# Network Effects in NBA Teams:

# Observations and Algorithms

by

Xiaoyu Zhang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2017 by the
Graduate Supervisory Committee:

Hanghang Tong, Chair
Jingrui He
Hasan Davulcu

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

The game held by National Basketball Association (NBA) is the most popular basketball event on earth. Each year, tons of statistical data are generated from this industry. Meanwhile, managing teams, sports media, and scientists are digging deep into the data ocean. Recent research literature is reviewed with respect to whether NBA teams could be analyzed as connected networks. However, it becomes very time-consuming, if not impossible, for human labor to capture every detail of game events on court of large amount. In this study, an alternative method is proposed to parse public resources from NBA related websites to build degenerated game-wise flow graphs. Then, three different statistical techniques are tested to observe the network properties of such offensive strategy in terms of Home-Away team manner. In addition, a new algorithm is developed to infer real game ball distribution networks at the player level under low-rank constraints. The ball-passing degree matrix of one game is recovered to the optimal solution of low-rank ball transition network by constructing a convex operator. The experimental results on real NBA data demonstrate the effectiveness of the proposed algorithm.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

## 1. Introduction

There is no doubt that the sport plays an indispensable role in the development of human history. It helps mankind build, maintain and improve physical health within both competitive and relaxing environment. The 1990s has witnessed the superior athletic ability of Michael Jordan, a miracle man who brought huge public attention to the basketball activities. Jordan's efforts on the court entirely elevated the people's perspective on basketball sport. In addition, Jordan exploited his talents in the pursuit of marketing and sponsorship. Under such background, athletic sports are not only ornamental games for on-site spectators but also evolved into an entertaining industry which contributes to the economy. For decades, the commerce and market have gained predominant status in modern society, influencing every aspect of people's daily life. Due to his great contributions, Jordan becomes an inspiring idol to many American people. Nowadays, basketball is one of the world's most popular and widely watched sports **[1]**. Founded in 1946, the National Basketball Association (NBA) is the most popular and widely considered to be the highest level of men's professional basketball league in the world **[2]**. NBA players are the world's best paid sportsmen, by average annual salary **[3][4]**. Over the 70 years, the league has grown up to 30 teams (29 in the United States and 1 in Canada), and divided into 2 conferences (Eastern and Western) and 6 divisions (Atlantic, Central, and Southeast in the Eastern Conference; Northwest, Pacific, and Southwest in the Western Conference). Routinely, there will be a game season hosted every year, comprised of three parts: Preseason, Regular season, and Playoffs. Every team will equally play 82 games for the regular season and only the top 8 teams among

each conference will be entitled to the playoffs. Taken regular season for simplicity, there will be at least 1230 games played every game season.

Before each game season starts, commentators from different sports media or website would deliver predictions in all kind, such as future trade-offs or estimations of average salaries for a specific team. It is remarkable that the win/loss prediction is one of the most heated topics. Each year millions of people get enthusiastic when watching their favorite NBA team winning the championship. People may want to ask, "How do basketball teams win games?" Dean Oliver identified what he called "the Four Factors of Basketball Success": Shooting, Turnovers, Rebounding, and Free Throws. In other words, the core idea of win-loss prediction approaches is to simulate the outcomes of basketball games by tracking related data.

With the advent of data mining techniques, many new metrics have been introduced for the time being. As a team sport, the good-looking statistics of "Four Factors" from one basketball team implicitly conveys better teamwork strategies than the opponent team during one game. Unlike many other sports, the basketball game does not revolve around a series of dyadic interactions (e.g. baseball, tennis) or an algebraic summation of individual efforts (e.g. track and field); it is dependent on a connected team network **[5]**. Hence, how to accurately employ network features on the NBA team model to learn teamwork effects becomes a favored problem.

Jennifer Fewell, a professor from School of Life Sciences in Arizona State University, explains that because teams are an integral part of both human and animal societies, understanding how interactions of one team affect its success or failure as a whole is important. To better study the teamwork dynamics, Dr. Fewell et al. compared the

patterns of different offensive strategies by National Basketball Association (NBA) teams during the first round of the 2010 play-offs to a measure of performance **[6]**. Based on Dr. Fewell's theory, there were two potential play styles for the offensive team: The first one is to always move the ball by personal expertise towards their shooting specialists, measured as "uphill/downhill flux", and the second one is to distribute the ball in a less predictable way, measured as team entropy. Dr. Fewell believed that the interactions within team members can be captured as a strategic network where the nodes are players, possible start-of-play (inbound, rebound, and steal) and outcomes (success, fail, foul, turnover), and a weighted link exists between possession origin, players and possession outcomes if ball movements were detected in the game.

In Dr. Fewell's network, player nodes from each team are categorized and represented by their positions. Traditionally, the modern NBA teams use a point guard, two wings, and two post players. To clarify the changes in the positional rules, five independent positions are now described: point guard (PG), shooting guard (SG), small forward (SF), power forward (PF), and center (C) in the basketball game **[7][8]**. From Dr. Fewell's initial observation, all offensive plays with at least three of the five starters on the floor were included **[6]**. Positions were therefore equated with specific players within each team and used as nodes instead of players.

To evaluate teams as networks, Dr. Fewell's lab members graphed player positions and ball movement among players, as well as ball status such as Shots Made and Turnovers by watching first round game videos of 16 play-offs teams. Then, they analyzed varieties of network properties such as path flow rate, degree centrality, and team entropy on the

graphed data to find out whether team decisions can be measured by network metrics in a useful way.

Apart from the achievements, there are three possible extensions for Dr. Fewell's research: (1) In NBA games, the present regulation allows unlimited player substitution within one game, which draws my attention that such action does not mandate same-position personnel swap and as a result may have impact on the experiment accuracy; (2) the analysis on teamwork dynamics treated two teams of one game as two isolated offensive networks, which depended heavily on intra-group cooperation but ignored the fact that how opponent team react will influence the offensive performance; (3) the research did not compare strength and weakness between different offensive patterns of teams.

Dr. Fewell's research mainly focused on analyzing network features of NBA teams. However, this thesis further surveys the possibility of automating the team offensive graph on games-wise basis in the NBA public domain. The way how Dr. Fewell's collect the data (manual labor) is time-consuming and nearly impossible to deploy given massive number (more than one thousand) of games. To resolve this issue, I went through the NBA official website http://stats.nba.com and found some useful resources named "Gamebook" under "Scores" tab. The Gamebook of NBA is an event log which takes down every play-by-play detail as well as traditional boxscore and lineups information of both teams within each single game. Even though being one of the most rich-content NBA public sources, the Gamebook does not offer full tracking of players passing information other than assist passing. So far, there is no direct way to build complete ball transition network of every moment by parsing NBA public materials. My hypothesis is

that by observing the indegree and outdegree together with few entries of matrix representation of the ball flow graph, we may be able to recover the network under a low-rank condition.

The main contributions of this thesis are: (1) This thesis mined NBA regular season gamebook information from online public resources; (2) This thesis built degenerated game-wise strategic flow graphs on top of the retrieved data; (3) This thesis developed an algorithm to infer real game ball distribution networks at the player level under low-rank constraints. It may be trivial but serves as a modest spur to induce someone to come forward with his/her valuable contributions.

The remaining chapters of this thesis are organized as follows:

Chapter 2 provides a review of past research literature. This chapter will review the sources related to the topic and raises the idea for this thesis experiments.

Chapter 3 defines the problem this thesis intends to solve.

