COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS




# NODE CENTRALITIES IN BIPARTITE NETWORKS

DIPLOMA THESIS

2021 Bc. Ján MAJHER

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

# NODE CENTRALITIES IN BIPARTITE NETWORKS

## DIPLOMA THESIS

| | |
|---|---|
| Study programme: | Mathematics of Economy, Finance and Modelling |
| Field of study: | 1113 Mathematics |
| Department: | Department of Applied Mathematics and Statistics |
| Supervisor: | doc. RNDr. Mgr. Beáta Stehlíková, PhD. |

Bratislava 2021 **Bc. Ján MAJHER**

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

# CENTRALITY VRCHOLOV V BIPARTITNÝCH SIEŤACH

## DIPLOMOVÁ PRÁCA

| | |
|---|---|
| Študijný program: | Ekonomicko-finančná matematika a modelovanie |
| Študijný odbor: | 1113 Matematika |
| Školiace pracovisko: | Katedra aplikovanej matematiky a štatistiky |
| Vedúci práce: | doc. RNDr. Mgr. Beáta Stehlíková, PhD. |

Bratislava 2021 **Bc. Ján MAJHER**






Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

# THESIS ASSIGNMENT

| | |
|---|---|
| **Name and Surname:** | Bc. Ján Majher |
| **Study programme:** | Mathematical Economics, Finance and Modelling (Single degree study, master II. deg., full time form) |
| **Field of Study:** | Mathematics |
| **Type of Thesis:** | Diploma Thesis |
| **Language of Thesis:** | English |
| **Secondary language:** | Slovak |

**Title:** Node centralities in bipartite networks

**Annotation:** Bipartite networks contain two groups of nodes, while the edges connect only nodes from different groups. As examples we note a network with nodes denoting actors and movies (an edge means that the actor starred in the movie), or people and social events (for example a well known "southern woman network"). These networks can be transformed into a form, in which all the nodes are of the same type. For example the "southern woman network" can be transformed into a network, where an edge connects women who both attanded some event; eventually we may assign the edge a weight equal to number of such events. If we are interested in node centralities, we may use centrality measures for the transformed network. Another possibility is to consider the original network and to use centrality measures proposed for this special kind of networks. In the master thesis we concentrate on the latter approach.

| | |
|---|---|
| **Supervisor:** | doc. RNDr. Mgr. Beáta Stehlíková, PhD. |
| **Department:** | FMFI.KAMŠ - Department of Applied Mathematics and Statistics |
| **Head of department:** | prof. RNDr. Marek Fila, DrSc. |
| **Assigned:** | 17.12.2019 |
| **Approved:** | 21.12.2019 |

prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme

.....
Student

.....
Supervisor






Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

# ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bc. Ján Majher
**Študijný program:** ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
**Študijný odbor:** matematika
**Typ záverečnej práce:** diplomová
**Jazyk záverečnej práce:** anglický
**Sekundárny jazyk:** slovenský

**Názov:** Node centralities in bipartite networks
*Centrality vrcholov v bipartitných sieťach*

**Anotácia:** Bipartitné siete obsahujú dve skupiny vrcholov, pričom hrany spájajú iba vrcholy z rôznych skupín. Príkladom môže byť sieť s vrcholmi znázorňujúcimi hercov a filmy (hrana znamená, že herec hral v danom filme), ľudí a spoločenské udalosti (napríklad známa sieť "southern woman network"). Tieto siete sa dajú previesť na tvar, v ktorom sú všetky vrcholy rovnakého typu. Napríklad zo siete 'southern woman network' môže vytvoriť takú, v ktorej hrana spája ženy, ktoré sa spolu zúčastnili nejakej akcie, prípadne jej priradiť váhu rovnú počtu takýchto spoločných akcií. V prípade, že chceme zisťovať centralitu vrcholov, môžeme použiť miery centrality pre takto trasformovanú sieť. Inou možnosťou je využiť pôvodnú sieť a použiť miery centrality navrhnuté pre tento špeciálny typ sietí. V diplomovej práci sa zameriame na tento prístup.

**Vedúci:** doc. RNDr. Mgr. Beáta Stehlíková, PhD.
**Katedra:** FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
**Vedúci katedry:** prof. RNDr. Marek Fila, DrSc.

**Dátum zadania:** 17.12.2019

**Dátum schválenia:** 21.12.2019 prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
vedúci práce

**Acknowledgement** I am truly indebted to my supervisor, doc. RNDr. Mgr. Beáta Stehlíková, PhD., for her guidance and extensive support during my work on this thesis.

Many thanks and appreciation also goes to my family and friends for their support.

# Abstract

MAJHER, Ján: Node centralities in bipartite networks [Diploma Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Mgr. Beáta Stehlíková, PhD., Bratislava, 2021, 59 pg.

Bipartite network is a type of data structure that offers the possibility to model relationships between two types of entities. Centrality measures allow us to identify important network members. This analysis is also referred to as ranking of nodes. Aim of the thesis is to model data from uncommon areas, using bipartite networks, with hopes of obtaining satisfactory results. In the first part, we introduce bipartite graphs as mathematical representation of networks and look into the properties and motivation behind various ranking algorithms. Then we address usage of bipartite networks in the field of sport analysis, where we will identify most valuable NBA players, while utilizing our developed ranking algorithm. Last chapter is dedicated do a machine learning based recommendation model for bipartite networks. Using this approach, we will help students in deciding which courses to enroll to.

**Keywords:** Bipartite networks, Node centralities, Ranking algorithms, Network analysis in sports, Recommender systems

# Abstrakt v štátnom jazyku

MAJHER, Ján: Centrality vrcholov v bipartitných sieťach [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Mgr. Beáta Stehlíková, PhD., Bratislava, 2021, 59 str.

Bipartitná sieť je typ dátovej štruktúry, ktorý poskytuje možnosť modelovania vzťahu medzi dvoma entitami. Miery centrality nám umožňujú identifikovať dôležitých členov siete. Táto analýza sa tiež nazýva ohodnocovanie uzlov. Cieľom práce je modelovať dáta z neobvyklých oblastí použitím bipartitných sieti, za úmyslom získania uspokojivých výsledkov. V prvej časti predstavíme bipartitné grafy ako matematickú reprezentáciu sieti a pozrieme sa na vlastnosti a motiváciu rôznych rankingových algoritmov. Potom sa zameriame na použitie bipartitných sieti v oblasti športu, kde budeme hľadať najužitočnejších hráčov v NBA, s použitím nami vytvoreného rankingového algoritmu. Posledná kapitola bude venovaná odporúčaciemu systému v bipartitných sieťach na báze strojového učenia. Použitím daného postupu budeme pomáhať študentom pri výbere predmetov na zapísanie.

**Kľúčové slová:** Bipartitné siete, Centralita vrcholov, Rankingové algoritmy, Analýza sieti v športe, Odporúčacie systémy

# Contents

**Introduction** **11**

**1 Bipartite networks** **13**

1.1 Bipartite graph . . . 13

1.2 Southern women network . . . 15

1.3 Centralities in bipartite network . . . 17

1.3.1 Degree centrality . . . 17

1.3.2 Eigenvector centrality . . . 18

1.4 Ranking algorithms . . . 20

1.4.1 HITS . . . 21

1.4.2 BiRank . . . 22

1.4.3 Other algorithms . . . 23

**2 Ranking NBA players** **25**

2.1 Network analysis in sports . . . 25

2.2 Players and lineups in bipartite network . . . 26

2.2.1 Dataset overview . . . 26

2.2.2 Graph analysis . . . 27

2.3 Introducing AEW ranking algorithm . . . 29

2.3.1 AEW's design . . . 29

2.3.2 Convergence Analysis . . . 30

2.4 Player rankings . . . 31

2.4.1 Offensive Rating . . . 33

2.4.2 Points per 48 minutes . . . 36

2.4.3 Defensive Rating . . . 37

2.4.4 Net Rating . . . 38

**3 Recommender systems** **40**

3.1 Bipartite networks in recommendation models . . . 40

3.2 Music recommendation . . . 41

3.2.1 Last.fm dataset . . . 41

3.2.2 Network construction . . . . . 42
3.2.3 Preference ranking . . . . . 43
3.2.4 Binary classification . . . . . 45
3.2.5 Evaluation and Results . . . . . 46
3.3 Course recommendation . . . . . 48
3.3.1 Enrollment data . . . . . 48
3.3.2 Ranking courses . . . . . 50
3.3.3 Model development . . . . . 50
3.3.4 Model deployment . . . . . 51

**Conclusion** **55**

**Bibliography** **57**

**A Players ranking** **60**

**B Course recommendation** **64**

# Introduction

With growing availability and volume of data, data-driven approaches are emerging as highly beneficial and even indispensable in both the research and business world. There are many techniques of data modeling which stem from nature and structure of the data and from the desired outcome. One of them are networks, used mainly for study of the relationships within the data.

