Abstract

This paper discusses various approaches to computerize division three tennis rankings and evaluate their accuracy based on the official rankings that were released this year. From this analysis I hope to establish a reliable method for ranking division three teams, and to expedite the time it takes for these rankings to be released. Currently all division three rankings are decided upon by a committee which is full of coaches with biased opinions. I used a metric system of assigning points for wins and introduced a coefficient for weighting the strength of each region in order to balance out some imperfections in the model. The rankings I got were generally accurate but it probably should not be used in practice. It could however be used jointly with a committee which would modify my results based on other criteria.
Introduction:

For the past couple of years the Intercollegiate Tennis Association (ITA) has been thinking about computerizing division three tennis rankings. Currently a committee gets together once a year and makes the rankings. This takes a long time and the committee is made up of coaches whose obvious priority is to boost their own team's standing. There are various problems that arise with any attempts to computerize rankings, however. In division one, for instance there is a lot of national play and almost all ranked teams from one region play all the ranked teams from all other regions. This makes computerizing rankings simple because one can assign some specific number of points for a win and at the end simply add up each team's points to come up with a fairly good ranking system. In division three, however; there is not much national play and many teams do not play an equal number of matches, hence a method is needed to somehow interlace rankings from each region, to come up with an overall national ranking. In this paper I will discuss a couple of attempted approaches and how they compare to the rankings released earlier this year. There has been no previous work in this area, so most of my methods are based on trial and error methods. I hope to come up with some system that will expedite the ranking formulation process, and in effect allow for them to come out more then once a year.

Modeling Approach:

The model used to rank teams in division one is actually a metric system which looks at a match
result and gives the winning teams a certain number of points based on the rank of the loosing team. So for instance if team A beats team B that was ranked 2 in the nation team A would get 30 points. If team B was instead ranked 10 then team A may only get 20 points.

I made a couple of simplifying assumptions for this model to save myself some time. First I could not possibly enter information for all division three teams, so I only took the top 13 ranked schools. Even then each school plays around 20 to 30 matches, and entering information regarding each of those matches would take a very long time so I entered a random collection of results for each team.

For my method I used a similar metric system to that of division one but with some modifications and extensions. The following is the point system that was used for matches played within a region.

For a win against a team ranked 1 through 5, 30 points are awarded.
For a win against a team ranked 6 through 8, 25 points are awarded.
For a win against a team ranked 9 through 12, 20 points are awarded.
For a win against a team ranked 13 through 16, 15 points are awarded.
For a win against a team ranked 17 through 20, 10 points are awarded.
For a win against a team ranked 21 through 30, 5 points are awarded.

Since matches are also played between teams of different regions I needed a point system for
them as well. However, since in division three there is less emphasis on national play, (due to lacks of funding for travel) I assigned fewer points for these matches. Here is the point system for matches between different regions:

For a win against a team ranked 1 through 5, 20 points are awarded.
For a win against a team ranked 6 through 8, 12 points are awarded.
For a win against a team ranked 9 through 12, 8 points are awarded.
For a win against a team ranked 13 through 16, 6 points are awarded.
For a win against a team ranked 17 through 20, 4 points are awarded.
For a win against a team ranked 21 through 30, 2 points are awarded.

I developed these point systems partly from the general structure of those used for division one and partly based on guess and check. By guess and check I mean that I would fiddle with the point assignments and see if the resulting rankings made sense based on each team's results. These results were taken from www.collegetennisonline.com. The main aspect of the project was to come up with some way to take each regional ranking and put them together into one national ranking. First I had to deal with the fact that some regions have more ranked teams then others and that each region can have a different number of teams. These two issues could create an effect where team A from region 1 has 30 points and team B from region 2 has 40 points, however, due to the number of matches that each team played and the strength of each region, team A's 30 points could actually have more bearing than team B's 40 points. To
resolve these issues I experimented with various methods, such as taking each team's 10 best
wins (the ones that give it the most points), and putting a weight on each region indicating that
region's strength (the number of ranked teams in that region). I would then scale all the weaker
region's points by that weight. Taking only the 10 best wins had a few problems. The most
important of these is that some teams do not play many matches early in the year and hence they
would have very few points and be ranked low, which in turn hurts teams that beat them, since
they would get fewer points. The second method (assigning weights to regions) seems to work
a lot better, except that after some testing I decided that I would need to add another term to
balance out the fact that some regions have fewer teams than other resulting in fewer possible
points. For the smaller regions I added a constant number of points to each team’s score based
on the average number of matches played in the small region versus the number of matches
played in larger regions. The value for this constant was chosen to be 20, and the region
strength constant was chosen to be 1.3. These constants may vary from year to year, but it is
highly unlikely that there will be enough perturbations in the rankings to change these constants
within a tennis season.

