reed

In 2003 The Supreme Court declared that all government funded universities, which choose to consider race in their admissions processes, must utilize a holistic process. A holistic process includes a thorough evaluation of all aspects of each applicant. For larger universities this type of admissions process would be very taxing. A computer scientist from Auburn University created an algorithm, Applications Quest, to handle large quantities of applications in a way that would evaluate applicants holistically with a computational tool. Applications Quest utilizes the Euclidean distance measure, Similarity matrices, Divisive Clustering, and Random Selection. This algorithm produces a diverse admittance class for a university. In this research we simulate this algorithm and run tests with hypothetical Rice University data. Ultimately, we are left with the following question: Can a computational use of arbitrary difference account for human qualities that define certain social phenomena, such as underrepresentation in higher education?
Acknowledgements

In completing this research I would like to acknowledge the following people: Dr. Richard Tapia (Chair), Dr. Brad Peercy, Dr. Juan Gilbert, Dr. Dennis Cox, Tamara Siler, Carlos Garcia, Dr. Rudy Guerra, Dr. Mark Embree, Dr. Michael Trosset, Dr. Kennard Reed Jr., Dr. Judith Reed, Marjorie Reed, Stellios Karnezis, Kary Green, Daria Lawrence, The Alliances for Graduate Education and the Professoriate, and the National Science Foundation.
Table of Contents

Abstract ........................................................................................................... Page ii
Acknowledgements ....................................................................................... Page iii
Table of Contents .......................................................................................... Page iv
A Clustering Algorithm for University Admissions (Text) ......................... Page 1
Bibliography .................................................................................................. Page 35
Appendix A (Code) ...................................................................................... Page 37
Appendix B (Clusters) .................................................................................. Page 43
Introduction

What is computational diversity? How do we measure the diversity of a population, a population that supposedly represents the demographic ideology of the University? Diversity can be described as, “The inclusion of individuals with diverse backgrounds, characteristics, and attributes” (Tapia 2005). Underrepresentation refers to the demographic discrepancy that marks certain groups in higher education. I am referring to the near inactivity of certain racial groups in higher education when I use the term underrepresentation. More specifically these racial groups are American born Black and Hispanic students, women, and Native Americans (Tapia 2005). Minority representation in colleges and universities has suffered greatly in the post-Civil Rights era and we have seen an enormous backlash from the progression and maintenance of structural and de facto inequality. Can a system that positions diversity as the goal address the implications of underrepresentation? Can arbitrary difference account for specific structural biases? I claim that diversity and underrepresentation are separate concepts that operate with respect to separate ideologies. If we position diversity as the goal and we define this diversity in a vacuum irrespective of historical, political, and cultural context we will be unable to affect the social forces that fuel underrepresentation. We must acknowledge not only the existing population but also the social biases that skew the demographics of this population in higher education. Ultimately we must keep in mind that utilizing difference does not necessarily achieve equality.

Further how do we quantitatively represent these concepts? Computing diversity through mathematical and algorithmic means can be an interesting quantification of diversity and underrepresentation; however, are these means fair? Can we discuss social
bias in the language of mathematics and computer science? We must first accept the fact that we objectify and reduce human complexity to the rigid labeling of Euclidean space and proceed to group individuals based on their associations to each other with respect to their finite numeric and nominal attributes. In this paper I will describe the social motivation for exploring computational diversity. Through a look at a clustering algorithm for university admissions I will explore the issues that arise when certain computational means are used in place of a subjective system. More specifically, we will see how a program entitled Applications Quest works and might affect the admissions process for a hypothetical group of Rice undergraduate applicants. Our discussion begins with the 2003 University of Michigan Supreme Court cases.

Chapter 1: The Michigan Cases

Applications Quest was created upon the conclusion of the 2003 Michigan Cases. These cases, Grutter v. Bollinger and Gratz v. Bollinger, introduced a new approach to achieving diversity on university campuses. The Supreme Court requested that universities that wish to consider “ethnicity and race as one of many ‘plus’ factors to achieve educational diversity,” conduct a “holistic” admissions process (Abrams 2006: 3). Initially, the University of Michigan conducted undergraduate admissions through a policy that numerically determined which students would be admitted. Each student received an evaluation score that represented a sum from the compilation of numerical values assigned to each attribute from the application.¹ In order to have been admitted a student would have to receive an evaluation score that was greater than a certain cut-off,

¹ Points were assigned to students on a hierarchical scale with respect to each attribute. E.g. if race was being considered, the numerical assignments might look like the following: White=6, Asian=7, Hispanic/African American=10.
so that the more points that are received for a certain attribute the more likely it is that that student would be admitted. Most attributes that were manipulated in this manner were not controversial, but when the University of Michigan applied this approach to race and ethnicity, a question of fairness arose. The problem with this scale was that non-minority students felt that it was unfair to award a minority student more “race points” in order to improve their chances of being admitted. They felt that race should not carry a weighted numerical value, and should perhaps not be considered at all (Franken 2003). Jennifer Gratz, a rejected undergraduate applicant for the University of Michigan, legally protested Michigan’s admissions policy as she claimed her reason for being denied admission was do a “super bonus” given to minority applicants (Franken 2003). The Supreme Court Ultimately found the universities undergraduate policy to be unconstitutional, and consequently ruled in favor of Gratz.

The University of Michigan’s law school also considered race and ethnicity in the admissions process. Barbara Grutter, a rejected law school applicant, claimed that racial discrimination resulted in her being denied admission, as the university treated race as a “plus factor” (Abrams 2006: 3). However, the Supreme Court claimed that the law school policy “is a highly individualized, holistic review of each applicant’s file, giving serious consideration to all the ways an applicant might contribute to a diverse educational environment” (Alger 2003: 2). Consequently this policy was upheld and ultimately became the standard for all government funded institutions that wished to consider race in admissions. These institutions are currently required to abide by the following:

Universities may consider race or ethnicity as a “plus factor” in the context of individualized review of each applicant, and admissions programs must be “flexible enough to consider all pertinent elements of diversity in light of the particular qualifications of each applicant.” Institutions may not, however, “establish quotas for
members of certain racial groups or put members of those groups on separate admissions tracks (Abrams 2006: 2).

A holistic evaluation is now the standard for most admissions policies. The overall benefit of implementing this type of policy is the possibility of “educational diversity” (Garcia 2006: personal communication). Educational Diversity is a term that refers to the idea that “racial diversity in colleges and universities can help enliven classroom discussions, break down racial stereotypes, and prepare students for success in our increasingly global marketplace” (Coleman and Palmer 2004: 13). The Supreme Court declared that

the diffusion of knowledge and opportunity through public institutions of higher education must be accessible to all individuals regardless of race or ethnicity…In order to cultivate a set of leaders with legitimacy in the eyes of the citizenry, it is necessary that the path to leadership be visibly open to talented and qualified individuals of every race (Coleman and Palmer 2004: 13).

Clearly the Michigan cases encourage and legally require a holistic process that may consider race and ethnicity as a means to achieve educational diversity for the undergraduate university experience. How can a major, public institution operate under a holistic admissions process in an efficient and somewhat objective manner, when in fact holistic consideration is inherently subjective and consequently vulnerable to questions of fairness? Computational Diversity could provide one answer to this new stipulation on admissions policies. Applications Quest provides an appropriate example to consider when trying to understand computational diversity.