Network analysis is a compelling methodological approach, as network structure can be a rich source of information about any content we can represent as such. This thesis deals with a specific type of networks, bipartite networks. Bipartite network is a ubiquitous data structure that can model relationships between two types of entities. Common examples include network of users and items, network of social affiliation between women and events or network of patients and their doctors.

Main focus of the thesis will be node centralities, which is the most widely used property in the network structures. Centrality in the networks is a well-studied and popular topic which captures essential information about the data. Goal of thesis will be to focus on less known fields such as network analysis applications in sports phenomena and utilization of bipartite networks in machine learning based recommender systems.

First chapter is dedicated to basic concepts of bipartite networks. With use of publication [15], we will introduce bipartite graphs as mathematical representation of the networks and present their properties. Afterwards, based on knowledge provided in [5] and [3], we will take closer look at centralities and ranking algorithms like HITS [14] and BiRank [8]. Presenting these algorithms in detail is crucial for the next chapter, where we propose our modification of a ranking algorithm.

Second chapter focuses on sports applications. In the beginning we overview the magnitude of the research already conducted and present a preview of new approaches that will be presented in the thesis. Core of the chapter will revolve around the desire to rank NBA players, in order to find the most valuable ones. Topic was introduced by authors in [17], however, we will approach it in a different manner. We start with presenting a corresponding bipartite network. Then we will propose our methodology for dealing with the particular problem, which is based on the creation of a new ranking algorithm, AEW. In the end, we will showcase results and compare them with official

quality measures, used in the sport analyst community.

Last chapter is devoted to bipartite networks as an instrument for user recommendations presented in [13]. This approach is based on machine learning workflow and we start with a summary of the whole procedure. Next, we go into detail of each step using a well known recommendation music dataset. We will explain the role of centralities in such recommender system and present possible enhancements to the original approach. After presenting results of the music application, which serve as a test of workflow, we will move to a prototypical course recommender system. Our aim is to implement procedure in real-life example. Consequently, this part will focus on deployment and interpretation of created model.

# 1 Bipartite networks

A network is an abstract structure consisting of nodes and links. Network analysis plays an important role in various domains such as science, engineering and commerce, as it can represent different types of information. Real-life information may have specific properties, hence specific types of networks are eligible as representation. Bipartite networks model a relationship between two different classes of nodes, where each link connects a node from one class with a node from the other class.

Consider for example social gathering, where one class contains women and the other events. In this case, the relationship of nodes is given by attendance of events. Or YouTube, where videos and users form bipartite relationship where link indicates a viewing action. Most fundamental indicator in any network is centrality. There are many types of centrality measures that identify the importance of nodes based on different properties.

Before moving to analysis of particular problems, we will devote the first chapter of the thesis to introduction of bipartite networks and different types of node centralities. First, we will focus on basic concepts of bipartite networks, provide examples and look into terminology, which is diverse throughout scholarly literature. In addition, we will compare bipartite networks with ordinary unipartite networks. Subsequently, we will inquire about different issues when dealing with a concept centrality in bipartite networks. We will be particularly interested in ranking algorithms of iterative nature, that are a core component in real-life applications presented in later chapters.

## 1.1 Bipartite graph

As we mentioned before, bipartite networks are modelling the relationship between two classes of nodes using links. That is why on many occasions they are called affiliation networks. This special type of network comes with its own mathematical representation widely known as bipartite graphs. Frequently these graph are called two-mode graphs. The terminology of modes refers to the number of different classes of nodes.

As a consequence, a bipartite graph consists of two sets of vertices, representing the nodes, and a set of edges, representing links and connecting vertices. According to [15] and [8], a bipartite graph is usually denoted as follows:

$$G = (U, V, E)$$

where $U = \{u_1, u_2, \ldots, u_p\}$ and $V = \{v_1, v_2, \ldots, v_r\}$ are two disjoint and independent sets of vertices with cardinalities $|U| = p$ and $|V| = r$, where cardinality denotes the number of elements in a set. $E$ is set of edges, such that every edge connects a vertex from set $U$ to solely one vertex from set $V$:

$$E = \{(u, v)|u \in U, v \in V\}.$$

We use $u_i$ to denote the $i$-th vertex in $U$ and $v_j$ to denote $j$-th vertex in $V$, where $1 \leq i \leq |U|$ and $1 \leq j \leq |V|$. Edges carry non-negative weights, $w_{ij}$, to represent strength of relationship between these vertices .

Although bipartivity gives special structure to a graph, most of the properties remain the same as in an unipartite graph. Unipartite graph, or one-mode graph, represents the usual case with a single class of vertices. Bipartite graphs can also have the character of weighted graphs, directed graphs, etc.

Additionally, we can also represent graph by a matrix. In a more ordinary case of unipartite graph an adjacency matrix is used, where columns and rows refer to the same set of entries. In the case of bipartite graph, they can be represented with a matrix known as biadjacency matrix. In our thesis biadjacency matrix will acquire the form of $W = [w_{ij}]$. Term two-modes now refers to two different kinds of entries in the rows and columns. This form of matrix is applicable if bipartite graph is not directed, hence only one information about its edge is required. In such a matrix, element $w_{ij}$ gives us the weight of the edge that connects vertex $u_i$ to $v_i$. Hence if $u_i$ and $v_i$ are not connected, the value of $w_{ij}$ is zero.

Biadjacency matrix can be quite easily transformed into adjacency matrix, which is equivalent transforming a two-mode graph into a one-mode graph. An example will be shown in the next part. This will help us understand transformation better as we will examine how this transformation changes information value of the network.

## 1.2 Southern women network

With basic concepts of how we can examine bipartite networks with graph theory settled, we will now take a closer look at one particular example of a bipartite network. In the introduction of this chapter we mentioned as an example of node classes of women and events. This network is known as the Southern women network, which is a well known dataset. An inspirational overview is provided by Stephen P. Borgatti and Daniel S. Halgin in in [3] . It portrays which woman attended which social event in a small southern town. Visualisation of this network is displayed in figure 1.

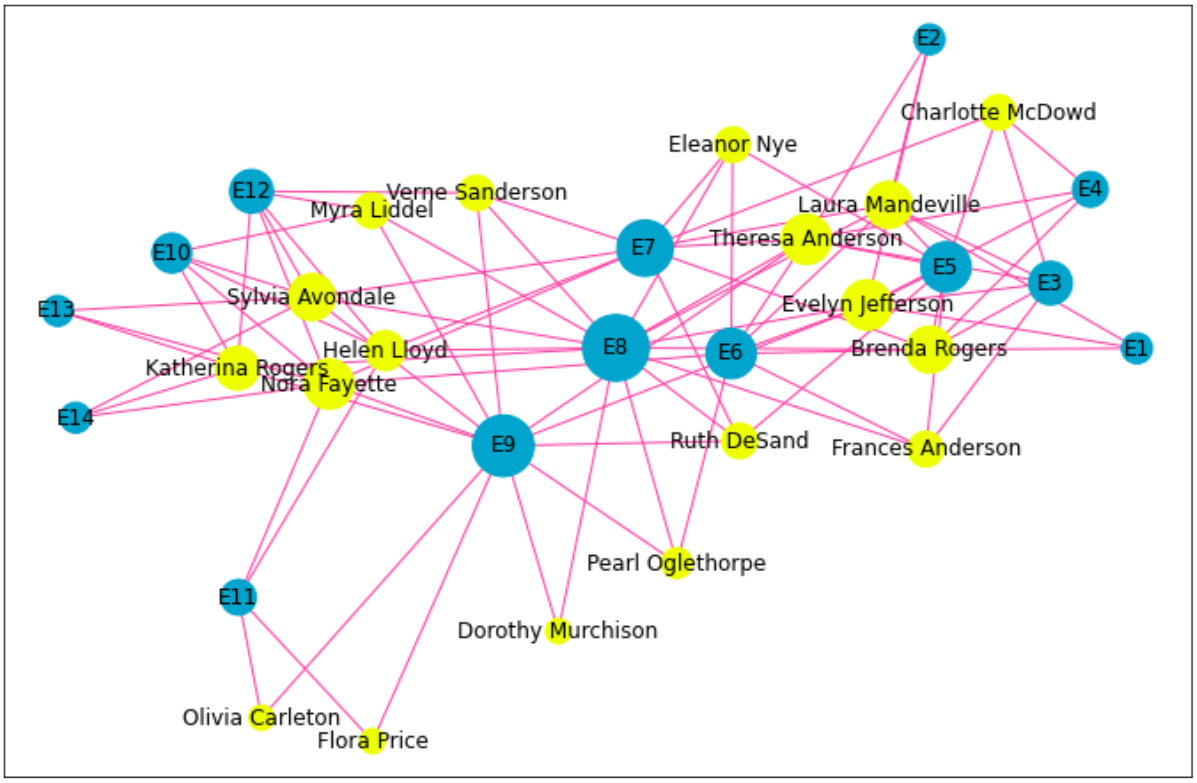



Figure 1: Southern women network

Before, we mentioned the term of affiliation network. Southern women network is a great example social network, in which we look at the relationship in a way as membership or participation. Consequently, biadjancy matrix carries information about participation of women in events. The appertaining biadjacency matrix is shown in figure 2.