Chapter 4 presents the solid observations from the network analysis. This chapter will firstly introduce the way to collect public resources of NBA regular season games and analyze the network effect in the context of NBA teams. In addition, statistics of player tracking data is also extracted online to enhance the observation part.

Chapter 5 describes the proposed algorithm to infer team player passing networks.

Finally, chapter 6 concludes the thesis.

## 2. Literature Survey

Before the evolution of data mining, numerous sports leagues such as NBA, MLB, and NFL relied heavily on those scouts, team managers, coaches, and players with sufficient expertise, who are believed to have the ability of transforming their discovery and inspiration into useful knowledge. When the scale of sources became overwhelming and diverse thanks to the growth of social media, sports organizations began to find it less effective to provide information to such specialists who made right decisions in the old-fashioned way, leading to an exclusive phenomenon "Rich data, but poor knowledge". The sports industry was thus motivated to seek solution of delivering the data they collected in more concise and efficient ways.

Data mining is about solving problems by analyzing data already present in databases **[9]**. Data mining is defined as the process of discovering patterns in data. In literal, it helps the end users to extract useful information from large databases. On the other hand, data mining is the nontrivial extraction of implicit, previously unknown and potentially useful information from the data mountain **[10]**. It features the function of the data transition process of 4 stages: from Data, to Information, to Knowledge, and finally Wisdom **[11]**. Data mining methods have been successfully applied to sports in many fields such as soccer, hockey, and so on. The most famous one is baseball. It was all started with the book Moneyball which shared a story about Billy Beane, a general manager of the Oakland A's, who pioneeringly used data mining knowledge to coach his club win the competition against opponents with much higher payrolls **[12]**. The Oakland A's, short for Oakland Athletics, had long been under serious financial distress. With members getting paid remarkably lower than the league average, it was rarely possible to see high

quality players remained on the A's. Instead, Beane used the data to fight back. Inspired by his pioneering mind, Beane implemented the "Sabermetrics" approach to unearth potentially outstanding players with affordable payroll who were in fact undervalued by the conventional statistics. Despite at low cost operation, Beane's effort has finally paid off when his team defeated tough competitors.

The publishing of Moneyball was clearly a breakthrough in the long run. As the aftershock of baseball, Dean Oliver began to popularize the use of possession statistics in the 1990s **[13]**. The term APBRmetrics (Association for Professional Basketball Research Metrics) was formerly invented by a group of amateurs for the sake of basketball analysis through objective evidence, especially basketball statistics. Like its sibling study Sabermetrics of baseball statistics, the APBRmetrics are used to create better measurements and statistical yardsticks for comparison purposes **[13]**.

The statistics of individual players and team performance have been adopted to understand the win-loss probabilities. For modern basketball analysts, a key tenet is that basketball is best evaluated at the possession level **[14]**. The possession, defined as the time a team gains offensive possession of the ball until it scores, loses the ball, or commits a violation or foul, is an important statistic since it allows teams to compute performance on a per possession basis **[14]**. It is customary practice that analysts use the points scored per 100 possessions as offensive ratings and points allowed per 100 possessions as defensive ratings. K. Kang first went over some traditional statistics used in NBA including simple plus-minus, and adjusted plus-minus **[15]**. Then Kang pointed out the drawbacks for both existed methods: simple plus-minus is basically the live difference in score while game is in progress, making it difficult to count individual's

contributions; Though adjusted plus-minus is designed to solve the previous flaw, evaluation result may fall in high variance given players of game season events in large amount. Kang further constructed a penalized regression model to identify the specific offensive and defensive contributions of each player on each possession, and tune the model using L2-regularization method to optimize its predictive power.

A second core tenet is that per-minute statistics are more useful for evaluating players than per-game statistics. R. Sisneros and M. V. Moer picked up plus-minus statistics of the NBA box-score data and proposed a new visualization tool **[16]**. A player's plus-minus is calculated as the difference of his team's and the opposing team's points while he is on the floor. Player efficiency rating (PER) is a single-number measurement of how many good things a player does minus the bad things he does per unit of playing time. In traditional evaluation, undervalue defensive experts whose presence is valuable but is not unaccounted. Plus, all contributions are divided by a player's minutes played. Hence, metrics mentioned above are relatively unfair towards players with lower playing time, though rankings are only provided for those with some minimum amount of minutes. Sisneros's considered traditional plus-minus numbers at the team level as a measurement of the quality of a win-loss likelihood for a team. The best player on a bad team is not necessary and is unlikely to be "better" than several of the best players on the best teams. When it comes to win-contribution metric, defensive players rank considerably higher. All-defensive first team member Tony Allen is widely considered a great perimeter defender and is a perfect example. His win-contributions rank him 61$^{st}$ versus 167$^{th}$ and 197$^{th}$ via his ESPN rating and PER. In summary, Sisneros's presented a specialized tool for visualizing the plus-minus of team statistics (PluMP), and a new metric for evaluating

the win-contribution of a player based on statistics most relevant at the team level. For Sisneros's detailed PluMP: The Plus-Minus Plot Games with the same point margin are binned together. Vertical range for the assists is then the highest and lowest assist totals for these bins. The value of probabilities in his calculations may be outside the range between 0 and 1.

The research paper Forecasting NBA Player Performance by D. Hwang described a method for predicting player performance by utilizing a statistical timing model and a 'Weibull-Gamma' distribution **[17]**. The author emphasized the need to use a Weibull-Gamma (WG) model that basketball performance is memory-dependent on previous events. In other words, sequence has strong effect on basketball performance.

To overcome the major trouble causing variances on individuals who over-perform on offense but under-perform on defense from existing evaluation methods, J. Piette et al. adapted a network-based algorithm to estimate centrality scores and corresponding statistical significances **[18]**. Piette's constructed a weighted network by taking players as nodes and drawing edges between two players who ever played together in the same 5-man unit. Then they applied different measures of unit performance other than traditional regression methods to the constructed network. As a result, their approach worked greatly in comparing the centrality importance of difference two teammates performance as a pair.

With the promotion of electronics industry, the STATS company which is an official tracking partner of NBA, offered a new generation camera system called SportVU over all 30 arenas starting in the 2013-14 season that collects the real-time tracking data of players and the ball at a rate of 25 times per second. Player tracking data is by far the

most revolutionary data source for basketball analysis. In recent studies by B. Skinner and S. J. Guy, they proposed a method using player tracking data combined with a network model of the offense data to learn players' skills and predict the related team performance **[19]**. The model they brought to life is called "high/low" model, in which the state of "high" or "low" corresponds to the distance of the player to the basket. Running through the hand-recorded data for testing purpose, their model could work consistently with good prediction accuracy despite a very little input data set.