Chapter 2: Applications Quest

---

2 The admissions committee at Rice University utilizes a holistic process.
Juan Gilbert, a computer scientist from Auburn University, felt that the Supreme Court decision provided an opportunity to computationally impact university admissions, as he believed that a holistic evaluation could be achieved through the implementation of a computer program entitled Applications Quest. This algorithm utilizes a variety of mathematical and statistical methods which include the Euclidean distance measure, similarity matrices, divisive clustering, and random selection. Applications Quest operates through an algorithm that creates points in \( n \)-dimensional space from the applications that the university receives. Each student has a finite number of attributes that is equal to the number of attributes of every other student so that each attribute can be treated as a dimension. Once each student is understood to be a point in \( n \)-dimensional space, a distance measure is computed. Using the squared Euclidean distance measure Applications Quest performs \( \frac{n(n-1)}{2} \) operations in order to calculate the distance between each student with each other student. These distances are then stored in a similarity matrix. Now that the most costly operations have been performed, Applications Quest references this matrix in order to perform the clustering part of the algorithm. Applications Quest uses a divisive clustering method where initially all of the points are treated as one cluster and then subsequently divided until the desired number of clusters are accomplished. Once the clusters are formed a point is chosen at random\(^3\) from each cluster in order to create the most diverse set.

Initially students' applications are received by an admissions committee and each attribute on the application would be put in two possible categories, nominal or numerical.

\(^3\) Juan Gilbert's version of Applications Quest calculates a difference index that allows him to determine which of the randomly selected sets of students is the most diverse or contains individuals who are the most different from each other. We simulate this step as well.
The numerical attributes are those attributes that are already numerical such as the SAT score, the GPA, or the class rank. The nominal attributes are those attributes that are not easily quantifiable such as race, state of residence, gender, or country of citizenship. All of the numerical attributes are evaluated together and all of the nominal attributes are evaluated together. The formula for determining the distance between two students is a variation of the typical Euclidean distance measure. In the equation below we see that \( d \) represents the distance, \( k \) is a variable that runs through each of the attributes, and \( i \) and \( j \) are the indices of the two students that we are comparing. We multiply by 100 in the final step so that the distance is a number between 1 and 100.\(^4\)

\[
d_{ij} = \sum_k \left( \frac{(a_{i,k} - a_{j,k})}{\text{maxval}_k} \right)^2 + \sum_k (1 - (a_{i,k} = a_{j,k}))
\]

\[
d_{ij} = \hat{d}_{ij} \times \left( \frac{100}{n} \right)
\]

Let us consider a specific data set and compute by hand a few of the distance measures.

This chart represents a few characteristics of 5 hypothetical students.

<table>
<thead>
<tr>
<th>GPA</th>
<th>SAT</th>
<th>Essay</th>
<th>Race</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>3.5</td>
<td>1450</td>
<td>5</td>
<td>W</td>
</tr>
<tr>
<td>S_2</td>
<td>3.8</td>
<td>1420</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>S_3</td>
<td>2.4</td>
<td>1250</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>S_4</td>
<td>4.0</td>
<td>1200</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>S_5</td>
<td>2.4</td>
<td>1310</td>
<td>4</td>
<td>H</td>
</tr>
</tbody>
</table>

If we compute the distance between S1 and S2 we would have

\(^4\) Juan Gilbert felt that a number between 0 and 100 would be easier for an admissions committee to compare with other values. It is viewed as a percentage so that the final value can be a percentage of difference.
\[ d_{12} = \left( \frac{3.5 - 3.8}{4.0} \right)^2 + \left( \frac{1450 - 1420}{1600} \right)^2 + \left( \frac{5 - 3}{10} \right)^2 + 1 + 0 \]

\[ d_{12} \approx (0.075)^2 + (0.001875)^2 + (0.2)^2 + 1 + 0 \left( \frac{100}{5} \right) \]

\[ d_{12} \approx 20.91953125 \]

So that when we compute the other distances between each student with each other student our similarity matrix would look like the following.

\[
\begin{bmatrix}
0 & 20.9195 & 42.6250 & 42.6008 & 41.8656 \\
0 & 22.6758 & 40.6281 & 42.7445 \\
0 & 43.4195 & 20.2281 \\
0 & 24.0945 \\
0 &
\end{bmatrix}
\]

We will explore more examples in greater detail once we have thoroughly explained each step of the algorithm separately. Now that we have a similarity matrix Applications Quest can proceed to cluster students together. At each cluster iteration, Applications Quest references this matrix in order to determine which data points or students to cluster other students with. If we were to perform one cluster iteration for this data set we would note that 43.4195 is the largest number in the matrix so we now know that S3 and S4 are the two most dissimilar students in this set. We would position these two students as centroids and figure out which of the two students the other students are the closest to and group accordingly. We will see a clear example of clustering in a later section.
Finally once we have attained the appropriate number of clusters we can randomly select students from each cluster so that ultimately we will have a very diverse group of students. In order to fully understand Applications Quest we must consider the three main steps in the algorithm, computing the squared Euclidean distance measure, building the similarity matrix, and clustering the data points.

**Chapter 3: Squared Euclidean Distance Measure**

Most people are familiar with the traditional Euclidean distance measure.

\[
d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}
\]

Here we see that \(d(i, j)\) is the value that will represent the Euclidean distance between two points, \(X_i\) and \(X_j\) (Kaufman and Rousseeuw 1990: 11). Applications Quest performs a similar computation when computing the squared Euclidean measure. We denote this measure by the following formula because we are combining both numerical and nominal dimensions when computing the distance between applicants.

\[
\hat{d}_{ij} = \sum_{k}^{\text{numeric, opinion}} \left( \frac{(a_{i,k} - a_{j,k})}{\text{maxval}_{k}} \right)^2 + \sum_{k}^{\text{nominal}} (1 - (a_{i,k} == a_{j,k}))
\]

\[
\hat{d}_{ij} = \hat{d}_{ij} \times \left( \frac{100}{n} \right)
\]

All of the numeric values are normalized to a value between 0 and 1. The opinion attributes, such as an essay score, assigned to certain attributes by an admissions
committee, are also scaled down to a value between 0 and 1. The nominal attributes are compared through a binary choice. If two nominal dimensional values are the same, (i.e. both students are from Texas), then the difference between these two values is 0. If these two values are not the same, (i.e. one student is from California and one student is from Texas), then the difference between these two values is 1. All of the difference values are summed together and then multiplied by 100/n in order to calculate the distance measure between two specific students. Because we utilize the ordering of the measurements, and order is preserved whether we square this measurement or not we can eliminate one operation by removing the square root from the distance formula. We do use the actual value of the distance but only to compare it to other distance values. The nominal values would be calculated with same binary choice even if we were using the traditional Euclidean distance measure (Gilbert 2005: personal communication).

Chapter 4: Similarity Matrices

The most crucial step in the Applications Quest algorithm is building the similarity matrix. Even though this process is computationally expensive with large data sets, once the matrix is stored, the majority of the work is done. A similarity matrix is a symmetric matrix with zeros on the diagonal. When storing this matrix however, we usually treat it as an upper triangular matrix because values across the diagonal are equal. In order to build this matrix we must compute, “an entry for every comparison and the similarity” between each pair of data points, or in our case each “pair of applications” (Gilbert 2004: 7). This is

\[ nCr = \frac{n!}{(n-r)!r!} \]
comparisons, where \( n \) represents the number of applicants and \( r \) is the number of applicants being compared at one time (Gilbert 2004: 7). Usually we would only be comparing two applications to each other at a time. A similarity matrix for a given set of \( n \) data points would look like the following.

\[
\begin{bmatrix}
0 & d_{12} & d_{13} & \cdots & d_{1n} \\
d_{21} & 0 & d_{23} & \cdots & d_{2n} \\
d_{31} & d_{32} & 0 & \cdots & d_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
d_{n1} & d_{n2} & d_{n3} & \cdots & 0
\end{bmatrix}
\]

\( d \) is a symmetric matrix, as \( d_{ij} = d_{ji} \).