**Programming Method:**

To generate rankings I used SQL to create a database with two tables (see appendix). From
there I used C# and XML to write a program that looks at each entry in the Scores table, and
assigns an appropriate number of points to the winner of a match played within a certain period
of time. After this period of time the program updates each teams ranking. The program then
continues by looking at matches played within the next period of time. When assigning points the program looks at the Teams table to see what region each team belongs too and to determine each team’s current rank. It is impossible to update rankings continuously because if 2 matches are played on the same day, entering in one result first could change the rank of other teams and in turn affecting the number of points awarded to winners of other matches. For reference all the code is attached in the appendix

**Results:**

After running my program with the data inputted, I came up with the following rankings:

TeamID: 32, Name: Gustavus, **Position: 1** Points: 154.7

TeamID: 31, Name: Emory, **Position: 2** Points: 107.9

TeamID: 15, Name: Middlebury, **Position: 3** Points: 107

TeamID: 34, Name: Williams, **Position: 4** Points: 101.4

TeamID: 35, Name: Mary Washington, **Position: 5** Points: 97.5

TeamID: 1, Name: CMC, **Position: 6** Points: 95

TeamID: 3, Name: Redlands, **Position: 7** Points: 93

TeamID: 5, Name: Trinity, **Position: 8** Points: 85

TeamID: 6, Name: UC Santa Cruz, **Position: 9** Points: 81

TeamID: 4, Name: Cal Lutheran, **Position: 10** Points: 74.1

TeamID: 24, Name: MIT, **Position: 11** Points: 58.5

TeamID: 36, Name: Bowdoin, **Position: 12** Points: 32.5

TeamID: 33, Name: Kalamazoo, **Position: 13** Points: 26
The TeamID is just that team’s id value in the table.

The official rankings are displayed in the appendix. Comparing my results with the official ITA rankings, we see that the first 3 places match up exactly, and then there is some deviation, although most of the teams are in the right places. There are a couple of exceptions, however, the main one being Mary Washington College. I’m not sure why the ITA placed them so low on the rankings, because according to their record they beat a lot of highly ranked teams, which should have placed them higher. The other offset is Cal Lu and UC Santa Cruz. Cal Lu should actually be higher than my model placed them, especially because they had two direct wins over Redlands which is ranked above them with my model. The reason for this is because Redlands played a lot more matches against ranked teams and my coefficient that is supposed to deal with this effect did not adequately stabilize it. Another reason for the offsets in rankings is due to the fact that I did not have enough time to enter all the division 3 schools and their results, hence some of the points that these schools could have gotten are not covered by these results.

Conclusion:

Based on my results, a couple of conclusions can be drawn. In general my ranking system ranks teams correctly relative to each other, however; it is not good enough to be used without human aid. The ITA ranking committee puts a lot of emphasis on direct wins, so even if team A has fewer points than team B but team A has a win over team B, the ITA could place team A
ahead of B. This would be very hard to encode with my approach. For division three, the best method might be to use a ranking system similar to mine and then have a committee look over it and make some changes where it sees fit. This would save the committee a lot of time, it would not have to look through each team's record and then spend hours comparing them, but rather just look over the computerized rankings and change them if something seems completely wrong. That way, rankings would be able to come out more than once a year, even though they would not be updateable as often as I had anticipated.
Acknowledgments:

I would like to thank Professor de Pillis for her help in the organization of this project. I would also like to thank Michael Sing of the ITA for providing me with information on the division one ranking system, and Hank Krieger for giving me the idea for this project.
Appendix:

The following is the ASPX code used to take data out of the database tables and perform calculations:

```csharp
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace WebApplication1
{
    /// <summary>
    /// Summary description for WebForm1.
    /// </summary>
    public class WebForm1 : System.Web.UI.Page
    {
        protected System.Data.SqlClient.SqlConnection conn;

        private void Page_Load(object sender, System.EventArgs e)
        {
```
//generates database connection
conn = new SqlConnection();
conn.ConnectionString = "server=(local);
database=ranking;persist security info=False;user id=sa;Password=tiramesu; packet size=4096";
SqlDataAdapter teamAdapter = new SqlDataAdapter("SELECT * FROM TEAMS", conn);

//the following two lines extract all information
//from both tables
SqlDataAdapter scoreAdapter = new SqlDataAdapter("SELECT * FROM SCORES", conn);

results resultsTDS = new results();
teamAdapter.Fill (resultsTDS,"Teams");
scoreAdapter.Fill (resultsTDS,"Scores");