3. Problem Definitions

| Notations | Definition |
|---|---|
| $G := \{T, Y\}$ | The entire graph of strategic flow-graph in regular season games |
| $T_{n\times n\times k}$ | The game tensor of $G$ in the form that the first two dimensions reference a weighted adjacency matrix $M_{n\times n}$ by indexing $k$ at $G(K)$ |
| $M_{n\times n}$ | The adjacency matrix reflecting strategy flows between each applicable node |
| $Y_{k\times 1}$ | The game Win-Loss label of $G$ with respect to Home team |
| $n$ | The total number of basketball positions (PG, SG, PF, SF, C) plus a fixed number of reserved nodes (REBOUND, STEAL, MISS, TURNOVER, SUCCESS) |
| $k$ | The index number of games in terms of Home-Away order pair |
| $G(K)$ | The game strategic flow-graph of $G$ indexed by game id $K(K \in k)$ |
| $p_{i\to j}$ | The probability that player with position $i$ determines to pass the ball to player with position $j$ |

Table 3.1 Table of Notations for Network Effect Observation

Problem 1. Team Offensive Pattern Extraction by NBA Position

Given: (1) A training set of graphs $G_{train} := \{T, Y\}$;

Output: A "best" choice of sub-graph *M'* of core factors for Home team to win the match.

| Notations | Definition |
|---|---|
| $N := \{C^{(H)}, C^{(A)}\}$ | The entire integration of passing networks during regular season games |
| $C^r_{k\times 1}$ | The cell vector of $N$ in the form that each row stores a weighted adjacency matrix $M^r_{n^r\times n^r}$ of team location $r$ |
| $M^r_{n^r\times n^r}$ | The adjacency matrix of player tracking data in the form that each entry $M^r_{pq}$ represents an absolute frequency of ball passing from player $p$ to player $q$ |
| $n^r(K)$ | The total number of players of team location $r$ appeared on court indexed by game id $K$ $(K \in k)$ |
| $k$ | The total number of matches during the regular season |
| $r$ | The team location of $\{(H)ome, (A)way\}$ |

Table 3.2 Table of Notations for Low-rank Matrix Completion

Problem 2. Low-rank Ball Flow Matrix Completion

Given: (1) $D = [r_1, \dots, r_n]'$ and $F = [c_1, \dots, c_n]$ be possible integer vectors such that

$$sum(D) = sum(F)$$

(2) the player assist matrix $T$ derived from the strategic flow-graph in Table 3.1

(3) the set $S(D,F)$ of non-negative integer $n\times n$ matrices with row sums $D$ and column sums $F$ such that

$$S(D,F)=\left\{M=(m_{ij}):\sum_{j=1}^{n}m_{ij}=r_i\right.$$

$$\sum_{i=1}^{n}m_{ij}=c_j$$

$$m_{ii}=0$$

$$\left.m_{ij}\geq T(i,j)\right\}.$$

Output: Find the solution of matrix completion of $M\in S$ that minimize the nuclear norm $\|M\|_*$.

# 4. Empirical Observations

## 4.1 Data Collection

### 4.1.1 Overview

The NBA Media Ventures officially runs a web service called stats.nba.com which can be utilized as a HTTP API that offers a whole bunch of endpoints for developers or sports fans to access. For conservative estimate, there are at least 100 active endpoints working online **[20]**. Results can be gathered and returned directly within seconds in JSON format by posting requests like "stats.nba.com/stats/{endpoint}/?{params}" where params are the payload of all parameters bound to that endpoint. However, for some information, in particular for player tracking data, the direct method is otherwise invalid. For scenarios like that, the indirect approach is presented later to scrape data through specific webpage by employing web mining techniques.

### 4.1.2 Web Mining Packages

A. Requests

The Requests library is excellent at handling complicated HTTP requests, without the need for manual labor **[21]**. Unlike urllib2, a default module comes with Python for opening URLs, Requests provides much simpler and cleaner codes when programming. It overcomes most of the difficulties faced with urllib2, of which API is thoroughly broken, making interacting with web services seamless.

Written in Python, Requests library can be installed with all major package distribution managers.

B. Selenium

Selenium is a powerful web scraping tool developed originally for website testing **[22]**. Selenium-Python bindings provide a convenient API to automating browsers like Firefox, Chrome, or Internet Explorer to load the website contents and take further actions on retrieved data.

The current supported Python versions are 2.7, 3.2, 3.3 and 3.4.

C. BeautifulSoup

BeautifulSoup is a Python library designed for quick turnaround projects like screen-scraping **[23]**. It has several advantages which overthrow other crawling tools: BeautifulSoup provides a few methods and Pythonic idioms for navigating, searching and modifying a parse tree. When an incoming document doesn't specify its encoding or the encoding of the content cannot be detected, BeautifulSoup automatically converts the file into Unicode and outputs results in UTF-8 encoding, which reduces the labor to specify the original file encoding. BeautifulSoup sits on top of popular Python parsers like XML and HTML5 lib, allowing different parsing strategies and flexibility. BeautifulSoup is capable of parsing various kinds of input, as well as implementing the tree traversal. It can interpret some natural language context such as "find the table heading that's got bold text".

In this way, valuable data that was once locked up in poorly-designed websites now can be taken in just several minutes.

### 4.1.3 Gamebook Crawling Setup

The modern NBA game season is comprised of a regular season, one All-star weekend, playoffs, and finals. Every season only the top 8 teams in each conference (East and West) will be able to enter playoffs, competing for final championship, bringing more

tensions to coaches and players on both side. Teams qualified for playoffs usually performed quite different than what they did during regular season due to many factors such as more coaching time to prepare for games, alternating offense strategies, and trading players. Therefore, I filtered out all records of playoffs season when crawling the entire 2014-15 NBA season gamebooks to fulfill a more reliable statistical learning.

In order to crawl the pdf file of each gamebook, I analyzed the type of the webpage on how it navigates in stats.nba.com. The navigation of gamebook pdf is different than that of data tables such as numbers obtained in stats.nba.com. Numeric record id of each game is iterated in an ascending order (e.g. from 21400001 to 21401230).

However, gamebook links are not retrievable through method above. The crawling started from the scratch of iterating the date of game days in the 2014-15 regular season (from Oct 28, 2014 to April 15, 2015). By iterating each day in the given range, a valid game information can be retrieved by Selenium package – BeautifulSoup module on Python. The gamebook pdf hyperlink is under the attribute "href". By getting the value of "href" tag, a permanent link for downloading the pdf file can be composed.

The present spider composed 1231 gamebooks that all 30 teams participated in the regular season (1230 games) along with 1 special match for EAST-WEST All-star (Members were chosen from players of multiple teams. Archived in advance). Besides, 6 gamebooks of 2015 NBA Finals between CLE and GSW (from June 4-16) were scraped and archived as well.

In total, 1230 pdf records per season were retained for further analysis.

#### 4.1.4 Players Profile

At the beginning part of each gamebook, the team roster of both Home and Away teams is given. However, only five lineup members on each side have been assigned with implicit notation of their position except for C-Center, making it quite difficult to interpret position PG/SG from G or PF/SF from F. Meanwhile, the rest other than lineups have no position information.

In order to have a better understanding of the player's feature, I targeted another website called basketball-reference.com where contains detailed intel of players such as explicit position, years of experience, and college each team each season **[24]**.

By traversing each page link and parsing the table element behind, team roster files of 15 seasons ranging from 2001-02 season to 2015-16 season were archived into 15 separate season-wise folders.

#### 4.1.5 Boxscore Summary

The boxscore summary data is retrievable through direct method, that is, by accessing webpage like "http://stats.nba.com/stats/boxscoresummaryv2?GameID={gameid}" where gameid is the numeric range from 0021400001 to 0021401230 for NBA Season 2014-15 and 0021500001 to 0021501230 for NBA Season 2015-16.

By traversing through the above webpages iteratively, files containing useful game information such as actual game date and time, Home and Away team names, and final scores were stored onto the local disk with the naming convention of game id.

In total, there were 2460 JSON files saved for the recent two seasons.