Columns refer to events, rows to women and number 1 indicates what event was attended by particular woman. Hence the matrix represents affiliation. Occasionally, we might be interested in the nature of relationship within one class of nodes instead

| | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | E12 | E13 | E14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **EVELYN** | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **LAURA** | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **THERESA** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **BRENDA** | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **CHARLOTTE** | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **FRANCES** | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **ELEANOR** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **PEARL** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **RUTH** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **VERNE** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| **MYRNA** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| **KATHERINE** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| **SYLVIA** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| **NORA** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| **HELEN** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| **DOROTHY** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **OLIVIA** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **FLORA** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

Figure 2: Biadjacency matrix

of between classes. In the case of the Southern women network, co-affiliation between women means attendance of the same event. As presented in [5], this can be achieved by obtaining adjacency matrix using following equations:

$$A_p = WW^T$$

and

$$A_r = W^T W.$$

First matrix $A_p$ is the co-affiliation matrix of woman displayed in figure 3. Matrix $A_r$ is option for nodes representing events. Indices $p$, $r$ are cardinalities of sets of women and events, respectively, while also being equal to measures of square matrices $A_p$ and $A_r$.

Biadjacency matrix $W$ uniquely determines both $A_p$ and $A_r$, while reverse statement is not valid. Consequently one can study bipartite networks using this transformation and look at the properties of relation in both one-mode and two-mode style.

| | EVE | LAU | THE | BRE | CHA | FRA | ELE | PEA | RUT | VER | MYR | KAT | SYL | NOR | HEL | DOR | OLI | FLO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EVELYN | 8 | 6 | 7 | 6 | 3 | 4 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 1 |
| LAURA | 6 | 7 | 6 | 6 | 3 | 4 | 4 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 |
| THERESA | 7 | 6 | 8 | 6 | 4 | 4 | 4 | 3 | 4 | 3 | 2 | 2 | 3 | 3 | 2 | 2 | 1 | 1 |
| BRENDA | 6 | 6 | 6 | 7 | 4 | 4 | 4 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 |
| CHARLOTTE | 3 | 3 | 4 | 4 | 4 | 2 | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| FRANCES | 4 | 4 | 4 | 4 | 2 | 4 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| ELEANOR | 3 | 4 | 4 | 4 | 2 | 3 | 4 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 |
| PEARL | 3 | 2 | 3 | 2 | 0 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 1 |
| RUTH | 3 | 3 | 4 | 3 | 2 | 2 | 3 | 2 | 4 | 3 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 1 |
| VERNE | 2 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 3 | 3 | 4 | 3 | 3 | 2 | 1 | 1 |
| MYRNA | 2 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 3 | 3 | 2 | 1 | 1 |
| KATHERINE | 2 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 3 | 4 | 6 | 6 | 5 | 3 | 2 | 1 | 1 |
| SYLVIA | 2 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 4 | 6 | 7 | 6 | 4 | 2 | 1 | 1 |
| NORA | 2 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 5 | 6 | 8 | 4 | 1 | 2 | 2 |
| HELEN | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 5 | 1 | 1 | 1 |
| DOROTHY | 2 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 |
| OLIVIA | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 2 |
| FLORA | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 2 |

Figure 3: Adjacency matrix for woman

## 1.3 Centralities in bipartite network

Most fundamental qualitative measurement in network analysis is centrality. Centralities give value, or rank, to a node based on its characteristics that are essential for particular centrality. There are several pivotal types of centralities:

- Degree
- Betweenness
- Closeness
- Eigenvector

Some of these focal centralities are foundation for more advanced and newly developed types that take crucial part in this thesis. Therefore, next part will be dedicated to their detailed description. We will focus on their properties and how they adapt in a bipartite network in comparison to an ordinary network.

### 1.3.1 Degree centrality

Degree centrality is one of the simplest centralities. In an ordinary network, the degree of a node is given by the number of edges connected to it. In bipartite networks

the proposition remains the same. For instance, in the Southern women network, degree centrality for a woman is the quantity of the events she attended. For an event it is the number of women that attended that particular event.

As mentioned in [3] and [19], on many occasions, degree centrality is being normalized. In a bipartite network, the maximum degree of a node is always equal to the cardinality of the opposing class. Thus we must apply separate normalizations depending on the belongingness as follows:

$$d_i^* = \frac{d_i}{r}, \text{for } i \in U$$

$$d_j^* = \frac{d_j}{p}, \text{for } j \in V$$

where $d_i$ and $d_j$ are degree centralities of nodes $u_i$ and $v_j$, respectively.

The key benefit of shown normalization is the assessment of relative centrality among nodes within the same class, but also between classes. In context of the Southern women network, we might say that a woman is more gregarious than an event is attractive.

### 1.3.2 Eigenvector centrality

Basic idea for eigenvector centrality in a one-mode network, is that centrality of the node should be proportional to the strength of node's connections within a network. In older literature, this centrality was also referred to as the popularity index. Name eigenvector comes from the fact that we are actually utilizing the principal eigenvector of a matrix. Principal eigenvector is such that it corresponds to the largest eigenvalue of the matrix. In case of an unipartite network it is principal eigenvector of adjacency matrix. Standard eigenvector-eigenvalue problem is expressed by the following equation (In [5]):

$$Ae = \lambda e$$

where $A$ is an adjacency matrix, $\lambda$ is the largest eigenvalue and $e$ is a vector of centrality scores. Hence the eigenvector centrality of the $i$-th node is the $i$-th position in vector $e$.

Suppose we have bipartite network and appertaining biadjacency matrix $W$. I will denote eigenvector centrality scores $e^U$ and $e^V$ for nodes in sets $U$ and $V$, respectively. One can find these scores by analysing transformed one-mode matrices, but as presented

in [5], there is also an approach of using biadjacency matrices. Putting them together we can use the following equation

$$\lambda \begin{bmatrix} e^U \\ e^V \end{bmatrix} = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix} \begin{bmatrix} e^U \\ e^V \end{bmatrix}$$

and recover demanded eigenvector scores.

To demonstrate both degree and eigenvector centrality, we can see results in figure 4 and 5 for nodes modelling events and women, respectively. In addition we attached a column showing the number of connections, i.e. degree centrality before normalization. We can observe that eigenvector centrality is closely correlated with degree centrality. This is a consequence of the nature of the network. There are no weights, meaning there is not any indicator of connection quality, therefore quantity of connections is only indicator. Even looking back at figure 1 where node size is scaled according to degree, we can see those are most popular nodes. In our example, events E8, E9 and E7 were most prominent.

| Node | Number of connections | Normalized degree centrality | Normalized eigenvector centrality |
|---|---|---|---|
| E8 | 14 | 0.78 | 0.51 |
| E7 | 10 | 0.56 | 0.38 |
| E9 | 12 | 0.67 | 0.38 |
| E6 | 8 | 0.44 | 0.33 |
| E5 | 8 | 0.44 | 0.32 |
| E3 | 6 | 0.33 | 0.25 |
| E12 | 6 | 0.33 | 0.20 |
| E4 | 4 | 0.22 | 0.18 |
| E10 | 5 | 0.28 | 0.17 |
| E2 | 3 | 0.17 | 0.15 |
| E1 | 3 | 0.17 | 0.14 |
| E13 | 3 | 0.17 | 0.11 |
| E14 | 3 | 0.17 | 0.11 |
| E11 | 4 | 0.22 | 0.09 |

Figure 4: Centralities of events

| Node | Number of connections | Normalized degree centrality | Normalized eigenvector centrality |
|---|---|---|---|
| Theresa Anderson | 8 | 0.57 | 0.37 |
| Evelyn Jefferson | 8 | 0.57 | 0.33 |
| Brenda Rogers | 7 | 0.5 | 0.31 |
| Laura Mandeville | 7 | 0.5 | 0.31 |
| Sylvia Avondale | 7 | 0.5 | 0.28 |
| Nora Fayette | 8 | 0.57 | 0.26 |
| Ruth DeSand | 4 | 0.29 | 0.24 |
| Eleanor Nye | 4 | 0.29 | 0.23 |
| Katherina Rogers | 6 | 0.43 | 0.22 |
| Verne Sanderson | 4 | 0.29 | 0.22 |
| Frances Anderson | 4 | 0.29 | 0.21 |
| Helen Lloyd | 5 | 0.36 | 0.20 |
| Myra Liddel | 4 | 0.29 | 0.19 |
| Pearl Oglethorpe | 3 | 0.21 | 0.18 |
| Charlotte McDowd | 4 | 0.29 | 0.17 |
| Dorothy Murchison | 2 | 0.14 | 0.13 |
| Olivia Carleton | 2 | 0.14 | 0.07 |
| Flora Price | 2 | 0.14 | 0.07 |

Figure 5: Centralities of women

Calculation of centralities and plotting of network (Figure 1) was done using python. We were particularly using library Networkx, which is the most popular library for network analysis. In addition it contains Southern women network dataset. Libraries Numpy and Matplotlib were used for mathematical operations and plotting, respectively.