Chapter 5: Clustering Students

Applications Quest relies heavily on a divisive clustering algorithm, even though there are many other clustering choices that could be implemented. Quite generally most researchers use an agglomerative method when conducting hierarchical cluster analysis (Kaufman and Rousseeuw 1990: 253). The agglomerative method is a hierarchical method that treats each data point as a single cluster initially and groups iteratively until all of the data points are contained in one cluster.\(^5\) We must first look for the smallest entry in the similarity matrix and group together the two points that correspond to that entry. Once we have a number of paired data points we must be able to compare clusters to points and clusters to clusters. The manner in which we compute this comparison determines the type of agglomerative clustering we are doing. If we want to know the

\(^5\) In order to reveal anything interesting about the data, one would probably terminate the algorithm before all of the points are grouped into one cluster.
distance between two clusters we can either consider the distance between the two
farthest points in each cluster; the maximum, the distance between the two closest points
in each cluster; the minimum, or the average distance between points in each cluster with
points in the other cluster. These comparison choices are referred to as single linkage,
complete linkage and centroid linkage (Kaufman and Rousseeuw 1990: 224-229).

*The Agglomerative Method: Single Linkage Clustering*

If we have three clusters, Cluster A, Cluster B, and Cluster C, and we wish to
determine the next step in clustering through the single linkage method we would need to
define the distance between Cluster A and Cluster B to be

\[ d(A, B) = \min_{i \in A, j \in B} d(i, j) \]

Once we know the distance between each cluster with each other cluster we can
determine which two clusters to join together, so that if the distance between Cluster A
and Cluster B is smaller than the distance between Cluster A and Cluster C, and also
smaller than the distance between Cluster B and Cluster C we would join all of the
members in Cluster A with all of the members in Cluster B, creating a new cluster so that
we are now left with only two separate groups. Of course if we clustered once more all of
the members in cluster C would be joined with the members in the preceding cluster.
Although this method gives a clear formula for determining distance between objects, it
not always clear or meaningful as to why we choose to define the distance between two
groups by their minimum distance (Kaufman and Rousseeuw 1990).
The Agglomerative Method: Complete Linkage Clustering

If we consider the same three clusters and we opt for complete linkage clustering we would need to define the distance between two clusters to be

\[ d(B, C) = \max_{i \in B, j \in C} d(i, j) \]

So now if the maximum distance between Cluster A and Cluster B is smaller than the maximum distance between Cluster A and Cluster C, and also smaller than the maximum distance between Cluster B and Cluster C, we would join all of the members of Cluster A with all of the members of Cluster B. Again since our example is so small if we clustered once more we would join all of the remaining elements from Cluster C with all of the elements in the preceding cluster, so that our final iteration would be achieved and all of the elements would have been agglomeratively joined together to make on single cluster (Kaufman and Rousseeuw 1990). It seems that choosing the maximum distance would be a better choice because ultimately we would want to cluster together those groups that are the least different from each other, but this is just speculation. Finally there is also the possibility of comparing clusters through an average distance measure, however for the sake of this discussion we need not even consider this method as we would not be able to utilize the similarity matrix and we would have to perform numerous expensive calculations. Juan Gilbert felt that the arbitrary nature of the choice of distance measure between objects made the agglomerative method a poor choice for Applications Quest. Consequently he opted for the divisive method.
Divisive Clustering

Divisive methods are hierarchical as well, meaning that this method, does not construct a single partition with $k$ clusters, but it deals with all values of $k$ in the same run. That is, the partition with $k=1$ (all objects are together in the same cluster) is part of the output, and also the situation with $k=n$ (each object forms a separate cluster with only a single element). In between all values of $k=2,3,\ldots,n-1$ are covered in a kind of gradual transition (Kaufman and Rousseeuw 1990: 44).

Divisive analysis initially treats all data points as members of the same cluster and iteratively divides clusters until ultimately all points belong to their own cluster. There are a number of ways to decide how to decide which cluster to split next. Traditionally the cluster diameter is considered. The cluster diameter is determined by the largest distance between two data points in a given cluster, so that the cluster that contains these two points will be split next. A short example can clearly illustrate this idea. If we are given the following similarity matrix for 5 data points,

\[
\begin{bmatrix}
0 & 4 & 7 & 1.5 & 3 \\
0 & 3.5 & 2 & 11 \\
0 & 4 & 12 \\
0 & 1 \\
0 
\end{bmatrix}
\]

we can see that 12 is the largest value in this matrix. This indicates that data point 5 and data point 3 are the farthest apart. Now we take these two data points to be the centroids and we group all of the remaining data points with either one of the centroids based on which centroid that data point is the closest to. After one iteration we would have the following two clusters.
\[ C_1 = \{d_5, d_4, d_1\} \]
\[ C_2 = \{d_3, d_2\} \]

\( C_1 \) is the cluster that is determined by \( d_5 \) or data point 5 and \( C_2 \) is the cluster that is determined by \( d_3 \). Now if we want to perform one more cluster iteration we must determine the cluster diameters for \( C_1 \) and \( C_2 \). We must consider the distances between each data point with each other data point that belongs to the same cluster. So we see that the distance between \( d_5 \) and \( d_4 \) is 1 and the distance between \( d_5 \) and \( d_1 \) is 3 and the distance between \( d_4 \) and \( d_1 \) is 1.5 so that the cluster diameter for \( C_1 \) is 3, as 3 is the largest distance between two data points contained in \( C_1 \). Similarly we see that the cluster diameter for \( C_2 \) is 3.5. Since 3.5 is larger than 3 our new centroids would be \( d_2 \) and \( d_3 \) as these are the two data points that correspond to the distance 3.5. So once we split \( C_2 \), our new clusters would be following.

\[ C_1 = \{d_5, d_4, d_1\} \]
\[ C_2 = \{d_3\} \]
\[ C_3 = \{d_2\} \]

Now that we have three clusters, (if this was perhaps the desired number of clusters), we can select randomly one data point from each cluster, and we would expect that these three points would be very different from each other or correspond to a large value in the similarity matrix.
If we happened to select \(d1, d2,\) and \(d3,\) as each of these is a member of a different cluster we can observe that the distance between \(d1\) and \(d2\) is 4, and the distance between \(d1\) and \(d3\) is 7, and the distance between \(d2\) and \(d3\) is 3.5. Clearly these are the larger values of the similarity matrix so that our final selection of data points is diverse. Later we will discuss a more meaningful example where we experiment with hypothetical undergraduate applicants to Rice University. Ultimately, the key points in divisive clustering are the hierarchical nature of the algorithm, determining the cluster diameter, and utilizing the similarity matrix.

**Chapter 6: MATLAB Version of Applications Quest**

In simulating this algorithm in MATLAB, I utilized a number of different methods for data storage. Initial data is stored in excel spread sheets, the distance measures are stored in a cell array, and the clusters are also stored inside different cells. The main function, ClusterDivOne, calls on three subfunctions, DMAT, diafind, and clusp.

ClusterDivOne reads in the similarity matrix from the function DMAT and builds a cell array with the size of this matrix. The next step is to split the data points into the first two clusters. In order to know which two points to cluster around we must know the diameter of the current cluster, or the largest value in the similarity matrix, Dmat. The function diafind returns this information and with the function clusp we assign each remaining data point to one of the two points that represented the cluster diameter, (i.e. if S1 and S2 are the two data points that are found to be the farthest apart in the entire data set, then each remaining data point would be grouped with either S1 or S2 in a cell array
through the function clusp). The actual code for ClusterDivOne can be found in the Appendix.