#### 4.1.6 Players Passing Tracking

As mentioned above, there is no direct way to trace down to player tracking level data from the public NBA website API. Instead, the player tracking data, such as players passing can be scraped through webpage like “http://stats.nba.com/players/passing/” followed by several filters to control when and which team members’ data to display. These filters are the payload of HTTP request: Season, SeasonType, DateFrom, DateTo, PerMode, and TeamID. The highest precision is the total number of ball passed and received by player per game each team.

In order to build the correct mapping between each single math date (by setting DateFrom and DateTo equal), the team id of Home and Away teams (TeamID), it is time to make the use of boxscore summary data mentioned in Section 4.1.5. Boxscore summary, retrieved through NBA official website API, contains dramatically trivial but helpful information on per-game basis, ranging from the abbreviation for Home/Away teams, number on player’s clothes, to the time-specified date of game being held. With the assistance of boxscore summary data, I managed to compose the correct link of target webpage for further web mining process.

In total, there were 4910 (2454 for 2014-15 season and 2456 for 2015-16 season) instances prepared.

### 4.2 Data Processing

#### 4.2.1 Constructing the Game-wise Offensive Network

All pdf files were first converted to text files using batch-processing tool on local disk. Next, two adjacency matrices denoting Home and Away team were initialized of size m×m and n×n where m, n corresponds to the number of show-up players from each team

plus 5 index spaces reserved for Rebound, Steal, Miss, Turnover, and Success status. Two additional matrices of size m×n and n×m were introduced to store ball transitions between Home and Away team.

By using regular expression, each text file was visited line by line to see if it contains certain formatted data. A valid line may look like “02:38 Harris 24' 3PT Jump Shot (Nowitzki)” or “:28.8 MISS Mack 25' 3PT Pullup Jump Shot”. It is remarkable that every time stamp will be only associated with one event, no matter scoring or non-scoring. Different event results in different operation. Two scoring events are: “X … (Y)” and “X …” with scores displayed at same line. There are four non-scoring events namely “MISS X …”, “Y REBOUND”, “X TURNOVER”, and “Y STEAL”. “X … (Y)” yields a directed path from assistant player Y to scoring player X, who further connected to SUCCESS. “X …” yields a directed path from scoring player X to SUCCESS. “MISS X …” yields a directed path from player X who failed scoring to MISS; If “Y REBOUND” appears on the next-closest line, then the complete path is updated to be X→MISS→REBOUND→Y where Y can be from either team. “X TURNOVER” yields a directed path from player X who lost the ball to TURNOVER; If “Y STEAL” appears on the same line, then the complete path is updated to be X→TURNOVER→STEAL→Y where Y is for sure from opponent team.

Once finished processing each text file, one csv file representing Home-Away team game-wise offensive flow graph was created to store the outputs in the form of:

- The dimension of both Home and Away team’s adjacency matrix are fixed (REBOUND, STEAL, MISS, TURNOVER, SUCCESS, PG, SG, PF, SF, C).
- The position of each player is parsed by matching his team roster record.

- The diagonal may have non-zero value due to players with same position were grouped into one node.

#### 4.2.2 Players Resume Buildup

There are many sources storing players information in all kinds. We however aim to find the correlation between players throughout the entire league. By walking through the 15 seasons of players profile, an integrated table was created for each year, of which rows represents each unique player and columns are 30 NBA teams plus 5 distinct position on court. The value of the content is logical matrix. A sample row will deliver information such that for the first 30 columns of all '1s' entries are the team he served for during current season and the entry marked '1' out of last 5 columns is the professional position he's good at.

### 4.3 Description of Methods

#### 4.3.1 Win-loss Prediction

The win-loss prediction is by literal a binary-class classification problem. There are several well-known methods associated with statistical classification that can be considered as workable solutions, such as Logistic Regression, Artificial Neural Network (ANN), and Support Vector Machine (SVM). The main idea of proposed methods is to train parameter vectors, by going through certain training data set with known game result, which can be later applied to predict unknown game outcome of testing data set. The optimization goal which is to minimize the error rate in general, defined as cost function, is proven to be convex.

To comply with the linear model, the graph was flattened in column-major order by sacrificing its structure and the success nodes of Home and Away teams were hidden.

### 4.3.2 PageRank Score

To validate that the automated graphs based on our description were built correctly, we move on to choose some graph-related methods to infer the network features.

PageRank algorithm was first introduced by Larry Page, the co-founder of Google. It is wildly used by search engines to rank websites of importance. The algorithm is given as follows:

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta$$

For each graph, we compute the PageRank vector, and recorder two PageRank values for home-success node and for away-success node. In this way, each graph becomes a vector of two numbers. Then we apply above action for all the 1230 graphs, and plot the PageRank scores in a 2-dimentional plot. In this way, each game-wise graph becomes a dot in this 2-dimentional plot. The x-axis represents the PageRank score for home-success node, and the y-axis represents the PageRank score for away-success node. Red color is used for the graph that home team wins and blue color is used for the graph that away team wins.

Furthermore, we can actually keep all the 20 PageRank scores for 20 nodes for each graph. In other words, we can generate a 1230×20 PageRank feature matrix of which rows are different graphs/matches, and columns are the PageRank scores for different nodes in that match. We can do some feature analysis/selection to see if the PageRank scores for some particular position/status nodes are correlate with who wins.

### 4.3.3 Graph Similarity

Based on our experience, the graph similarity variance between Home-wins and Home-losses should be large. For Home versus Away teams, we compute the pairwise similarity between the 1230 graphs, in which 707 are wins and 523 are losses. The cosine similarity is denoted as:

$$\mathrm{sim}(A_1, A_2) = \frac{\mathrm{vec}(A_1)^{\mathrm{T}} * \mathrm{vec}(A_2)}{\|A_1\|_{\mathrm{F}} * \|A_2\|_{\mathrm{F}}}$$

Then the output is visualized by assigning win-win, win-loss/loss-win, and loss-loss node groups with three distinct colors. For each colored group, a histogram was created to store the normalized similarity values.

In this way, an initial insight into the team networks comparison is produced.

## 4.4 Observation Results and Evaluation

### 4.4.1 The Logistic Regression of Win-loss Prediction

For the first round of logistic regression, the testing set accuracy is 91.129% and validation set accuracy is 93.0612%. This triggers the doubt that whether linear model is suitable for graph classification problems. It is also possible that the training data contains hidden information about output label.