## 1.4 Ranking algorithms

Based on centrality score we are able to rank nodes. Ranking of nodes means how significant they are in comparison to each other. Ranking algorithms are the main instrument in this thesis. This section will be an overview of several of them. We will clarify the idea behind their implementation, and how to choose the right ranking algorithm for particular a problem.

All rankings algorithms are in certain way modifications of the fundamental centralities. Most famous is the PageRank algorithm. PageRank was developed by Larry Page and Sergey Brin and is utilized in Google search engine. When a user searches for information, the search engine employs the algorithm to rank pages, and that gives them an

order in which they are displayed to the user. However, PageRank is considered mainly a one-mode network algorithm and using it on a bipartite network would not be efficient, because of the possibility of information loss. Therefore, we will firstly introduce HITS algorithm dedicated to the two-mode networks, which is designed in a similar way to PageRank.

### 1.4.1 HITS

HITS stands for Hyperlinked-Induced Topic Search. Jon Kleinberg developed and published it in 1999 [14]. It was fundamentally designed for web networks. Biparitity comes from the original application. Main idea is to rank authorities and hubs, which are two different classes. Authority is good if there are many decent hubs linked to it, and a hub is decent if there are many good authorities linked to it. Notice the similarity with the eigenvector centrality. Centrality score depends on the quality and on the quantity of connections.

HITS, as other ranking algorithms, is an iterative process. Meaning, when applied on a bipartite graph, the algorithm starts at a the certain initial score of the vertices and builds its way up to the final result. As a consequence of the algorithm motivation, update rule will be based on a mutually-reinforcing relationship. This naturally implemented into the iterative process that refines each vertex's score as the sum of contribution from it's connected vertices:

$$u_i = \sum_{i=1}^{|U|} w_{ij} v_j; \qquad v_j = \sum_{j=1}^{|V|} w_{ij} u_i \tag{1.1}$$

which is in a matrix form expressed as

$$u = W^T v; \qquad v = Wu. \tag{1.2}$$

As this is an additive update rule, normalization is necessary to ensure convergence and stability. In HITS it is done after each update. Consequently, the algorithm will terminate and we obtain eventual ranking vectors. Formulation of the whole procedure is showcased in algorithm 1.

**Algorithm 1** HITS algorithm

**Input:** Biadjacency matrix $W$, initial vectors $u^0$, $v^0$;
**Output:** Ranking vectors $u$, $v$;

1: **while** Stopping criteria is not met **do**
2: $\quad u' = W^T v$
3: $\quad v' = Wu$
4: $\quad u = u'/|u'|$ ▷ Normalize
5: $\quad v = v'/|v'|$ ▷ Normalize
6: **end**
7: **return** $u$, $v$

### 1.4.2 BiRank

Another ranking algorithm we will introduce is BiRank, as many of its properties are essential for the modification presented in our work. BiRank is a modification of HITS. Similarly to HITS, it is also of an iterative nature and was introduced recently in 2016 in paper [8] . In addition, BiRank brings a new possibility to impact the final ranking vector with initial values of $u^0$, $v^0$ as we will show later.

Premises stay the same. Consider a bipartite network represented by biadjacency matrix $W$ and initial scoring vectors $u^0$ and $v^0$. In a similar way, we will express each vertex score as the sum of the contributions using equations 3.1. However, in this case, normalization is not exercised after execution of the update rule, but rather inside it. In BiRank method, authors adopted symmetric normalization scheme, of which main purpose is to smooth the edge weight buy the degree of its two connected vertices simultaneously:

$$u_i = \sum_{i=1}^{|U|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} v_j; \qquad v_j = \sum_{j=1}^{|V|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} u_i, \tag{1.3}$$

where $d_i$ and $d_j$ are weighted degrees of the vertices $u_i$ and $v_i$. Weighted degree of the vertex is the sum of the weights of its connected edges. This use of the symmetric normalization is the main characteristic of BiRank. It allows edges connected to the vertices with high weighted degrees to suppress their impact.

Now let us formulate 1.3 in a matrix form. As presented in [8], we will denote diagonal matrices $D_u = [d_i]$ and $D_v = [d_j]$, containing sum of weights of the edges connected to the vertices in $U$ and $V$, respectively. We will denote symmetrically normalized $W$:

$$S = (D_u)^{-1/2} W (D_v)^{-1/2}. \tag{1.4}$$

Hence update rule in a matrix form will take form:

$$u = S^T v; \qquad\qquad v = Su. \tag{1.5}$$

At the introduction of BiRank, we mentioned the possibility of using initial values $u^0$, $v^0$. This is for cases, where we wish to impact the final ranking values based on some prior knowledge or our preferences. Best way to illustrate use of these adjusted initial vectors is to show the procedure of BiRank algorithm:

**Algorithm 2** BiRank algorithm

**Input:** Weight matrix $W$, initial vectors $u^0$, $v^0$, hyper-parameters $\alpha$, $\beta$;
**Output:** Ranking vectors $u$, $v$;

1: Calculate $S$: $S = (D_u)^{-1/2} W (D_v)^{-1/2}$
2: **while** Stopping criteria is not met **do**
3: $\quad u = \alpha S^T v + (1 - \alpha) u^0$
4: $\quad v = \beta S u + (1 - \beta) v^0$
5: **end**
6: **return** $u$, $v$

There we can notice that new hyperparameters $\alpha$ and $\beta$ were denoted to control impact of the initial vectors. This way we are able to regularize convergence of the scoring vectors by choosing values of $\alpha$ or $\beta$.

### 1.4.3 Other algorithms

There are of course other ranking algorithms that share similar characteristics with BiRank and HITS, which was topic of discussion in [8]. Most cases properties change for the sake of better scoring results for a specific application.

BiRank varied from HITS not only by utilizing initial vector, but also by exercising different update rule. This is also the case for other algorithms, as they are suppressing or rather inciting certain structural characteristics of a network. Each of them have certain strengths and weaknesses. Key difference is form of the update matrix $S$. Note that for

HITS we can denote $S = W$, since there were not any adjustments. In table 1 we can observe a summary of such methods.

| **Algorithm** | **Update Matrix** |
|---|---|
| HITS | $S = W$; $S^T = W^T$ |
| BiRank | $S = (D_u)^{-1/2} W (D_v)^{-1/2}$; $S^T = (D_v)^{-1/2} W (D_u)^{-1/2}$ |
| Co-HITS | $S = W(D_v)^{-1}$; $S = W^T (D_u)^{-1}$ |
| BGER | $S = (D_u)^{-1} W$; $S^T = (D_v)^{-1} W^T$ |
| BGRM | $S = (D_u)^{-1} W (D_v)^{-1/2}$; $S^T = (D_v)^{-1/2} W (D_u)^{-1}$ |

Table 1: Different formulations of update matrix

In conclusion, different applications demand different approaches. Our goal may be to focus on quality of connection, in preference of quantity, which will be the case in the first application presented in the next chapter.

# 2 Ranking NBA players

Network analysis, as a methodological approach, is widely used in a variety of fields. This chapter will be dedicated to sports, a topic that is not that common as a subject of network analysis. Particularly, we will investigate centralities of basketball players within a bipartite network.

In basketball, particularly the NBA, there is a broad spectrum of statistics, from points scored, shooting efficiency, offensive and defensive ratings, etc. Goal of sport analysis in this thesis is to rank NBA players according to their team contribution. Specifically, by using statistics of lineups.

At the beginning of this chapter we will do an overview of utilization of networks analysis in sport applications, we encountered during research. Next we will introduce nature and purpose of our application. During this examination, we will outline inapplicability of studied ranking algorithms. Consequently, we will propose our modification of a ranking algorithm for a bipartite network and with properties that serve our application. At the end of this chapter, we will present results and findings.

## 2.1 Network analysis in sports

Although sport is a very popular and huge topic around the world, it has not received much attention in the scientific community. When it comes to benefiting from network analysis, authors in [11] and [4] presented usage of networks for effective scheduling of sports competitions, which is one possible domain. In 2017, article [20] was published that presents social networks analysis in sport research as an emerging paradigm. Authors conducted a systematic review of social network analysis applications in sports. This review classified six domains of social network applications in sports. One of those domains are affiliation networks, which are the topic of this chapter.