The function DMAT produces the similarity matrix. We call this matrix Dmat. First the data is read into the function from an excel spreadsheet. Then the information is stored in a raw matrix that allows for numerical and nominal values to be stored in the same location. Next we calculate the squared Euclidean distance measure for each pair of students in a nested for loop that accounts for the numerical attributes. We use a separate loop for the nominal attributes using the command strcmp. Once these calculations are complete we add all of the values together and store these values in a symmetric matrix. We fill in the lower triangular portion of the matrix with values identical to those that are across the diagonal, as we did not compute these in our loop in order to decrease computational expense. Once all of these values are filled in we have a symmetric similarity matrix, Dmat. The entire code for DMAT is available in the Appendix.

The next function is diafind. Diafind compares all of the distance measures together within a given cluster and finds the largest one. Once the largest value is located it easy to determine which two students to cluster around, by referencing Dmat. This largest value is called the cluster diameter.

\[
dia_{\text{new}} = \text{Dmat}(C(i),C(j)); \\
\text{if} \ dia_{\text{new}} \geq dia
\]

Diafind returns the largest value and the two students that correspond to that value. The entire code for diafind is available in the Appendix.

The next function is clusp. Clusp splits the current cluster into two new clusters based on the two students that diafind returns. Clusp compares the distance between each
student with each of the students that diafind returns in order to determine which student
the other students are the closest to.

\[
d1 = Dmat(sn,s1);
d2 = Dmat(sn,s2);
\]

if \(d1<\)d2
\[C1 = [C1,sn];\]
else
\[C2 = [C2,sn];\]
end

If sn is closer to s1 then sn will be grouped in C1. The entire code for the clusp function
is also available at the Appendix. Ultimately once we have implemented the major steps
of Applications Quest our algorithm is implemented with a driver entitled, FinalStuff,
which calls on three additional functions that do a random selection (Selection.m),
compute a difference index (DifInd.m), and computes the statistics of the group of
applicants that are admitted (attstat.m). The codes for all of these functions are available
in full in the Appendix.

**Chapter 7: Revisiting the Five Point Example**

In this example we have five students and five attributes. Two of the five
attributes are nominal entries. This example will demonstrate how Applications Quest
works from beginning to end as well as give some insight into how the number of
nominal and numerical attributes affects the outcome of the clustering.

*Data*
<table>
<thead>
<tr>
<th>GPA</th>
<th>SAT</th>
<th>Essay</th>
<th>Race</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>3.5</td>
<td>1450</td>
<td>5</td>
<td>W</td>
</tr>
<tr>
<td>S2</td>
<td>3.8</td>
<td>1420</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>S3</td>
<td>2.4</td>
<td>1250</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>S4</td>
<td>4.0</td>
<td>1200</td>
<td>2</td>
<td>H</td>
</tr>
<tr>
<td>S5</td>
<td>2.4</td>
<td>1310</td>
<td>4</td>
<td>H</td>
</tr>
</tbody>
</table>

$Dmat$

\[
\begin{bmatrix}
0 & 20.9195 & 42.6250 & 42.6008 & 41.8656 \\
0 & 22.6758 & 40.6281 & 42.7445 \\
0 & 43.4195 & 20.2281 \\
0 & 24.0945 \\
0 &
\end{bmatrix}
\]

$Clusters$

If we decide to select two students from this group of five we would have to generate two clusters. We see that S3 and S4 are the farthest apart so that we will create two clusters based on the proximity of each student with each of these students. We see that S1 is closer to S4 and that S2 and S5 are closer to S3 so that our clusters are

$C1 = \{S2, S3, S5\}$

$C2 = \{S1, S4\}$

S3 and S4 are two students with similar SAT scores, significantly different GPA’s and different races and different states of residence. It seems that the race and the state attributes, as nominal values, significantly affect the outcome of the clusters. Clearly when we have only a few attributes if a large portion of those attributes are nominal those attributes will be heavily weighted. If we make a random selection from each cluster our
two students might be S2 and S4. We see that these two students vary mainly in the nominal values as their GPA’s and SAT scores are in a similar range. We must consider a larger example in order see how much impact different choices for nominal values might actually have on the final admissions class. Creating hypothetical applicants for an incoming freshman class at Rice University serves as an appropriate example to consider when exploring issues such as the affects of nominal attributes along with many other issues that will come to the fore in the following sections.

Chapter 8: An Example with Rice Applicants

In this example we will consider 800 hypothetical Rice University Applicants. Approximately 8000 students apply to Rice each year and around 1900 students will be accepted. We will consider a scenario where we have 800 students and we wish to admit 190. The demographics in this data set mimic the data in the public document, Rice University: A Factual Sketch 2005-2006. The data set that we use is randomly generated with respect to the following percentages. Grades are randomly generated for values that fall between 3.0 and 4.0 as it is understood that clustering algorithms such as this one are utilized after a particular cutoff point. We assume here that only an insignificant amount of students with a grade point average below 3.0 apply. The majority of students that apply to Rice have an SAT verbal score higher than 750 as the demographics indicate that 44% of applicants score between 800-750, 25% of applicants score between 750-700, 18% of applicants score between 700-650, 9% of applicants score between 650-600, 3% of applicants score between 600-550, .4% of applicants score between 550-500 and .6%

---

6 It is important to note that most of these percentages are for the demographics of students that matriculated into Rice, but we are using these percentages to guide the demographic for not only students that matriculated, but also students that applied and may not have chosen to go to Rice.
of the applicants score below a 500 for the SAT verbal attribute. The demographics show a similar distribution for the SAT math attribute as 47% percent of Rice applicants score between 800-750, 29% score between 750-700, 15% score between 700-650, 6% score between 650-600, 2% score between 600-550, .8% score between 550-500 and .2% of Rice applicants score below a 500 on the SAT math attribute.

Further, class rank is recorded demographically in ranges. We see that 13% of the applicants are #1 in their class, 6% are #2 in their class, 40% are in the top 5% of their class, 4% are in the top 10% of their class, 2% are in the top 20% of their class, .4% are in the top 30% of their class, .5% are in the top 40% of their class, .28% are in the bottom 50% of their class, and 33.82% are from schools that do not calculate a class rank. Academic rigor, student presentation, and recommendations are randomly generated numbers that are not restricted to any percentage values because these demographics were not available.

The demographics for personal qualities such as gender, race, residence, and major interest are consistent with following percentages. Rice applicants are typically 53% male and 47% female. 54% of the applicants classify themselves as White, 16% as Asian, 12% as Hispanic, 7% as Black, 7% as Other, and 3% as International. The majority of the applicants, and consequently the student population are from Texas, as 48% are from Texas, 9% are from the West Coast, 6% are from the Midwest, 4% are from the Southwest, 18% are from the South, and 15% are from the East Coast. Some of the students’ major academic interest areas overlap but a rough estimate of the

---

7 It should be taken into consideration that these percentages are not typical or consistent from year to year because the universities regional preferences change. For example selecting students from the East Coast was a priority for the 2005-2006 admissions season (Siler 2006: personal communication).
demographics for academic interest are the following. The majority of the students indicate that their major academic interest will be in the Social Sciences as 44% of applicants declare a Social Sciences discipline as their initial major. 22% major in Natural Science, 24% major in Engineering, 6% major in Architecture, and 4% major in Music.