### 4.4.2 The 2-D Plot of PageRank

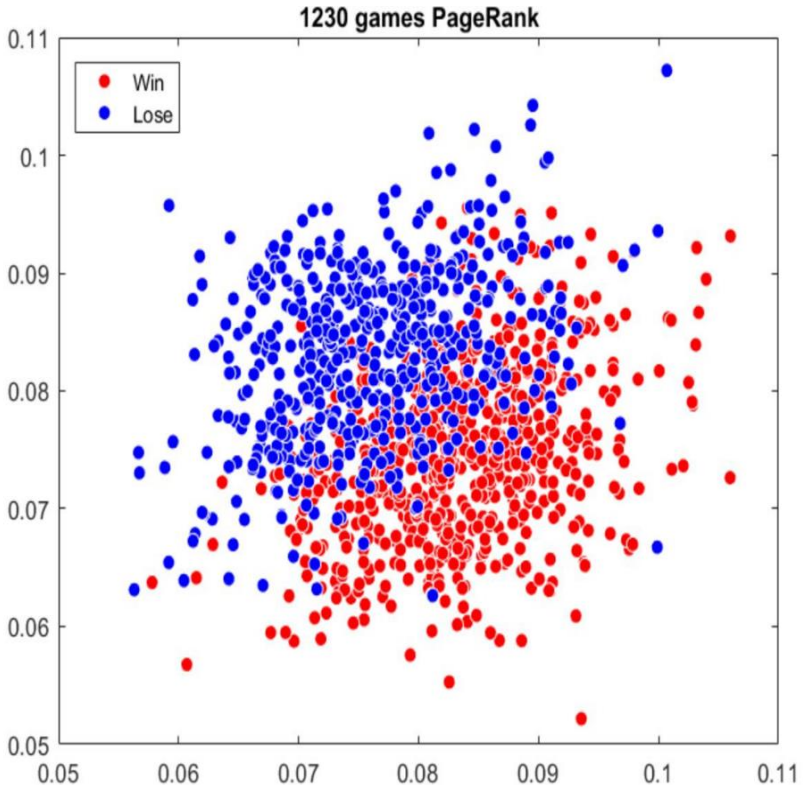


Figure 4.4.1 PageRank Score Plot for Game-wise Offensive Network

Now we plot a straight line from left bottom corner to the right top corner. As we can see from the plot, the PageRank algorithm works properly on the majority of game graphs, giving correct centrality importance to the corresponding sink (Home-success or Away-success) node.

### 4.4.3 The Histograms of Three Groups of Graph Similarity

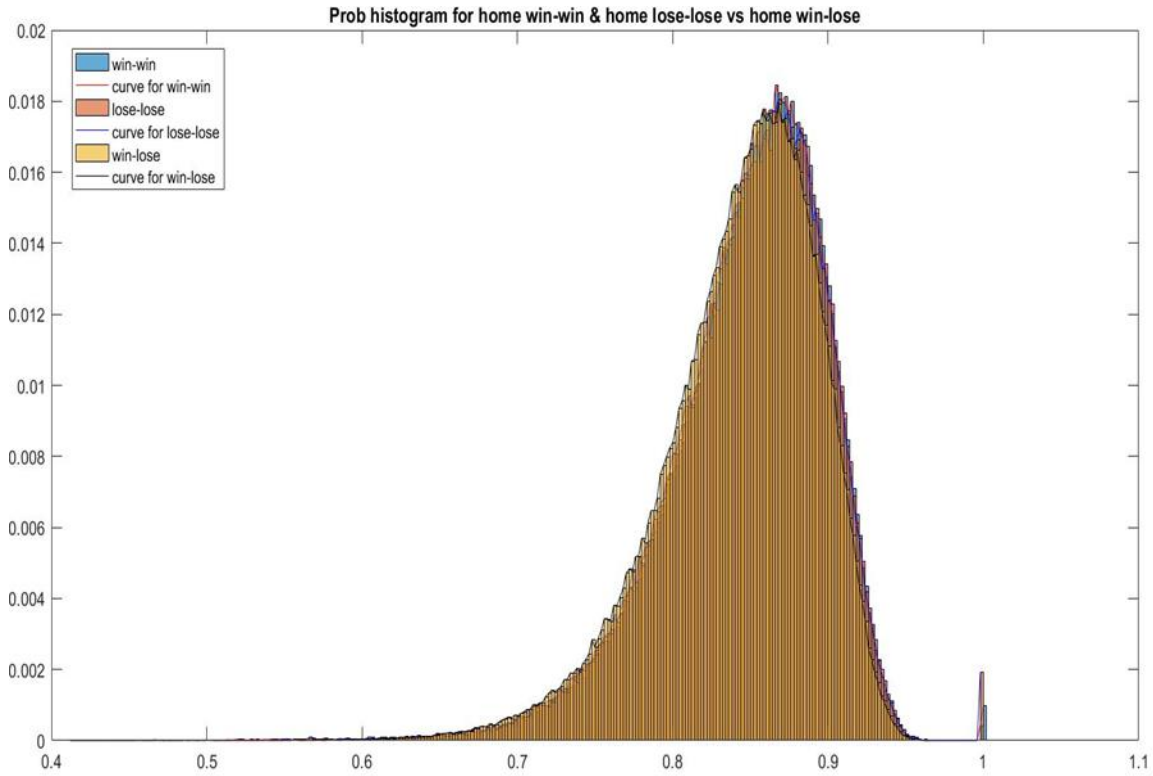


Figure 4.4.2 Normalized Probability Histogram for Home Graph Similarity

As we can see from the above graph, the curves of three histograms of graph similarity values overlap with each other greatly. For game flow graph with win/loss outcome analysis, we wish to see remarkable similarity variance between win-win and win-lose graphs to make our assumption effective. However, neither win-win nor lose-lose graphs differ greatly from win-lose graph in such pattern, making the network effect rarely obvious.

## 5. Algorithms to Infer Team Networks

### 5.1 Overview

In 2008, J-F. Cai's introduced a singular value thresholding (SVT) algorithm in details to approximate the matrix with minimum nuclear norm among all matrices obeying a set of convex constraints **[25]**. The essence of SVT algorithm is to recursively take singular value decomposition (SVD) on top of the updated observations. After each step, observations on the original unknown matrix are updated by running a thresholding function which filters out singular values smaller than the pre-set threshold on the diagonal matrix of last SVD operation.

Further reading his paper, I found my problem (See 4.2) can be adapted to a special case of the more generalized constraint SVT algorithm. By constructing a linear mapping like $\mathcal{A}(\mathbf{X}) = \mathbf{A} * \text{vec}(\mathbf{X})$, the SVT method is guaranteed to have low-rank solution **[26]**. In this way, the algorithm problem is transformed into modeling the correct operation matrix **A** of our original observations.

This is the first time that a recommendation algorithm has been taken full advantage of to implement matrix completion under NBA scenario. There are mainly two reasons why the real ball transition network may have low-rank property: (1) For a regular NBA team, the 5-man unit on court will not distribute the ball in a total random way, the tactics focus on creating denser links to some core players with better shooting abilities, leaving the entire team-level players passing network to be low-rank; (2) For teams whose team members have more balanced ball distribution manner, the rank of such players passing network is close to one.

The original formula was defined as:

$$\min_{X \in C} \tau \|X\|_* + \frac{1}{2}\|X\|_F^2$$

$$C=\{X \in R^{m \times n}\ s.t.\ X(i,j)=M(i,j)\ \forall (i,j) \in \Omega\}$$

## 5.2 Linear Equality Constraints

### 5.2.1 Optimization Goal

$$\underset{X}{\operatorname{argmin}} \left(\tau \|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X}\|_F^2\right)$$

$$\text{s. t. } \mathcal{A}(\boldsymbol{X})=\boldsymbol{b}$$

$$\text{start with } \mathbf{y}^0 = 0, \begin{cases} \boldsymbol{X}^k=\mathcal{D}_\tau\left(\mathcal{A}^*(\boldsymbol{y}^{k-1})\right), \\ \boldsymbol{y}^k=\boldsymbol{y}^{k-1}+\delta_k\left(\boldsymbol{b}-\mathcal{A}(\boldsymbol{X}^k)\right). \end{cases}$$

### 5.2.2 Algorithm

Input: linear vector $b$ and linear mapping $A$, matrix size $n$, threshold parameter $\tau$, step size $\delta$, tolerance $\epsilon$, and maximum iteration $k_{max}$

Output: $X^{opt}$

Description: Recover a low-rank matrix $X$ from a linear mapping of original matrix