With regard to basketball, in [21] the author observed the effect of networks in NBA teams. Our application was inspired by Sohu Misra ([17]), who tried to determine who are the best players in the NBA using statistics of lineups. In paper, the problem was approached by transforming two-mode network into one-mode as we presented in 1 chapter and ranked players based on centralities within unipartite network. In this thesis, we will

keep network in two-mode state, and apply ranking a algorithm designed for bipartite networks.

## 2.2 Players and lineups in bipartite network

Basketball is a team sport where two teams, 5 vs 5 player lineups at the time, face each other in a competitive game. Objective is to, within a certain time period, score more points than the opponent, which can be accomplished by having good defence and efficient offence.

Most popular and default statistic, are points scored. In the NBA season 2019-20, James Harden led the league with 34.3 points scored on average per game. Even though points scored is a major aspect of how we can evaluate performance of a team, it is not the only aspect. There are different, whether basic or advanced, offensive and defensive statistics and are all available for different basketball entities such as a team, a player or a lineup.

Our goal is to use statistics that are foremost in capturing performance of a lineup and connect it to the player's importance for the team. Two classes in our bipartite network will therefore be represent players and lineups. If a player played in a particular lineup, there exists an edge connecting them and weight of edge is determined by the statistical attribute of particular lineup.

Motivation is following: If a lineup has good statistic, it performed well, and a player, that was a member or particular lineup, directly contributed to its success. Overall we would like to know how valuable a player is for the team. Therefore we aim to exercise a certain ranking algorithm, whose outcome will inform us about the player's value. In addition we will also keep track of individual statistical attributes, of which usage will be explained later in the chapter.

### 2.2.1 Dataset overview

Our main source of data are official NBA statistics websites [18], [1]. Different kinds of statistics are available for different seasons and season types. You can look into total volumes or per game averages of players, teams or lineups. Other traditional statistical attributes available than points scored are assists, rebounds, block or shooting efficiencies

for different types of shot attempts. Nowadays various advanced statistics like offensive rating, assists ratio or net rating are also available and several of them will be used on our applications.

We decided to choose data from NBA playoffs 2019/2020, which are more important than the regular season, and teams are expected to perform at their highest level. Therefore, we consider performance of players in this key part of the season.

As part of data preparation we omitted lineups with insignificant playing time. This was done in order to clean data from outliers. If a lineup has only a few minutes of playing time, its statistic as a measure of performance is not reliable. To consider someone's impact on the floor, a significant amount of time is a fair requirement. Analysis in this thesis considers 117 players and 59 lineups.

### 2.2.2 Graph analysis

In the NBA playoffs 8 teams from either western or eastern conference play with each other in best-of-seven series in conference quarterfinals, semifinals and finals, to ultimately face the best team from the other conference for the championship. On that account, some teams play more games than others. As a consequence, a player like LeBron James, whose team Los Angeles Lakers reached the NBA finals, is connected to 8 lineups. In comparison, Luka Doncic, whose team lost in the first play off round, only has connection to 2 lineups. As a consequence, we might come upon structures and situations in networks as showcased in figure 6.

Vertices represent the two players and their corresponding lineups, distinguished by colour. Players 1 is Enes Kanter, who is mentioned in one lineup only. However, this lineup was doing really well as the offensive rating is 140.8. Enes Kanter is well known throughout the league for his offensive accolade, what explains the high rating. On the other hand, there is player 2 Monte Morris, whose team played more games than Kanter's team. Therefore he has more connections but with worse offensive rating. In conclusion, degree of node is a misleading factor, and it does not portray the popularity of the node.

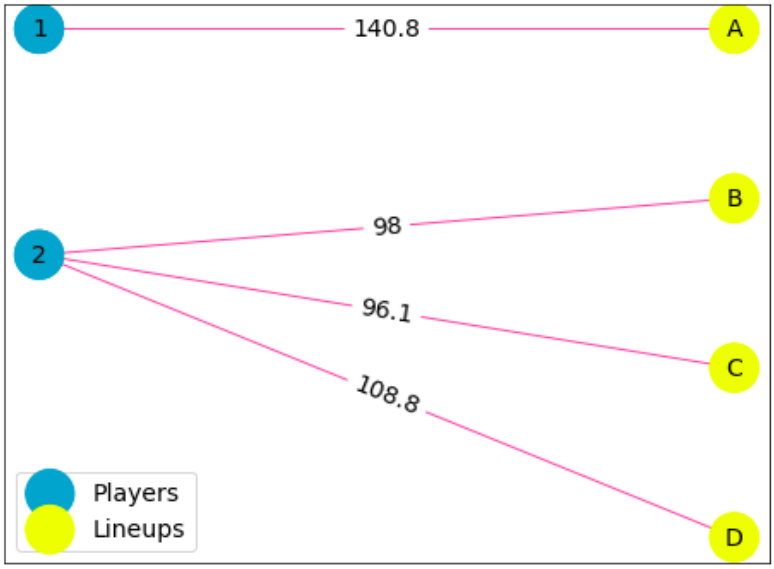



Figure 6: Example 1 of graph's structure

Another problem is difficulty to distinguish players within the same lineup. There might be a situation, where players only played together, hence they have the same connection within the network. But one must be better than the other, they should not be ranked exactly the same. Example of this particular structure of graph displayed in figure 7 where three vertices (players) have exactly the same connections with other class of vertices (lineups).

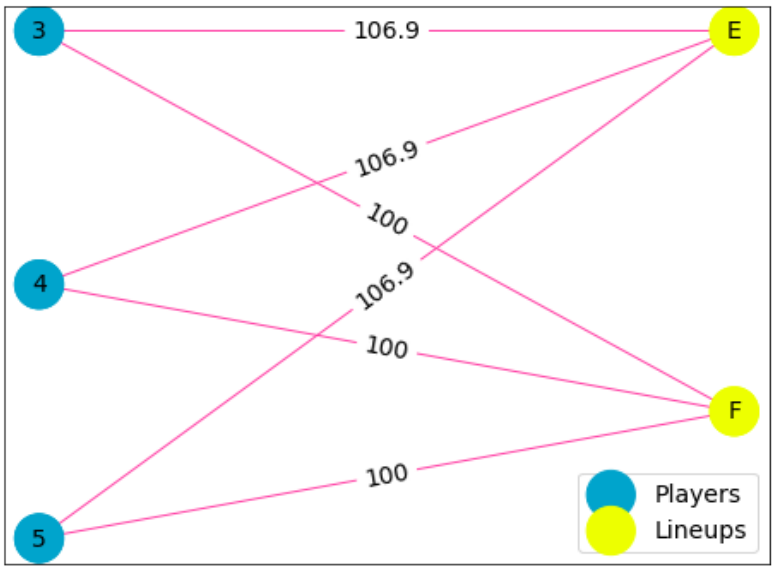



Figure 7: Example 2 of graph's structure

To solve this problem we will utilize the initial vector. In order to avoid players having the same ranking score, we will choose initial vector's values based on their individual performance metrics. For example, offensive rating will be used as our main metric of team offensive performance. Points per game for each player might be reasonable information to put inside the initial vector, on account of giving the algorithm prior knowledge of the player's offensive quality.

For reasons presented, it is not clear whether the ranking algorithms presented in chapter 1 are suitable for our problem. As was mentioned in section 1.4.3, update rule, which is formulated in matrix $S$, is granting algorithm possibility to capture properties of network that are essential for effectiveness of ranking in different applications. In our case, we are more interested in quality of connections rather than quantity.

## 2.3 Introducing AEW ranking algorithm

As we mentioned in the previous part, the structure of our network makes it difficult to use known ranking algorithms presented in 1.4. Therefore, we propose our own algorithm, which is modification of the HITS algorithm with incorporation of the BiRank's query vector component.

### 2.3.1 AEW's design

In a similar way as HITS, we will express each vertex score as the sum of contribution like in equations 3.1. Then in order to weaken impact of degree centrality we denote diagonal matrices $K_u = [k_i]$ and $K_v = [k_j]$, containing a number of edges connected to vertices in $U$ and $V$, respectively. We will use these matrices in the update matrix, which will then acquire the following form:

$$S = K_u^{-1}W, \qquad S^T = K_v^{-1}W^T. \tag{2.1}$$

Multiplying Biadjacency matrix $W$ with $K_u, K_v$ suppresses weights of the edges according to degree centrality of particular vertex. This is the reason why we named our proposed algorithm AEW, as in average edge weight. Consequently, we obtain:

$$u' = S^T v, \qquad v' = Su. \tag{2.2}$$

After the update rule is exercised, we will normalize vectors as in algorithm 1. In the next step when we will use hyperparameters $\alpha$, $\beta$. As result of normalization, both adjusted initial vector and updated ranking vector have the same scale:

$$\begin{aligned} u &= \alpha u' + (1-\alpha)u^0, \\ v &= \beta v' + (1-\beta)v^0. \end{aligned} \tag{2.3}$$

Whole procedure can be observed in algorithm 3.