The attributes used in this data set are loosely consistent with the five major categories that the Rice University admissions committee uses to evaluate applicants. These five categories are “Academic Rigor, Physical Grades, Recommendations, Student Presentation, and Personal Qualities” (Siler 2006: personal communication). In order to create an accurate representation of these qualities I have broken down each of the five categories into several separate attributes. Under the grades category students are evaluated by their grade point average, SAT verbal score, SAT math score, and their class rank. The grade point average is a numerical value with a maximum value of 4. The SAT verbal and SAT math scores are recorded in ranges and the lower bound of the range is entered into the distance formula. If a student receives a 790 on the SAT math then 750 is used in the distance formula because 790 falls between 800 and 750 so that 750 is the maximum for these attributes. The class rank is also divided into ranges and the entry is used as a nominal value. If a student is in the top five percent of their class then “TopFivePercent” is entered into the distance formula so that if one student has an entry of “TopFivePercent” and another student has an entry of “Valedictorian” then for this attribute these students would receive a 1 in their distance formula for the class rank attribute.

---

8 Many of the Social Science and Natural Science majors double major in areas such as Engineering and Humanities.
The Academic rigor category is a numerical value scored between 1 and 5 and 5 is the maximum value. The Student Presentation category is also a numerical value scored between 1 and 10 and 10 is the maximum value. Teacher recommendations operate on the same scale. These scores are assumed to be assigned by admissions personnel based on increasing improvement so that a student that challenged themselves the maximum amount in academics (i.e. Advanced Placement courses) would receive a 5 for this attribute. All of the personal qualities are recorded and used as nominal values. There are four attributes under personal qualities: gender, race, residence, and major academic interest. The gender attribute is a choice between male and female. The race attribute is a choice between Asian, Black, Hispanic, International, Other, and White. The residence category is a choice between Texas resident, Southwestern resident, Southern resident, Midwestern resident, East Coast resident, and West Coast resident. The major interest category is a choice between, Natural Science, Social Science, Engineering, Architecture, and Music. These attributes combine to create eleven different attributes which we use to evaluate the distance between 800 different students.

Once we ran Applications Quest with these 800 applicants 190 admitted students were returned. We ran the data set twice and received two different sets of 190 students that would have been admitted if Rice University implemented Applications Quest. We consider one of these admissions classes, Admissions Class A, to explore very specific selection issues and the second admissions class, Admissions Class B, to explore larger demographic issues. We will discuss Class B in a later section. Ultimately upon entering
the hypothetical 800 students we get the following clusters\(^9\) and students for Admissions Class A.

**Clusters**

\[ C_1 = \{2, 95, 355\}, C_2 = \{531, 683, 778\}, C_3 = \{27, 664, 680, 770\}, C_4 = \{404, 412, 772\}, C_5 = \{604\}, C_6 = \{36, 41, 109, 222, 238, 304, 328, 403, 463, 669, 771, 795\}, C_7 = \{261, 796\}, C_8 = \{457, 602, 736, 780\}, C_9 = \{53, 426\}, C_{10} = \{9, 259, 415, 419, 472, 711\}, C_{11} = \{535\}, C_{12} = \{97, 142, 256, 299, 339, 340, 657\}, \ldots, C_{50} = \{68, 542, 760, 765, 794\}, \ldots, C_{75} = \{4, 12, 31, 33, 107, 186, 209, 246, 247, 383, 469, 481, 487, 533, 732\}, \ldots, C_{100} = \{46, 161, 226\}, \ldots, C_{125} = \{550\}, \ldots, C_{150} = \{81, 392, 445\}, \ldots, C_{190} = \{54\}. \]

**Admissions Class\(^{10}\)**

Admissions Class A:


Admissions Class B:


---

\(^9\) The entire set of clusters is available in the Appendix. We only show a few here for the sake of understanding the final admissions class.

\(^{10}\) These values are selected at random, where one value is selected from each cluster.
Admissions Class A had a Difference Index of 36.71. Observe that several of the students selected in Admissions Class A were ultimately clustered into a group of their own so that upon random selection they were automatically chosen for the admissions class. For example S305, S336, S583, and S424 are some students in this situation. S305 is a valedictorian, from the South, interested in Natural Science, classified as Other, Male with a 3.68 grade point average. His SAT score is 1400 and he received and 3, 1, and a 10 for academic rigor, student presentation, and recommendations respectively. S336, S583, 424 have the following attributes:

S336: TopTenPercent, Texas, Engineering, White, Female, 3.34, 1300, 2, 3, 6.
S583: TopTenPercent, Southwest, Music, White, Female, 3.91, 1450, 4, 3, 2.
424: TopTenPercent, Texas, Social Science, Black, Female, 3.61, 1450, 2, 1, 3.

We will later discuss further some of the drawbacks for this type of selection as it is not clear that these automatic choices above would have been the ideal choices for the actual Rice admissions class. Consider for example that S424 has three very low numerical scores. For academic rigor she received a 2, and for student presentation she received a 1, and for recommendations she received a 3. It seems as though poor performances on many numerical categories enabled her to be so different that she was a more diverse choice.

If we consider the entries in the similarity matrix for a few of the selected students form Admissions Class A we should observe that these measurements are some of the larger values in the matrix. The distance between S13 and S51, two of the selected
students is 41.1330. Similarly the distance between S56 and S68 is 37.7877, the distance between S121 and S410 is 44.7130, and the distance between S434 and S437 is 28.3907. These numbers are considerably larger than the entries we would find if we consider the distance between two students from the same cluster. Consider the distance between S13 and S65. These students are both from C20 so that their distance measure is 12.1576. Clearly this value is smaller than all of the other distances we considered when observing the measurements for students from the admissions class.

Now we must consider a few of the more interesting selections that were made for Admissions Class A.\textsuperscript{11} Clustering algorithms for university admissions can vary in the way that the final selection is made. Here we have randomly selected one student from each cluster. We must now consider the implications of this type of selection process. It is understood with this algorithm as well as Applications Quest that these clustered students are admissions choices that survived beyond a certain cutoff point. If a student has a 1 for example on academic rigor, as S424 has, this does not necessarily mean that this student is one of the least challenged students who applied, but of the students that were retained in a certain threshold she is at the bottom. It is important to note that since our data is randomly generated in Microsoft Excel and our percentages that we use to restrict these choices are guided by the typical Rice freshman class demographic that our final selection percentage-wise looks correct. So then the more important thing to consider here is specific students and how choosing them is consistent or conflicting with certain admissions ideologies.\textsuperscript{12} There are some very specific examples of students that

\textsuperscript{11} Here we only consider analysis for a fixed data sheet of students with respect to the same eleven attributes.

\textsuperscript{12} When I use the term ideology here I am referring to the different perspectives on the admissions committees.
were not chosen with our clustering algorithm, but if perhaps viewed by a pro-
Affirmative Action committee member for example may have been selected. Clearly any
sort of recognition for race or gender would be a subjective acknowledgement of the
students’ personal challenges that may or may not come out in a student essay. We do not
directly consider personal challenges in our data because this is not an easily quantifiable
category, but I reference race and gender for the sake of discussion in this topic.

S190 is a male, Hispanic student from the South, who is interested in Engineering
and had a 1450 on his SAT, a 3.53 GPA, was valedictorian of his class and had high
scores on student presentation and recommendations. S190 was not chosen for
admissions with our clustering algorithm. Clearly this student is a competitive
underrepresented minority who perhaps would have been chosen with a more subjective
selection process.

S2 is a female, Black student from the East Coast, who is interested in Natural
Science and had a 950\textsuperscript{13} on her SAT, 3.49 GPA, was valedictorian of her class, and had
high scores for academic rigor and student presentation. S2 was not chosen for admission.
We know that Rice has a very subjective approach to admissions in general and maybe
S2’s SAT score would have eliminated her from admissions, but her other qualities seem
to make up for this academic drawback.\textsuperscript{14}

S263 is a female, White student from Texas, who is interested in Engineering and
had a 1500 on her SAT, a 3.46 GPA, and received high scores on student presentation
and recommendations. S263 was not chosen for admission. It was also clear that Rice

---

\textsuperscript{13} It was very clear to me upon interviewing an admissions member that often times SAT scores were
evaluated differently for different students, especially a student that was strong in other areas.