Set $y^0 = \delta b$

Set $r_0 = 0$

for $k = 1$ to $k_{max}$

Compute $Y^{k-1} = ivec((A^T * A) \backslash A^T * y^{k-1}, n)$

Compute $[U^{k-1}, \Sigma^{k-1}, V^{k-1}]$

Set $r_k = \max\{j: \sigma_j^{k-1} > \tau\}$

Set $X^k = \Sigma_{j=1}^{r_k}(\sigma_j^{k-1} - \tau)u_j^{k-1}v_j^{k-1}$

if $\left|\left|A * vec(X) - b\right|\right|/||b|| \leq \epsilon$ then break

Set $y^k = y^{k-1} + \delta * (b - A * vec(X))$

end for k

Figure 5.2.1 Pseudocode for SVT Algorithm with Linear Equality Constraint

### 5.2.3 Some Observations

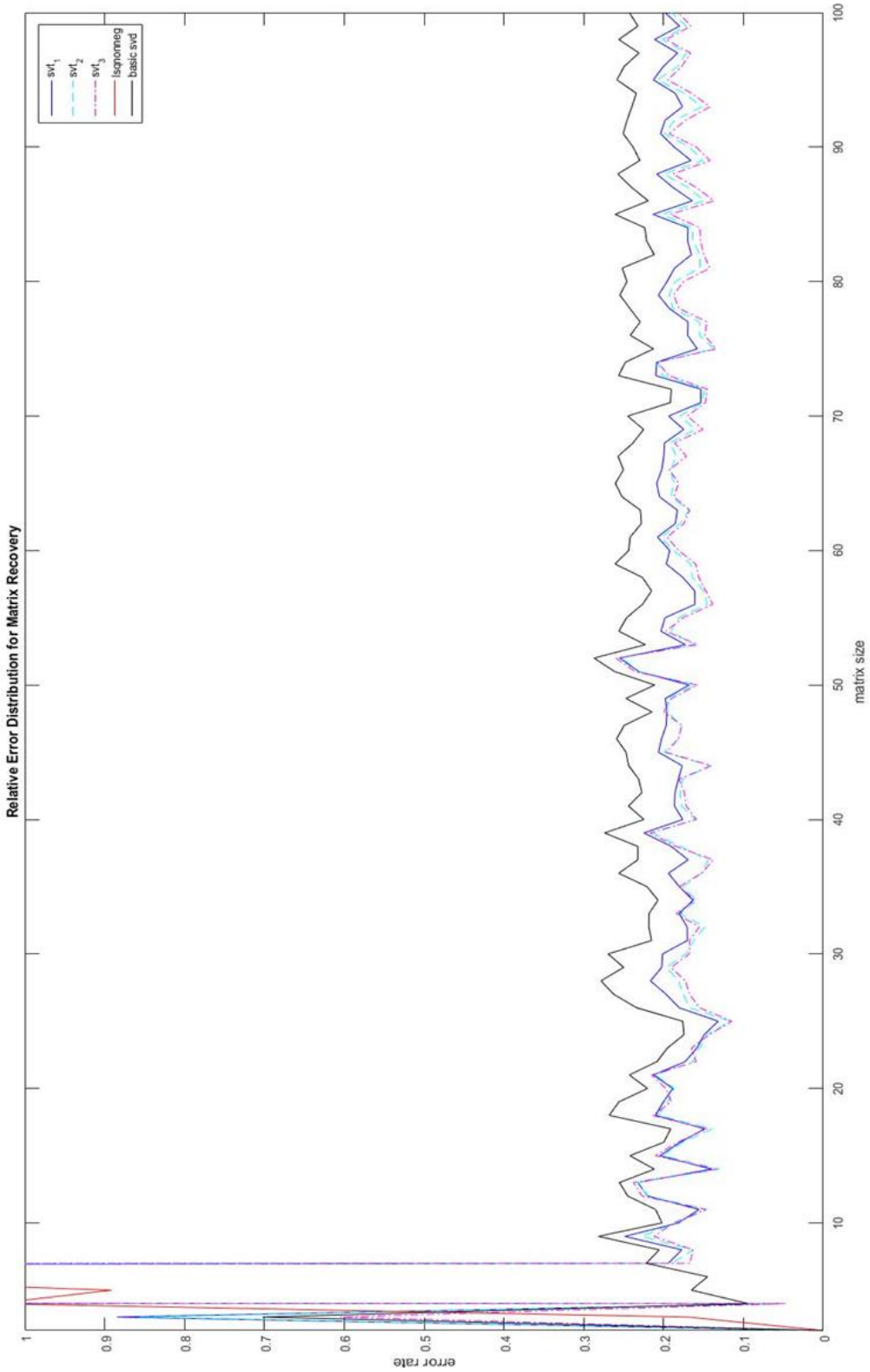


Figure 5.2.2 Error Effectiveness for SVT Algorithm with Linear Equality Constraint