**Algorithm 3** AEW algorithm

**Input:** Biadjacency matrix $W$, (adjusted) initial vectors $u^0$, $v^0$, hyperparameters $\alpha$, $\beta$;
**Output:** Ranking vectors $u$, $v$;

1: Calculate matrix $S^T$: $S^T = K_v^{-1}W^T$
2: Calculate matrix $S$: $S = K_u^{-1}W$
3: **while** Stopping criteria is not met **do**
4: $\quad u' = S^T v$
5: $\quad v' = Su$
6: $\quad u' = u'/|u'|$ ▷ Normalize
7: $\quad v' = v'/|v'|$ ▷ Normalize
8: $\quad u = \alpha u' + (1-\alpha)u^0$
9: $\quad v = \beta v' + (1-\beta)v^0$
10: **end**
11: **return** $u$, $v$

### 2.3.2 Convergence Analysis

AEW was created as modification of HITS, with utilization of the initial vector component, what was introduced in BiRank algorithm. Normalization steps (algorithm 3 line 6) and (algorithm 3 line 7) replicated from HITS, serve as certainty of converge and stability.

Iterating procedure is terminated when stopping criterion is met, meaning when the scoring vector's new update does not change from the previous iteration. In our implementation of the algorithm, which was used for ranking of the NBA players, we decided to use absolute value norm. Stopping criterion was met, if the difference of the last two consecutive score vectors within $L1$-norm was smaller than predetermined tolerance value.

Consider scoring vectors $u, v$ and tolerance value $t$. Algorithm is terminated, hence regarded as converged, after $k$ iterations, if following stopping criterion is met:

$$||u^k - u^{k-1}|| < t \wedge ||v^k - v^{k-1}|| < t \tag{2.4}$$

## 2.4 Player rankings

Now that we have proposed ranking algorithm that fits our problem, we can implement it to find the solution. At the beginning of this chapter, we mentioned there are numerous statistics of the lineups available, whether traditional or advanced. Our final selection contains following statistics:

- Offensive Rating - OR [1]
- Points per 48 minutes - PT/48 [2]
- Defensive Rating - DR [3]
- Net Rating - NR [4]

First two attributes are representing the ability of a lineup to score points. In contrast, defensive rating expresses the ability to stop the opposing team from scoring, and net rating is a combination of offensive and defensive capability, thus overall winning potential.

When analysing properties of the graph, we ran upon certain obstacles, i.e. how to differentiate between players in same the lineup. We decided to manage this problem with the initial vector. This vector will be assembled from following individual statistics:

- Point Per Game - PPG
- Offensive Rebounds Per Game - ORB
- Assists Per Game - AST
- Defensive Rebounds Per Game - DRB
- Steals Per Game - STL
- Blocks Per Game - STL
- Player Efficiency Rating - PER[5]

[1] Amount of points produced by a lineup per 100 possessions
[2] Points score by per 48 minutes of play by lineup
[3] Amount of points allowed to other team by a lineup per 100 possessions
[4] Difference between Offensive and Defensive Rating
[5] A measure of per-minute production standardized such that the league average is 15

We can notice there are more individual statistical attributes than statistical attributes for lineups. We will explain and showcase their incorporation, and analyse behaviour of ranking with respect to different values of parameters $\alpha$ and $\beta$ in the first player ranking application.

After the algorithm provides players ranking, we will compare several players to analyze how AEW algorithm handled inputs and weather output met our requirements. We will also compare ranking result to relevant advanced statistical attributes, that are already used in [1] as a performance measure:

- Box Plus/Minus - BPM
- Offensive Box Plus/Minus - OBPM
- Defensive Box Plus/Minus - DBPM
- Win Shares Per 48 Minutes - WS/48
- Offensive Win Shares Per 48 Minutes - OWS/48
- Defensive Win Shares Per 48 Minutes - DWS/48

We consider these measures relevant to our goal, which is the value of a player in terms of team contribution. Win shares estimate the number of wins that are contributed by a player due to his offence, defence or overall performance. Next, is box plus/minus, which is a box score estimate of offensive, defensive or just points per 100 possessions a player contributed.

Results presented in this chapter were acquired in python. Libraries such as Pandas were used for data handling. Networkx for working with network objects and Matplotlib provided visualization. We created function AEW, which with correct input returns ranking vectors based on the proposed algorithm 3. Function AEW utilizes libraries such as Numpy and Scipy for mathematical operations and vector/matrix handling. Data and source codes appertaining to thefollowing solutions are available at [16].

### 2.4.1 Offensive Rating

First ranking of players will be based on OR as linuep's offensive measure and PPG as player's individual offensive measure. In this particular example, we will go into detail of the whole procedure.

For better understanding of AEW's performance, we decided to interpret the results by showcasing several players and compare their statistics and ranking score. Statistics are useful indicators of how the algorithm dealt with graph's properties (2.2.2) that dissuaded us from using other ranking algorithms.

Input, as presented in algorithm 3, will be biadjacency matrix $W$, containing information about the offensive ratings of the lineups. It also stores information about membership of a player in a lineup. Next, we will set initial vectors $u^0$ and $v^0$. We will adjust initial vector $u^0$ with a individual PPG statistic and normalize it using $L1$-norm. Initial vector $v^0$ will have for each element same default value $\frac{1}{|V|}$, thus it is also normalized. For hyperparameters $\alpha$ and $\beta$, pertaining to the algorithm equation for scoring vector of players and lineups, respectively, we will choose different combinations of values to showcase their impact on the ranking. Nevertheless, in the end we will select single values for these hyperparameters, to eventually obtain final ranking of the players.

In our case, the whole procedure takes between 5 to 20 iterations based on values of the hyperparameters. The higher the value of $\alpha$, $\beta$ is, the more iteration it takes for the algorithm to converge. This is due to the fact that algorithm learns more from structure of the graph. If they are smaller, algorithm will converge fast, since initial vectors have greater impact and the results will not deviate from them substantially. Nonetheless, number of iterations that is required for the convergence is rather small. Iterative ranking algorithms are generally known for a modest number of iterations.

Results of the ranking for several players are showcased in table 2. We decided to calculate ranking for 5 different combinations of $\alpha$, $\beta$. Besides that, we displayed additional information. Number of the lineups a player is member of in the network, average OR of those lineups, PPG statistic of a player, could be a relevant indicator of how the algorithm behaves for different values of hyperparameters. It also provides us with an expectation of the final ranking.

| Player | #lineups | $\overline{OR}$ | PPG | Player AEW Ranking | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | $\alpha = 0.3$ $\beta = 0.5$ | $\alpha = 0.7$ $\beta = 0.5$ | $\alpha = 0.5$ $\beta = 0.5$ | $\alpha = 0.5$ $\beta = 0.3$ | $\alpha = 0.5$ $\beta = 0.7$ |
| LeBron James | 8 | 117.33 | 27.6 | 7 | 4 | 6 | 6 | 5 |
| Luka Doncic | 2 | 103.45 | 31 | 2 | 11 | 4 | 3 | 6 |
| Tyler Herro | 7 | 108.9 | 16 | 43 | 49 | 43 | 42 | 46 |
| Russel Westbrook | 1 | 94 | 17.9 | 37 | 73 | 47 | 46 | 51 |
| Kentavious Caldwell-Pope | 6 | 116.85 | 10.7 | 66 | 47 | 58 | 58 | 56 |
| OG Anunoby | 5 | 101.7 | 10.5 | 80 | 92 | 85 | 83 | 86 |
| Maxi Kleber | 2 | 103.45 | 6.7 | 104 | 103 | 10 | 102 | 104 |

Table 2: NBA players AEW ranking analysis

Firstly, let us take a look on how algorithm performed and whether the result correspond to our propositions. One of the demands was that score should be independent of the number of the lineups, a players is member of. We established that in properties of the algorithm and we can look at players like Luka Doncic and Tyler Herro, where latter's higher number of lineup participations does not improve the final rank.

Next proposition was distinction of players within the same lineup spectrum. We managed this by implementing initial form of the scoring vector. Maxi Kleber and Luka Doncic are members of the same two lineups and their ranks are poles apart due to Luka Doncic's superior PPG statistic.