\textsuperscript{14} When we mention other qualities this does not only reference race and gender, but the certain regions of
residence have become preferable so that her being form the East Coast is also seen as a “plus factor.”
values increasing the number of women in Computer Science and Engineering so that S263 might have been an excellent admit for this type of diversity.

Now let us consider some of the students that were chosen instead of the previously mentioned students. In S2’s cluster we see that we admitted S95, a Black female student from the East Coast who had a much higher SAT score of 1350 and was in the top 5 percent of her graduating class so that perhaps our algorithm chose fairly, but it is important to observe that since we are using random selection our algorithm could have chosen S2 instead of S95 and clearly we see here that S95 is a more successful student.

When we consider the student that was chosen instead of S190 we see an interesting predicament. Since we can only choose one student from each cluster we run the risk of eliminating some very good students and then selecting some mediocre students from another cluster. S698 is a Black male, who is interested in Engineering, he is from the Midwest, had a 1400 on his SAT, was valedictorian and had a 3.72 GPA. Clearly S698 is a good choice, but so is S190. Both of these students are successful underrepresented minorities who would have most likely been chosen under the traditional Rice admissions, but we had to choose only one of these students so that we sacrifice improvement with underrepresentation for a form of diversity that may not be consistent with Rice’s or any university’s diversity goals.

We certainly have a few lower scoring students that were selected as these students were clustered together and we had to choose one of them from that cluster. S761 is a female, white student from the Midwest, interested in Natural Science with a 3.18 GPA and an SAT score of 1200. Clearly these are not bad scores, but in relation to
the scores of S190 it seems unfortunate that random selection and a lack of subjectivity in the process dispose of students like S190 and guarantee the admittance of mediocre students like S761. There are many more interesting outcomes that can be viewed in the Appendix. Now we should consider a few of the computational and algorithmic issue that are made clear in this example.

As mentioned before the way in which we choose students after clustering could be more effective if we evaluated which clusters are actually suitable or preferable so that we might not consider the students that are in a cluster like the cluster S761 was grouped in. The SAT and GPA scores of the other students in this cluster were in the same range. Similarly if we have many good students grouped into one cluster it may be beneficial to choose more than one student from this cluster. These are a few issues about selection that could be altered to ensure not only diversity but an academically appropriate selection.

Another issue that is indirectly addressed in this example is how the nominal values affect the outcome. It is clear that these attributes are weighted when we compute the distance because we normalize everything else and we select students with scores that are well above the minimum possible score so that no two people would ever attain a 1 for a numerical attribute difference. For example no student is ever going to have a 0 GPA so that when we compute GPA by subtracting their scores and dividing by 4, the maximum value, the value would never be 1. However, when we compare two students with different nominal values the only option is to add a 1 or a 0 to the distance measure. This arrangement allows for questions about how much this weighting of nominal values will affect the outcome. When we are dealing with very few students and attributes and
the majority of those attributes are nominal it is clear that the weighting of those values greatly skews the outcome, but when dealing with 800 students and only 45% of the attributes are nominal it is not so clear that this weighting is significant. The interesting thing here is that not all universities will have the appropriate amount of numerical attributes to offset the nominal attributes so that we are not always safe from the effects of this weighting. Why is weighting significant? For two reasons: The Supreme Court says it is illegal to weight racial components of the application and programs like Applications Quest are being advertised as a fair approach that does not weight race. It is not clear that this claim can be made because it seems to be dependent on the specific data set that is chosen.

Furthermore, we can now consider broader issues such as the overall demographic of a typical Admissions Class chosen with Applications Quest. Recall Admissions Class B:


We will not only consider the overall demographics of Admissions Class B but also consider a comparison of Class A and Class B. Is Applications Quest consistent? How close are the overall demographics from an Applications Quest selection to the Rice Admission’s committee selection? We can answer these questions through evaluation of Admissions Class B.
Admissions Class B had a difference index (DI) of 36.71 while Admissions Class A has a DI of 37.09, so both classes are relatively equally diverse. Using attstat.m we can explore the demographics of Admissions Class B. Gender: 50% male and 50% female; Race: 42.11% White, 8.95% Black, 14.21% Hispanic, 16.84% Asian, 2.63% Native American, 5.26% International, and 10% other; Residence: 40.53% Texas, 11.05% West Coast, 24.74% South, 15.79% East Coast; Major Interest: 38.95% Social Science, 21.58% Natural Science, 23.68% Engineering, 6.84% Architecture, 8.95% Music; SAT Verbal: 40.53% 800-750, 23.68% 740-700, 18.95% 690-650, 8.95% 640-600, 6.32% 590-550, 1.05% 540-500, .53% 490-450; SAT Math: 44.21% 800-750, 28.42%740-700, 16.32% 690-650, 7.37% 640-600, 1.58% 590-550, 1.58% 540-500, .53% 490-450; Grade Point Average: 23.16% 4.0-3.8, 17.37% 3.79-3.6, 24.21% 3.59-3.4, 18.42% 3.39-3.2, 16.84% 3.19-3.0; Class Rank: 7.37% Salutatorian, 34.74% Top 10 percent, 1.58% Top 20 percent, 1.58% Top 50 percent, 2.63% Bottom half, 27.89% Unranked. All of these categories can be compared to the actual Rice admissions data for 2005. We see that most of the percentages are close to the actual except for the low numbers in certain residential areas and variances in class rank. The real issues come to the fore when we consider individual applicants as we did with Admissions Class A. We do however observe that Applications Quest is consistent with the amount of diversity that it delivers in an admissions class, as the DI's are relatively close, and given certain restrictions initially with the applicant pool, AQ seems to return percentages that mirror what was chosen by the admissions committee for the 2005-2006 school year.\textsuperscript{15} It is also important to note the percentage of Black students increased by 2% and the percentage of Hispanic students increased 2%. Clearly race does play a role in valuing difference.

\textsuperscript{15} This is excluding Spring 2006 transfers.
Academic Rigor, Recommendations, and Student Presentation\textsuperscript{16} are all randomly generated percentages that are not based on any real Rice University data. For this reason we do not discuss these in great detail. Ultimately what have we learned from simulating programs like Applications Quest? Computational Admissions could work as a preliminary process perhaps for a large institution, but for a small intimate environment like Rice we eliminate applicants that may be acceptable. For issues of underrepresentation we should note that with Admissions Class B we lowered our percentage of Black students. We did not admit the most academically talented minority students and we traded certain minority students for students who attained difference by way of an average academic performance. Applications Quest manipulates the percentages in such a way that initially we see improvement, but the people that are eliminated and the small changed in the percentages could prove to be detrimental to the role of underrepresentation.

\textbf{Chapter 9: Social Concerns}

After the Michigan cases it became clear that all major institutions needed to define what sort of holistic policies they were going to use when dealing with diversity in admissions. Most people agree that educational diversity is a major benefit for all students and should be a goal of the university. Rice University, like many other selective universities, clearly values educational diversity and consequently conducts a holistic evaluation process when selecting students for the incoming class. In speaking with a

\textsuperscript{16} Academic Rigor: 16.32\% 5, 22.11\% 4, 14.74\% 3, 22.63\% 2, 24.21\% 1; Recommendations: 6.32\% 10, 49.74\% 9-5, 44.21\% 5-1; Student Presentation: 11.58\% 10, 55.79\% 9-5, 32.63\% 5-1.
director of admissions it became clear that the preferred process of admissions for Rice is very subjective and, perhaps, could not be adequately represented by a computational evaluation like Applications Quest.