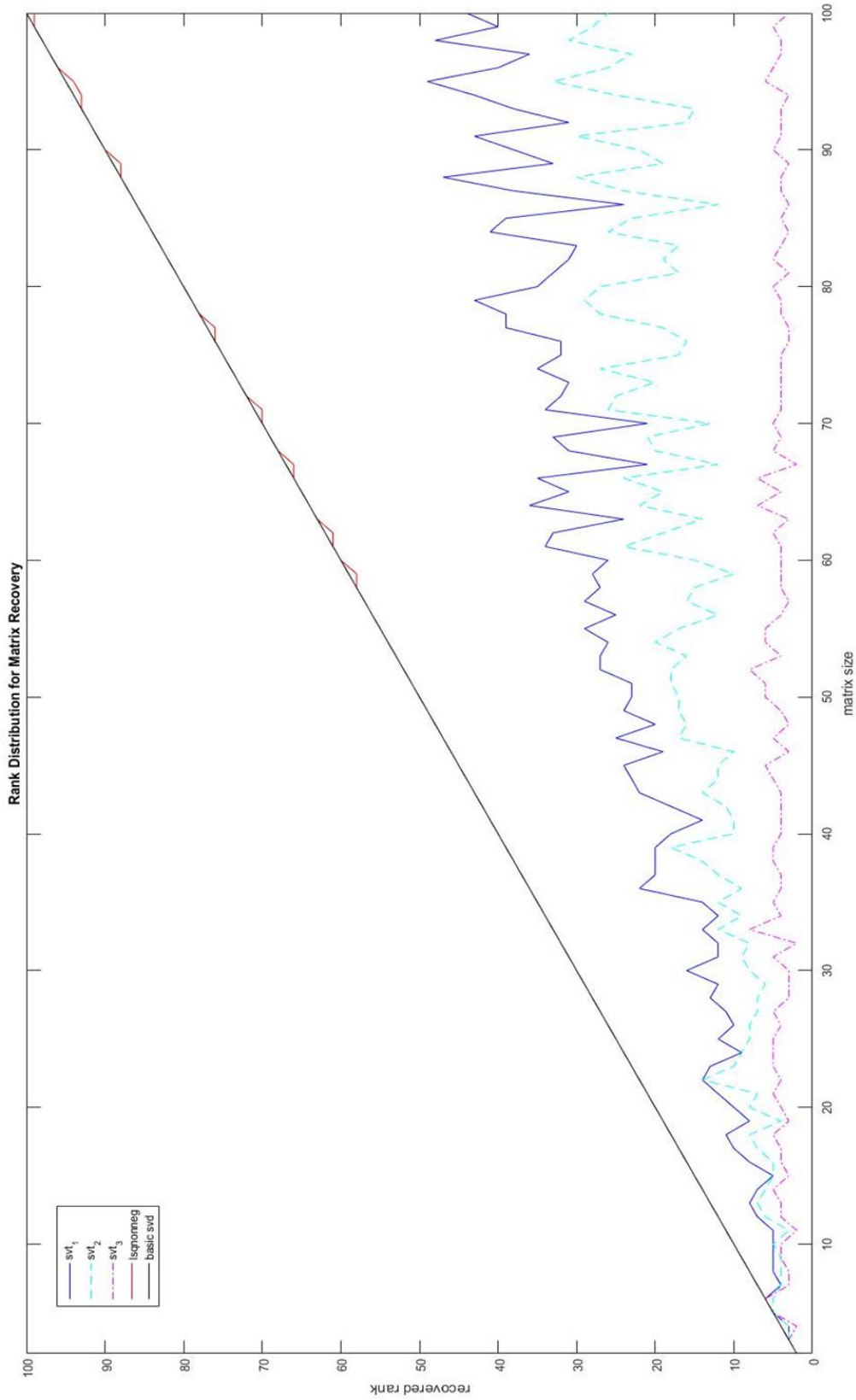


Figure 5.2.3 Rank Effectiveness for SVT Algorithm with Linear Equality Constraint

## 5.3 General Convex Constraints

### 5.3.1 Optimization Goal

$$\underset{X}{\operatorname{argmin}} \left( \tau \|\boldsymbol{X}\|_* + \frac{1}{2} \|\boldsymbol{X}\|_F^2 \right)$$

$$\text{s.t.} \quad \boldsymbol{b} - \mathcal{A}(\boldsymbol{X}) \leq 0$$

$$\text{start with } \mathbf{y}^0 = 0, \begin{cases} \boldsymbol{X}^k = \mathcal{D}_\tau \left( \mathcal{A}^* (\boldsymbol{y}^{k-1}) \right), \\ \boldsymbol{y}^k = \left[ \boldsymbol{y}^{k-1} + \delta_k \left( \boldsymbol{b} - \mathcal{A}(\boldsymbol{X}^k) \right) \right]_+ . \end{cases}$$

### 5.3.2 Algorithm

Input: linear vector $b$ and linear mapping $A$, matrix size $n$, threshold parameter τ, step size δ, tolerance ϵ, and maximum iteration $k_{max}$

Output: $X^{opt}$

Description: Recover a low-rank matrix $X$ from a linear mapping of original matrix

Set $y^0 = \delta b$

Set $r_0 = 0$

for $k = 1$ to $k_{max}$

Compute $Y^{k-1} = ivec((A^T * A) \backslash A^T * y^{k-1}, n)$

Compute $[U^{k-1}, \Sigma^{k-1}, V^{k-1}]$

Set $r_k = \max\{j: \sigma_j^{k-1} > \tau\}$

Set $X^k = \Sigma_{j=1}^{r_k} (\sigma_j^{k-1} - \tau) u_j^{k-1} v_j^{k-1}$

if $\left| ||A * vec(X) - b| \right| / ||b|| \leq \epsilon$ then break

Set $y^k = \max(y^{k-1} + \delta * \left(b - A * vec(X)\right),\ 0)$

end for k

Set $X^{opt} = X^k$

Figure 5.3.1 Pseudocode for SVT Algorithm with General Convex Constraint

### 5.3.3 Some Observations

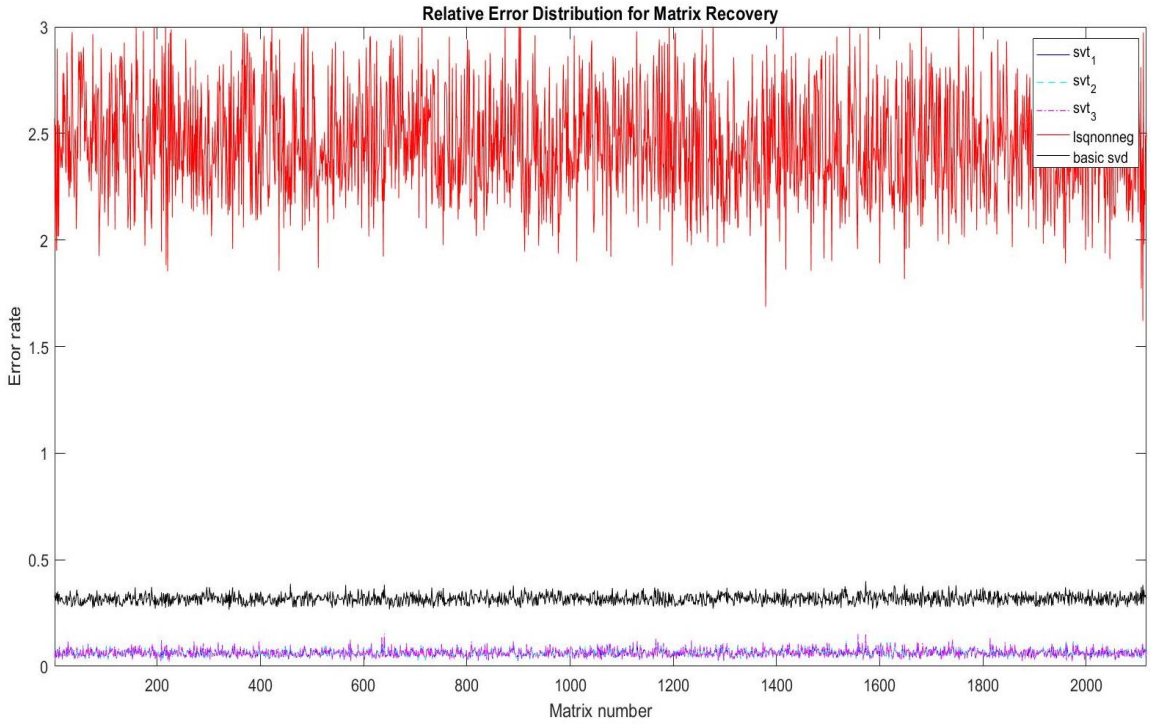


Figure 5.3.2 Error Effectiveness for SVT Algorithm with General Convex Constraint