//goes through each row in the Teams table
foreach (results.TeamsRow tr in resultsTDS.Teams) {  
    // outputs each teams original information  
    Response.Write(String.Format("TeamID: {0}, Name: {1}, Position: {2}<br>", tr.TEAMID, tr.name, tr.IsdescriptionNull()? (short)-1 : tr.position));  
    //goes through the Scores table and outputs each result to screen  
    foreach (results.ScoresRow wsr in tr.WeWonFrom()) {  
        Response.Write(String.Format("Wins: RecordID: {0}, selfID: {1}, TeamID: {2}<br>", wsr.ID, wsr.winner_id, wsr.loser_id));
        // finds the loser in the Teams table and assigns points to the winner
        foreach (results.TeamsRow tq in resultsTDS.Teams) {  
            if(tq.TEAMID == wsr.loser_id)
                if(tq.description == tr.description)
                    if(tq.position > 1 &&
                        tq.position <=5)
tr.points += 30;

if(tq.position > 5 &&
if(tq.position <=8)
tr.points += 25;

if(tq.position > 8 &&
if(tq.position <=12)
tr.points += 20;

if(tq.position > 12 &&
if(tq.position <=16)
tr.points += 15;

if(tq.position > 16 &&
if(tq.position <=20)
tr.points += 10;

if(tq.position > 20 &&
if(tq.position <=30)
tr.points += 5;

// if the teams are from
different regions assign
different points

else
{
if(tq.position >= 1 &&
if(tq.position <=5)
tr.points += 20;

if(tq.position > 5 &&
if(tq.position <=8)
tr.points += 12;

if(tq.position > 8 &&
if(tq.position <=12)
tr.points += 8;

if(tq.position > 12 &&
if(tq.position <=16)
tr.points += 6;

if(tq.position > 16 &&
if(tq.position <=20)
tr.points += 4;

if(tq.position > 20 &&
if(tq.position <=30)
tr.points += 2;

}
}

})
Response.Write("<br><br>");
// apply the region strength constant of 1.3 to
//region c2 and add the other region 20 points
// because it is smaller
foreach (results.TeamsRow tr in resultsTDS.Teams)
{
if (tr.description == "c2")
    tr.points *=1.3;
else
    tr.points +=20;
Response.Write(String.Format("TeamID: {0}, Name: {1},Position: {2}<br>", tr.TEAMID, tr.name, tr.IsdescriptionNull()? (short)-1 : tr.position));
Response.Write (String.Format("Points: points: {0} &nbsp;&nbsp;<br>",tr.points));
}

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    //
    // sqlConnection
    //
    this.Load += new System.EventHandler(this.Page_Load);
}
#endregion

The following is the xml code that describes the database organization

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="results" targetNamespace="http://tempuri.org/results.xsd" elementFormDefault="qualified" attributeFormDefault="qualified" xmlnss="http://tempuri.org/results.xsd"
<xs:element name="Teams">
<xs:complexType>
<xs:sequence>
  // elements describe the columns
  <xs:element name="TEAMID" msdata:ReadOnly="true" msdata:AutoIncrement="true" type="xs:int" />
  <xs:element name="name" type="xs:string" />
  <xs:element name="description" type="xs:string" minOccurs="0" codegen:nullValue="_empty" />
  <xs:element name="position" type="xs:int" minOccurs="0" />
  <xs:element name="points" type="xs:double" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Scores">
<xs:complexType>
<xs:sequence>
  <xs:element name="ID" msdata:ReadOnly="true" msdata:AutoIncrement="true" type="xs:int" />
  <xs:element name="winner_id" type="xs:int" />
  <xs:element name="loser_id" type="xs:int" />
  <xs:element name="winner_score" type="xs:int" minOccurs="0" />
  <xs:element name="loser_score" type="xs:double" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
The Official ITA tennis rankings

1 Gustavus Adolphus
2 Emory
3 Middlebury
4 UC-Santa Cruz
5 Williams
6 Claremont
7 Cal Lutheran
8 Redlands
9 Trinity (TX)
10 Mary Washington College
11 Washington University
12 MIT
13 Bowdoin

The following two tables are the database tables.

**TEAMS TABLE**

<table>
<thead>
<tr>
<th>TEAMID</th>
<th>name</th>
<th>description</th>
<th>position</th>
<th>points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CMC</td>
<td>c1</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Pomona</td>
<td>c1</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Redlands</td>
<td>c1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Cal Lutheran</td>
<td>c1</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Trinity</td>
<td>c1</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>UC Santa Cruz</td>
<td>c1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Linfield</td>
<td>c1</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Middlebury</td>
<td>c2</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>MIT</td>
<td>c2</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>Emory</td>
<td>c2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>Gustavus</td>
<td>c2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>Kalamazoo</td>
<td>c2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>Williams</td>
<td>c2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>Mary Washington</td>
<td>c2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>Bowdoin</td>
<td>c2</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>ID</td>
<td>winner_id</td>
<td>loser_id</td>
<td>winner_score</td>
<td>loser_score</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>31</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>31</td>
<td>24</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>31</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>31</td>
<td>33</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>31</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>32</td>
<td>31</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>32</td>
<td>3</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>