Every year approximately 8,000 people apply to the Rice University undergraduate program and approximately 1,900 people are ultimately accepted. Since there is a yield of 35 to 40 percent, the incoming class is usually around 725 students in an incoming class. The admissions process at Rice, in brief, consists of consideration of five major categories of evaluation, "academic rigor, physical grades, recommendations, student presentation, and personal qualities" (Siler 2006: personal communication).

Academic rigor consists of the different types of academic challenges the student took on, such as advanced placement courses or academic challenge teams. Physical grades are of course the actual grades on a transcript or the actual performance in each of those classes or academic challenges. Recommendations are just the high school teacher evaluations and the counselor evaluations. Student presentation is made up of a number of factors, such as writing ability and organization of the application. Personal Qualities, also called "special interests," are qualities such as personal challenges, hobbies, and significant accomplishments such as knowing another language or being the first generation in a family to go to college. The personal qualities category is where diversity is established (Siler 2006: personal communication).

It is clear that the Rice admissions process is subjective and holistic. It seems that with this type of process diversity could be easily accomplished, depending on what we take diversity to mean. Some people think of diversity as regional or national difference, or even difference in academic interests, (i.e. special consideration for a student who
applies as a linguistics major for example), while others think of diversity as a direct response to the underrepresentation of certain groups of people in higher education. This is where the controversy with diversity is located. Every one agrees that diversity is important, but there are a variety of perspectives on what diversity means and how we should go about achieving diversity.

In speaking with one admissions committee member at Rice it became clear that she felt that Applications Quest would be a difficult program to incorporate into the current method of selection at Rice. Upon suggesting that using Applications Quest be only one step in the process, she made it clear that doing a subjective stage of admissions and an objective computational stage was unnecessary and perhaps damaging. If an individual is eliminated in the objective portion of selection based on a computational analysis of how this individual will bring diversity to the university, it is not clear that that type of evaluation would be holistic or even fitting with the admission committee's definition of diversity. All individuals do not bring the same type of diversity to the university and it seems that the ability to holistically categorize what type of educational diversity will be achieved through a certain individual is a highly subjective process. Clustering Algorithms for university admissions seem to equate all types of difference whereas the Rice admissions process seeks different types of diversity from within different individuals (Siler 2006: personal communication).

Chapter 10: Conclusions

There are many interesting issues that arise with the possibility of implementing Applications Quest in many different selective universities. The major benefit of this
program is that perhaps some sort of consensus could be reached among individuals who
have opposing beliefs with respect to different affirmative action policies. This paper
clearly illustrates some of the computational and social benefits and drawbacks of using a
program such as Applications Quest, but ultimately everyone would agree that it is in fact
an objective and consequently a fair process as human evaluation is minimized. Perhaps
if human evaluation was not part of the environmental strain that initially compromises
who applies to schools, who can be retained in schools, and who can excel in schools,
eliminating it would be a good idea. However we all know these social constraints play a
major role in underrepresentation. Ultimately eliminating a human analysis of these
social constraints and reducing them to an objective system is not holistic. Perhaps if we
can eliminate the structural forms of bias that skew the representation of certain groups in
higher education we can resort to a completely computational means of selecting students
for university admissions, but until we reach this point it is clear that programs such as
Application Quest will only avoid the issues that the underrepresentation of certain
groups implicate. Although Applications Quest is a fair and unbiased program, there is no
realistic data set that we can enter into Applications Quest that can maintain that fairness.
Programs like this one will be constrained by the current social structure and until we
understand certain structural problems we will ultimately implement Applications Quest
in vain.
Bibliography

Abrams, Floyd
2003 Brief of Amici Curiae Columbia University, Cornell University, Georgetown University, Rice University, and Vanderbilt University in Support of Respondents. New York: New York.

Alger, Jonathon

Coleman, Arhur L. and Scott R. Palmer

Franken, Bob

Garcia, Carlos

Gilbert, Juan


Kauffman, Leonard and Peter J. Rousseeuw

Rice University Office of Admissions

Siler, Tamara

Tapia, Richard
APPENDIX A

CODE

ClusterDivOne

Dmat=DMAT;

Ns = size(Dmat,1);
clear C
C{1} = linspace(1,Ns,Ns);

culim = 190;

i = 1;
count = 1;

while count < culim

    CurrNumClus = length(C);

    for j = 1:CurrNumClus

        [dia(j),s1(j),s2(j)] = diafind(C{j},Dmat);

    end

dia
[val,i] = max(dia)
[C{i},C{end+1}] = clusp(C{i},s1(i),s2(i),Dmat);

    count = count + 1;

end

C{:}

DMAT

function [Dmat]=DMAT

[num, txt, raw] = xlsread('RiceApplicantsFixed.xls'); %Applications
raw;

data = raw(2:end,2:end); %getting rid of the first row of raw because it is the title row.
[Nst,m]=size(data);
numid = [];
nomid = [];
for i = 1:m
    if ischar(raw{2,i})
nomid = [nomid,i];
else
    numid = [numid,i];
end
end

maxval = [4,750,750,5,10,10,0,0,0,0,0];
numid = [1:6];
nomid = [7:11];

for i = 1:Nst
    for j = i+1:Nst
        Na = 0;
        term = [];
        for k = numid
            if data{i,k} == 0 & data{j,k} == 0
                term(k) = ((data{i,k} - data{j,k})/maxval(k))^2;
                Na = Na + 1;
            end
        end
        for p = nomid
            T1 = isnan(data{i,p});
            T2 = isnan(data{j,p});
            if T1(1) & T2(1)
                term(p) = 1-strcmp(data{i,p},data{j,p});
                Na = Na + 1;
            end
        end
        d(i,j) = (sum(term)/Na)*100;
    end
end

Dmat=d;

for i=1:Nst
    for j=i+1:Nst
        Dmat(j,i) = Dmat(i,j);
    end
end

Dmat

DIAFIND

function [dia,s1,s2] = diafind(C,D)
Dmat = DMAT;
dia = 0;
n = length(C);

for i = 1:n
   for j = 1:n
      dia_new = Dmat(C(i),C(j));
      if dia_new >= dia
         dia = dia_new;
         s1 = C(i);
         s2 = C(j);
      end
   end
end

CLUSP

function [C1,C2] = clusp(C,s1,s2,D)

C1 = [];
C2 = [];
n = length(C);

for i = 1:n
   sn = C(i);
   d1 = Dmat(sn,s1);
   d2 = Dmat(sn,s2);
   if d1<d2
      C1 = [C1,sn];
   else
      C2 = [C2,sn];
   end
end

FinalStuff (Driver)

ClusterDivOne

s=Selection(C);

D1=DifInd(s,Dmat)

[num, txt, raw] = xlsread('RiceApplicantsFixed.xls');

raw = raw(:,2:end);
statmat = attstat(s,raw)

DifInd

function [D] = DifInd(s,Dmat)

D = 0;
m = length(s);
for i = 1:m
  for j = i+1:m
    D = D + Dmat(s(i),s(j));
  end
end

D = D/m;
D = D*(2/(m-1))

Selection

function [s] = Selection(C)

for i = 1:length(C)
  n = length(C{i});
  ind = ceil(n*rand);
  s(i) = C{i}(ind);
end
s

attstat

function statmat = attstat(s,raw)