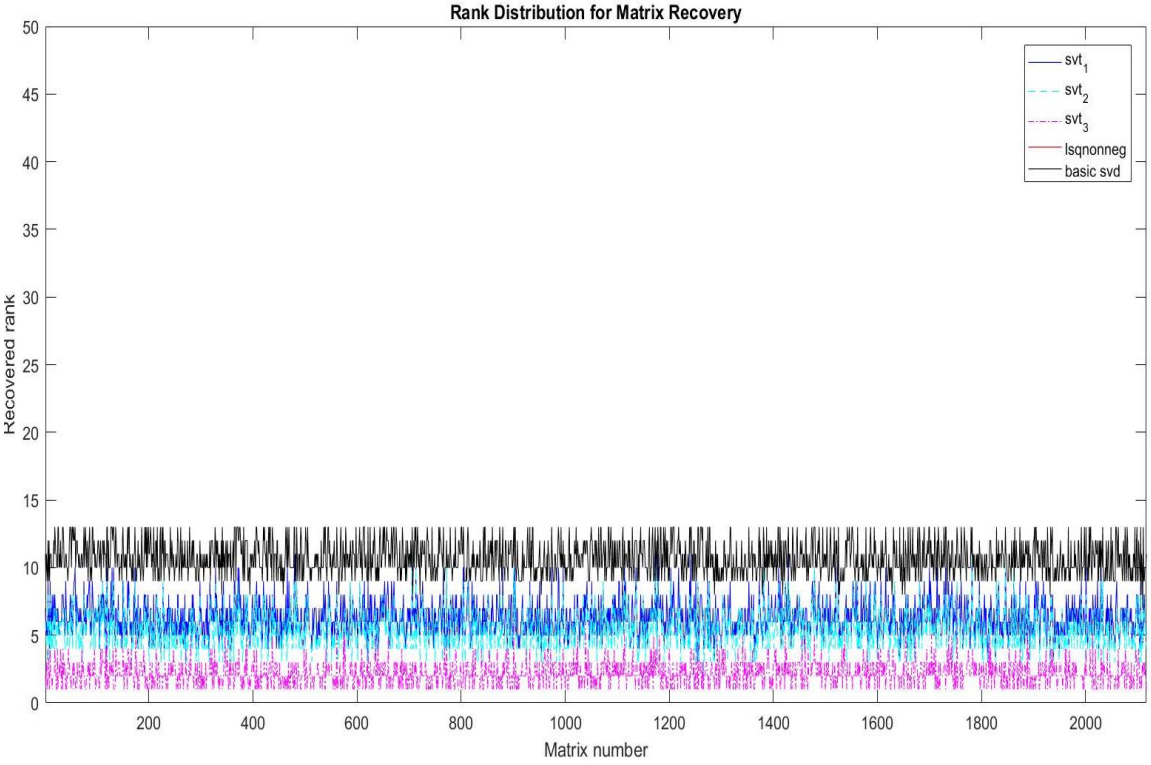


Figure 5.3.3 Rank Effectiveness for SVT Algorithm with General Convex Constraint

## 6. Conclusion

For network inference part, the generalized SVT algorithm works better under low-rank circumstances. By choosing larger threshold tau, the rank of recovered matrix converges to be low-rank. By choosing larger threshold tau, the recovered matrix tends to be more accurate. For full-rank situation, we can possibly transform it into a low-rank problem by adding a certain diagonal matrix to the original matrix and update all degree values for the algorithm implementation.

In summary, this thesis offered an alternative method to leverage public-available NBA game records into detailed game-wise sequential offense flow graphs. This thesis proposed an algorithm to simulate the real ball distribution network at the player level under low-rank constraints from the player ball-passing degree matrix. This thesis performed experiments on real NBA data to demonstrate the potential effectiveness of the proposed algorithm.

References

[1] Griffiths, S. (September 20, 2010). The Canadian who invented basketball. *BBC News*. Retrieved October 3, 2016 from http://www.bbc.com/news/world-us-canada-11348053.

[2] Jessop, A. (June 14, 2012). The Surge of the NBA's International Viewership and Popularity. *Forbes.com*. Retrieved October 3, 2016 from http://www.forbes.com/sites/aliciajessop/2012/06/14/the-surge-of-the-nbas-international-viewership-and-popularity.

[3] Harris, N. (May 1, 2012). REVEALED: The world's best paid teams, Man City close in on Barca and Real Madrid. *SportingIntelligence.com*. Retrieved October 3, 2016 from http://www.sportingintelligence.com/2012/05/01/revealed-the-worlds-best-paid-teams-man-city-close-in-on-barca-and-real-madrid-010501.

[4] Gaines, C. (May 20, 2015). The NBA is the highest-paying sports league in the world. *BusinessInsider.com*. Retrieved October 3, 2016 from http://www.businessinsider.com/sports-leagues-top-salaries-2015-5.

[5] Skinner, B. (2010). The Price of Anarchy in Basketball. *Journal of Quantitative Analysis in Sports*, *6*(1), 3. doi:10.2202/1559-0410.1217.

[6] Fewell, J. H., Armbruster, D., Ingraham, J., Petersen, A., & Waters, J. S. (2012). Basketball Teams as Strategic Networks. *PLoS ONE*, *7*(11), e47445. doi:10.1371/journal.pone.0047445.

[7] Basketball: Rules: Players. *BBC Sport*. Retrieved October 5, 2016 from http://news.bbc.co.uk/sportacademy/bsp/hi/basketball/rules/players/html/default.stm.

[8] Bonsor, K. (March 10, 2003). How Basketball Works. *Howstuffworks.com*. Retrieved October 5, 2016 from http://entertainment.howstuffworks.com/basketball2.htm.

[9] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Cambridge, MA: Morgan Kaufmann Publisher.

[10] Sumathi, S., & Sivanandam, S. N. (2006). *Introduction to Data Mining and its Applications*. Berlin: Springer.

[11] Ackoff, R. L. (1989). From Data to Wisdom. *Journal of Applied Systems Analysis, 16*(1), 3-9.

[12] Cohan, F. M. (2012). Science Needs More Moneyball. *American Scientist*, *100*(3), 182. doi:10.1511/2012.96.182

[13] Schumaker, R. P., Solieman, O. K., & Chen, H. (2010). Sports Knowledge Management and Data Mining. *Annual Review of Information Science and Technology*, *44*(1), 115-157.

[14] Basketball Possession. *Sporting Charts*. Retrieved Aug 28, 2016 from http://www.sportingcharts.com/dictionary/nba/possession.aspx

[15] Kang, K. (2014). Estimation of NBA players' offense/defense ratings through shrinkage estimation.

[16] Sisneros, R., & Moer, M. V. (2013). Expanding Plus-Minus for Visual and Statistical Analysis of NBA Box-Score Data. In *The 1st Workshop on Sports Data Visualization*. *IEEE*.

[17] Hwang, D. (2012). Forecasting NBA Player Performance using a Weibull-Gamma Statistical Timing Model. In *MIT Sloan Sports Analytics Conference*.

[18] Piette, J., Anand, S., & Pham, L. (2011). Evaluating Basketball Player Performance via Statistical Network Modeling. In *MIT Sloan Sports Analytics Conference*.

[19] Skinner, B., & Guy, S. J. (2015). A Method for Using Player Tracking Data in Basketball to Learn Player Skills and Predict Team Performance. *PLoS ONE*, *10*(9), e0136393. doi:10.1371/journal.pone.0136393.

[20] Uriegas, E. (September 9, 2015). stats.nba.com Endpoint Documentation. Retrieved Nov 7, 2016 from http://github.com/seemethere/nba_py/wiki/stats.nba.com-Endpoint-Documentation

[21] Reitz, K. (2015). Requests: HTTP for Humans. Retrieved Nov 9, 2016, from http://docs.python-requests.org/en/master/.

[22] Mitchell, R. (2015). Web scraping with Python: collecting data from the modern web. "O'Reilly Media, Inc.".

[23] Smedt, T. D., & Daelemans, W. (2012). Pattern for python. *Journal of Machine Learning Research*, *13*(Jun), 2063-2067.

[24] BASKETBALL-REFERENCE.COM. *NBA League Index*. Retrieved Feb 3, 2017 from http://www.basketball-reference.com/leagues/

[25] Cai, J. F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, *20*(4), 1956-1982.

[26] Recht, B., Fazel, M., & Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, *52*(3), 471-501.

# APPENDIX A

## 2014-2016 REGULAR SEASON GAMEBOOK

# APPENDIX B

## 2002-2016 TEAM ROSTER STATISTICS

# APPENDIX C

## 2014-2016 PLAYER PASSING TRACKING DATA