When it comes two impact of hyperparameters, remember that $\alpha$ regulates influence of graph properties in update rule (algorithm 3 line 8) for scoring vector $u$ of players and $\beta$ in update rule (algorithm 3 line 9) for scoring vector $v$ of lineups. When $\alpha = 0.3$, Russell Westbrook was ranked higher than Tyler Herro, since impact of initial vector is greater. However, ranking is different when the structure of the graph takes upper hand for $\alpha = 0.7$, as Tyler Herro was member of lineups with overall better OR. Hyperparameter $\beta$ operates with a similar trend but with lesser impact on ranking score. Reason is that ranking vector for lineups $v$ has influence on ranking vector $u$, thanks to Algorithms essential characteristic: mutually reinforcing relationship between the two classes of nodes within

the network. Algorithm satisfied our expectations and is producing favourable ranking.

Second, a more important way to measure quality of our result will be comparison with mentioned advanced statistics, that have the nature of measuring player's contribution to his team. For final ranking, we decided to set values of hyperparameters $\alpha = 0.6, \alpha = 0.6$. These values give us well balanced impact of lineup's OR and individual performance of the player.

Tolerance value from stopping criterion 2.4 was set to $t = 1 \times 10^{-4}$, which led to termination of the algorithm after 13 iterations. Figure 8 showcases 20 iteration of the algorithm and how the ranking score of two players evolved.

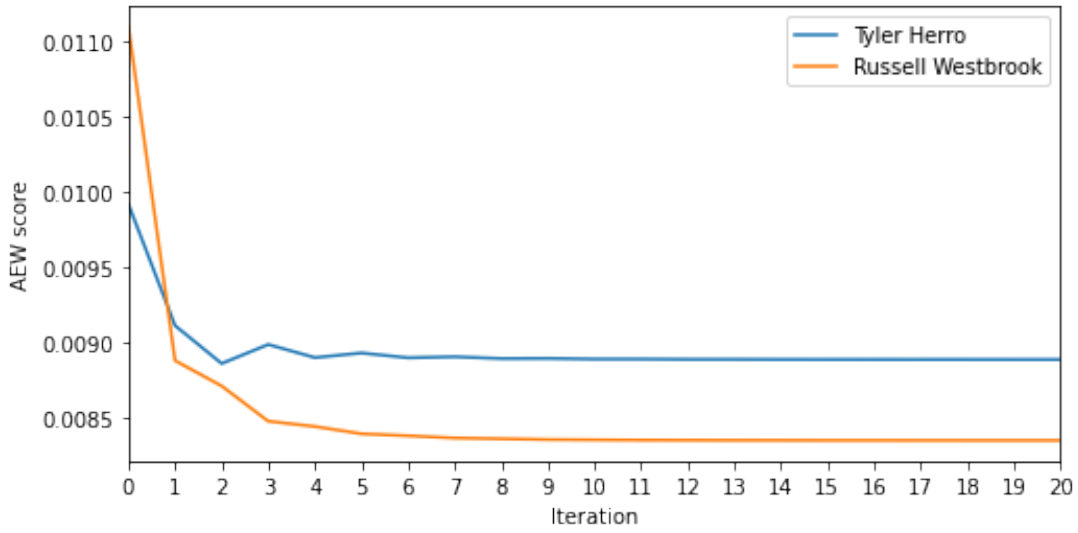



Figure 8: Convergence of ranking scores throughout iterations

Based on this setting, highest ranking positions can be observed in figure 9. Ranking for all 117 players can be found in Figure A.1 in appendix A and the same applies for the rest of rankings in this chapter.

All offensive performance measures agreed on Donovan Mitchell being best offensive player in NBA playoff 2019/2020. From 15 players, AEW has agreed on 11 with OBPM and on 9 with OWS/48. Even OBM and OWS/48 ranked players differently but agreed on 10 players. Advanced statistics are calculated from basic statistical attributes, but what basic statistical attributes are crucial and their importance is what creates differences. Overall, we are pleased with the result from our scoring algorithm, as it seems to be hand to hand with official statistics.

| Rank | AEW | OBPM | OWS/48 |
|---|---|---|---|
| 1 | Donovan Mitchell | Donovan Mitchell | Donovan Mitchell |
| 2 | Kawhi Leonard | Giannis Antetokounmpo | Mike Conley |
| 3 | Anthony Davis | LeBron James | Anthony Davis |
| 4 | LeBron James | James Harden | Kristaps Porziņģis |
| 5 | Jayson Tatum | Luka Dončić | LeBron James |
| 6 | Joel Embiid | Mike Conley | James Harden |
| 7 | Luka Dončić | Anthony Davis | Jamal Murray |
| 8 | James Harden | Kawhi Leonard | Giannis Antetokounmpo |
| 9 | Giannis Antetokounmpo | Nikola Vučević | Serge Ibaka |
| 10 | Jamal Murray | Kristaps Porziņģis | Kawhi Leonard |
| 11 | Jaylen Brown | Jamal Murray | Jimmy Butler |
| 12 | Paul George | Nikola Jokić | Nikola Jokić |
| 13 | Mike Conley | Damian Lillard | Robert Williams |
| 14 | Nikola Jokić | Serge Ibaka | Georges Niang |
| 15 | Kemba Walker | Joel Embiid | Rudy Gobert |

Figure 9: Top offensive NBA players according to AEW (OR), OBPM, OWS/48

### 2.4.2 Points per 48 minutes

Just like advanced statistics are assembled from different basic statistical attributes, we can use other statistics in our model. In this part, we will construct a bipartite graph with use of PT/48 statistic. We will also change the individual statistical attribute that is embedded into the initial vector. With other than points, a player also contributes to team offence when assisting and rebounding. Therefore, in this case we will consider sum: PPG + AST + ORB, his individual statistical attribute. Hyperparameters are set to:$\alpha = 0.5, \alpha = 0.5$. Results for this kind of graph setting are displayed in figure 10

We can notice that Donovan Mitchell remains the highest ranked offensive player. Our ranking again agreed on 11 players with OBPM and on 9 with OWS/48. In this case however, these players are not all the same. Nicola Vucevic and Jimmy Butler are now ranked in top 15 while missing in the previous ranking.

| Rank | AEW ranking | OBPM | OWS/48 |
|---|---|---|---|
| 1 | Donovan Mitchell | Donovan Mitchell | Donovan Mitchell |
| 2 | Kawhi Leonard | Giannis Antetokounmpo | Mike Conley |
| 3 | LeBron James | LeBron James | Anthony Davis |
| 4 | Luka Dončić | James Harden | Kristaps Porziņģis |
| 5 | James Harden | Luka Dončić | LeBron James |
| 6 | Giannis Antetokounmpo | Mike Conley | James Harden |
| 7 | Anthony Davis | Anthony Davis | Jamal Murray |
| 8 | Jayson Tatum | Kawhi Leonard | Giannis Antetokounmpo |
| 9 | Joel Embiid | Nikola Vučević | Serge Ibaka |
| 10 | Caris LeVert | Kristaps Porziņģis | Kawhi Leonard |
| 11 | Jamal Murray | Jamal Murray | Jimmy Butler |
| 12 | Nikola Vučević | Nikola Jokić | Nikola Jokić |
| 13 | Nikola Jokić | Damian Lillard | Robert Williams |
| 14 | Paul George | Serge Ibaka | Georges Niang |
| 15 | Jimmy Butler | Joel Embiid | Rudy Gobert |

Figure 10: Top offensive NBA players according to AEW (PT/48), OBPM, OWS/48

### 2.4.3 Defensive Rating

First two rankings were dedicated to offensive value of a player. Now, we will try to rank players based on their defensive contribution. At the beginning of this section we introduced DR of a lineup. DR is the amount of points allowed to opposing team per 100 possessions. Therefore, covariance of DR and defensive quality of a lineup is negative. Consequently, assigning weights to edges within the graph based on DR will result in false ranking. We will adjust DR values following way:

$$DR'_m = 2 * mean(DR) - DR_m$$

where $DR'_m$ is adjusted value of $m$-th DR. Adjusted DR now has a positive relationship with quality of defence and we can construct graph.

In Previous part we used a combination of basic offensive statistical attributes to represent the initial ranking vector. Similarly, we will consider blocks and steals the player's contribution on defensive end and consider sum: BLK + STL, his individual statistical attribute.