Amat = raw(s+1,:); % names are in 1st row so student 1 is in row 2
for j = 1:size(raw,2)
  att = raw{1,j};
  if strcmp(att,'Personal Qualities-Gender')
    types = {'Male','Female'};
    statmat{j} = Nomfun(types,Amat,j)
  elseif strcmp(att,'Grades-GPA')
    types = [3.8,3.6,3.4,3.2,3.0]
    statmat{j} = Numfun(types,Amat,j)
  end
end
else if strcmp(att, 'Academic Rigor-(1-5)')
    types = [5,4,3,2,1]
    statmat(j) = Numfun(types,Amat,j)

else if strcmp(att, 'Recommendations-(1-10)')
    types = [10,5,1]
    statmat(j) = Numfun(types,Amat,j)

else if strcmp(att, 'Student Presentation-(1-10)')
    types = [10,5,1]
    statmat(j) = Numfun(types,Amat,j)

else if strcmp(att, 'Personal Qualities-Race')
    types = ['white','black','hispanic','asian','native american','international','other']
    statmat(j) = Nomfun(types,Amat,j)

else if strcmp(att, 'Personal Qualities-Residence')
    types = ['Texas','West Coast','Midwest','Southeast','South','East Coast']
    statmat(j) = Nomfun(types,Amat,j)

else if strcmp(att, 'Grades-SAT Verbal')
    types = [750,700,650,600,550,500,450]
    statmat(j) = Numfun(types,Amat,j)

else if strcmp(att, 'Personal Qualities-Major Interest')
    types = ['Social Science','Natural Science','Engineering','Architecture','Music']
    statmat(j) = Numfun(types,Amat,j)

else if strcmp(att, 'Grades-Class Rank')
    types = ['Valedictorian','Salutatorian','TopFivePercent','TopTwentyPercent','TopFiftyPercent','Bottom FiftyPercent','Unranked']
    statmat(j) = Nomfun(types,Amat,j)

else if strcmp(att, 'Grades-SAT Math')
    types = [750,700,650,600,550,500,450]
    statmat(j) = Numfun(types,Amat,j)
end

function statvec = Nomfun(types,Amat,attcol)
    for i = 1:length(types)
        vals = strmatch(types(i),Amat(:,attcol));
        statvec(i) = length(vals)/size(Amat,1);
    end
end

function statvec = Numfun(types,Amat,attcol)
    numvec = cell2mat(Amat(:,attcol));
    for i = 1:length(types)
if i==1
    vals = find(types(i)<=numvec);
else
    vals = find(types(i)<=numvec & types(i-1)>numvec);
end
statvec(i) = length(vals)/size(Amat,1);
end
APPENDIX B

Full Set of Clusters from the Rice Example

C1 = {2,95,355}
C2 = {531,683,778}
C3 = {27,664,680}
C4 = {404,412,772}
C5 = {604}
C6 = {36,41,109,222,238,304,328,403,463,669,771,795}
C7 = {261,769}
C8 = {457,602,736,780}
C9 = {53,426}
C10 = {9,259,415,419,472,711}
C11 = {555}
C12 = {97,142,256,299,339,340,657}
C13 = {686}
C14 = {17,49,126,149,224,272,280,312,532,603,758,799}
C15 = {190,698,781}
C16 = {476,619,655,720}
C17 = {229,313,440,501,615,742,767}
C18 = {58,232,479,516,593,648}
C19 = {139,283,656,666}
C20 = {13,65,286,411,491,526,596,607,754}
C21 = {277,461,544,641}
C22 = {25,448,756}
C23 = {167,221,331,468}
C24 = {305}
C25 = {640,718}
C26 = {347,701}
C27 = {125,178,452,768}
C28 = {163,176,210,332,370,738,745}
C29 = {110,175,273,451,661,709}
C30 = {402,442,467,647}
C31 = {179,240,257,639,789}
C32 = {122,228,250,338,654}
C33 = {34,217,227,290,696,706}
C34 = {702,744}
C35 = {266,300,311,327,360,362,413,430,614,634,675,676,719,753,783,800}
C36 = {292,506}
C37 = {143,626}
C38 = {690}
C39 = {611,757}
C40 = {521}
C41 = {177}
C42 = {89}
C43 = {522}
C44 = {213,409,572}
C45 = {141,470,502,590,646}
C46 = {460,710}
C47 = {72,294,528,571,613,644}
C48 = {367,520,589}
C49 = {366,600}
C50 = {68,542,760,765,794}
C51 = {134,519}
C108 = \{90,555\}
C109 = \{333,606,715,751\}
C110 = \{445,55,56,201,244,262,350,645,699\}
C111 = \{60,242,423,586\}
C112 = \{106,151,341\}
C113 = \{255,291,371,624\}
C114 = \{30,317,345\}
C115 = \{76,82,94,389,504,663,730\}
C116 = \{301,410,422,548,708,793\}
C117 = \{543,731\}
C118 = \{20,86,203,241,348,474,500,545,610,734\}
C119 = \{28,137,231,356,494\}
C120 = \{98,397,790\}
C121 = \{601,605,766\}
C122 = \{361,524\}
C123 = \{375,717\}
C124 = \{121,643\}
C125 = \{550\}
C126 = \{140,156,198\}
C127 = \{5,346,480,724,773\}
C128 = \{337\}
C129 = \{212,484,556,573\}
C130 = \{21,50,220,282,623,688\}
C131 = \{152\}
C132 = \{38,51,128,235,439,444,490,594,722,762,775\}
C133 = \{204,459\}
C134 = \{79,132,147,237,488,554,635,769,788\}
C136 = \{1,352,595,649,692,695\}
C137 = \{258,308,344,507,549,574,714\}
C138 = \{8,18,180,145,195,351,365,449,782\}
C139 = \{14,155,233,359\}
C140 = \{223,349\}
C141 = \{47\}
C142 = \{159,274,281,284,303,539,560,569,637,667,774\}
C143 = \{84\}
C144 = \{425,446,537\}
C145 = \{103,354,386,432,536,559\}
C146 = \{517,693\}
C147 = \{64,129\}
C148 = \{183,456,485\}
C149 = \{24,245,434,518,530\}
C150 = \{81,392,445\}
C151 = \{182,566,578\}
C152 = \{111,307,433,579,678,761\}
C153 = \{100,243,630\}
C154 = \{424\}
C155 = \{302,707\}
C156 = \{193,437,562,662\}
C157 = \{118,253,512\}
C158 = \{202,218,271\}
C159 = \{23,62,93,148,158,160,174,199,216,326,528,353,314,394,441,511,570,584,776\}
C160 = \{40,251,343,384,588,759\}
C161 = \{3,22,45,170,185\}
C162 = \{144,180,492\}
C163 = \{83, 369, 514, 587, 752\}
C164 = \{87, 146\}
C165 = \{295\}
C166 = \{39, 69, 493, 498, 510, 580, 728, 791, 798\}
C167 = \{368, 691\}
C168 = \{335, 725\}
C169 = \{131, 428\}
C170 = \{486, 629, 779\}
C171 = \{123, 192, 214, 435, 608, 617, 716\}
C172 = \{420, 477\}
C173 = \{184, 194, 552, 673\}
C174 = \{150\}
C175 = \{319, 527\}
C176 = \{66\}
C177 = \{334, 704\}
C178 = \{381\}
C179 = \{104, 188, 616\}
C180 = \{321\}
C181 = \{73, 153, 187, 378, 723\}
C182 = \{398\}
C183 = \{363, 405, 609\}
C184 = \{120, 181, 509, 525\}
C185 = \{318, 329, 729\}
C186 = \{115, 119\}
C187 = \{78, 473\}
C188 = \{162, 558\}
C189 = \{465\}
C190 = \{54\}