Defensive advanced statistics that will serve as quality measures of ranking are DBPM and DWS/48. Hyperparameters are set to:$\alpha = 0.5, \alpha = 0.5$. Comparison is displayed in figure 11.

| Rank | AEW ranking | DBPM | DWS/48 |
|---|---|---|---|
| 1 | Myles Turner | Robert Covington | Robert Covington |
| 2 | Robert Covington | Patrick Beverley | Robert Williams |
| 3 | Justin Holiday | Giannis Antetokounmpo | Serge Ibaka |
| 4 | T.J. Warren | Derrick Jones Jr. | Kyle Lowry |
| 5 | Victor Oladipo | LeBron James | Gordon Hayward |
| 6 | Kawhi Leonard | Kawhi Leonard | Jayson Tatum |
| 7 | Anthony Davis | Grant Williams | Fred VanVleet |
| 8 | James Harden | Serge Ibaka | Giannis Antetokounmpo |
| 9 | Jimmy Butler | Kyle Lowry | OG Anunoby |
| 10 | Brook Lopez | Marc Gasol | Marc Gasol |
| 11 | Kyle Lowry | Mike Conley | James Ennis |
| 12 | OG Anunoby | Chris Paul | Pascal Siakam |
| 13 | Joel Embiid | Rajon Rondo | Daniel Theis |
| 14 | Jayson Tatum | Andre Iguodala | Jaylen Brown |
| 15 | Fred VanVleet | Anthony Davis | LeBron James |

Figure 11: Top defensive NBA players according to AEW (DR), DBPM, DWS/48

We can notice that Robert Covington was ranked high in our ranking and also in both DBPM and DWS/48 rankings. Other than that results are not similar at all. From 15 players our ranking agreed only on 4 players with DBPM and 3 players with DWS/48. Furthermore, DBPM and DWS/48 are not in agreement neither, with only 6 players occurring in both statistics. In general, it is hard to capture defensive quality of a player only with simple statistics such as STL and BLK. On offensive end it is more straightforward.

### 2.4.4 Net Rating

Last ranking obtained from our algorithm will be also probably most relevant. NR is difference of OR and DR, it captures both offensive and defensive quality of a lineup, and indicates overall superiority over opposing lineups. NR of a lineup can have negative value. Since edges in biadjacency matrix $W$ carry strictly non-negative values, rescaling is necessary. Rescaled values will be obtain from:

$$NR'_m = (b-a)\frac{NR_m - min(NR)}{max(NR) - min(NR)}$$

where $NR'_m$ is adjusted value of $m$-th NR and $(a, b)$ is interval values are rescaled to. In our case we will set interval $(80, 130)$. This way ratings will have similar distribution as in OR and ratios between ratings will be sensible.

As individual statistical attribute we will use Players Efficiency Rating. PER is advanced measure that combines offensive and defensive abilities of a player. Our ranking

(figure 12) is compared with advances statistics WS and BPM and we set:$\alpha = 0.5, \alpha = 0.5$.

| Rank | AEW ranking | BPM | WS/48 |
|---|---|---|---|
| 1 | Donovan Mitchell | Giannis Antetokounmpo | Anthony Davis |
| 2 | LeBron James | LeBron James | LeBron James |
| 3 | Giannis Antetokounmpo | Kawhi Leonard | Donovan Mitchell |
| 4 | Anthony Davis | James Harden | Mike Conley |
| 5 | Kawhi Leonard | Donovan Mitchell | James Harden |
| 6 | Mike Conley | Mike Conley | Serge Ibaka |
| 7 | James Harden | Anthony Davis | Giannis Antetokounmpo |
| 8 | Kristaps Porziņģis | Luka Dončić | Kawhi Leonard |
| 9 | Luka Dončić | Serge Ibaka | Robert Williams |
| 10 | Enes Kanter | Nikola Jokić | Jimmy Butler |
| 11 | Serge Ibaka | Nikola Vučević | Kristaps Porziņģis |
| 12 | Joel Embiid | Jayson Tatum | Bam Adebayo |
| 13 | Nikola Jokić | Joel Embiid | Enes Kanter |
| 14 | Jimmy Butler | Jimmy Butler | Jamal Murray |
| 15 | Jayson Tatum | Jamal Murray | Nikola Jokić |

Figure 12: Top NBA players according to AEW (NR), BPM, WS/48

From 15 players our ranking agreed on 13 with BPM ranking and 12 with WS/48. Moreover, all 15 player are ranked in at least one from advanced statistic rankings.

Presented ranking proved to be adequate considering its comparison to officially used advanced statistical measures. Presented AEW ranking algorithm meet its expectations. Disadvantage may be in data used, as one can argue that there was not enough information for players which teams lost in early rounds of playoffs. Possible extensions and modifications would be to consider the whole season, or more years of playoffs together.

AEW is useful for applications where degree centrality is not a deciding factor. One example of such an application could be ranking of movies. Suppose we have an online movie service where users are able to review movies with a rating. Movies can have different numbers of reviews, depending on how watched they are. But does higher watching count of famous Hollywood movies automatically make them superior to less known European movie? With the use of AEW we can suppress the number of reviews and focus solely on rating. However, too small of a number of reviews may be a problem. This can be solved by setting a threshold for the minimum number of reviews. Therefore only relevant movies will be ranked and based only on ratings.

# 3 Recommender systems

Recommender systems are playing an essential role in sales, entertainment industry and many other businesses. Knowledge of previous customer behaviour and interactions offers possibility to make personalized product suggestions. This method is called collaborative filtering, which is one of two major techniques used in recommender systems. In this chapter, we will focus on how bipartite networks are utilized in machine learning based recommendation models.

Firstly, we will explain why usage of bipartite networks is an adequate approach and what is the role of ranking algorithms. We will go into detail of the proposed recommendation method and explain an idea behind implementation of ranking algorithms in typical machine learning workflow. For this part we will take advantage of well-known recommendation music dataset that will serve as a trial application of the presented approach. In the end, we will build a recommendation system for an educational institution, as we will make personalized courses suggestions for students.

## 3.1 Bipartite networks in recommendation models

As we showcased in the previous chapters, bipartite graphs provide universal language to represent relationships between two classes. In that manner we have the opportunity to investigate interactions between customers and products, services or content. Moreover, quality of relationship can be derived from a specific application, whether it is review of a product, rating of a movie, etc.

In the thesis, we will refer to the one class of nodes as users and the other class of nodes as items. Network representation allows utilizing both node-based information and structural information. Main idea is to a create new connection within the network based on gathered data from previous interactions. Approach shows traits of collaborative filtering method, as we are collecting preference or taste information from users. We will obtain these preferences from networks specific centralities using ranking algorithms. In other words, we aim to model similarity among users and items.

Consequently, this becomes a link prediction problem and for such we will implement the machine learning solution introduced by authors in [13]. Main idea is utilization of

the centrality/similarity metrics as predictors/features in the binary classification models. In their study, the random walk-based metric ItemRank proved to be the best similarity metric used as input feature of machine learning models. In this thesis, we will experiment with the BiRank algorithm, introduced in chapter 1, to see how it performs against ItemRank and whether they complement each other, when used together as features.

This can be achieved by evaluating machine learning models. Their performance directly indicates how centrality based features were able to capture user-artist pairs, where a new connection occurred. We will employ machine learning models such as logistic regression, random forest, support vector classifier and extreme gradient boosting. After deployment of binary classification models, we will obtain probability that a new link occurs. In other words, in what measure is recommendation on an item suitable to the user based on previous behaviour.

Implementation to obtain results presented in this chapter was undertaken in Python. Pandas library was used for data handling and Networks for graph operations. Seaborn and Matplotlib provided visualization. We also utilized Numpy and Scipy as mathematical libraries. For machine learning procedures we used Scikit-learn and XGBoost. Data for applications and source codes appertaining this chapter are available at [16].

## 3.2 Music recommendation

The following application used for music recommendation serves to demonstrate and test machine learning-based approach that utilizes centralities in bipartite networks.

### 3.2.1 Last.fm dataset

We will take advantage of data generated from the Last.fm online music system. This data structure contains information about 1892 users and 17632 artists. Detailed description can be found at [9] and it is available to download at [10].

Part of the dataset contains user and artist interactions, particularly listening counts. This will serve as a user review of an artist. We decided to select only relevant artists and omit those with less than 5 fans, i.e. users listening to an artist. Altogether we arrive at 1881 users, 2828 artists and 71426 listening counts, which cover 1.34 % of all possible relationships, which is called density.

For listening counts, we engaged in quantile binning, meaning we divided listening counts into 10 quantiles with equal fractions of the sample. Then we rated 10 % of samples with the highest listening count with rating of 10. This represented artists' rating by the user, i.e. values on scale from 1 to 10. This was done in order to construct a reasonable topology of the graph, as edges in the graph obtained weights based on listening counts.

### 3.2.2 Network construction

When it comes to network construction, we will begin with separating user-artist relations data and use first part, a training part, to create graph $G$ at time $T0$. This bipartite graph has 40k edges that carry knowledge of previous users interactions. More than 30k remaining known relations, testing relations, will be considered as happening in the future, i.e. a user will start listening to an artist. Including them in $G[T0]$ we obtain $G[T1]$, as displayed in figure 13. If a non-edge in $G[T0]$, becomes an edge in $G[T1]$, it is positively labeled - 1. Next, we will randomly select 50k user-artist pairs where an edge did not occur and label them negatively - 0, to capture instances, where a user did not start listening to an artist. This will be a response/target variable in the latter supervised learning algorithms, representing occurrence of connections.

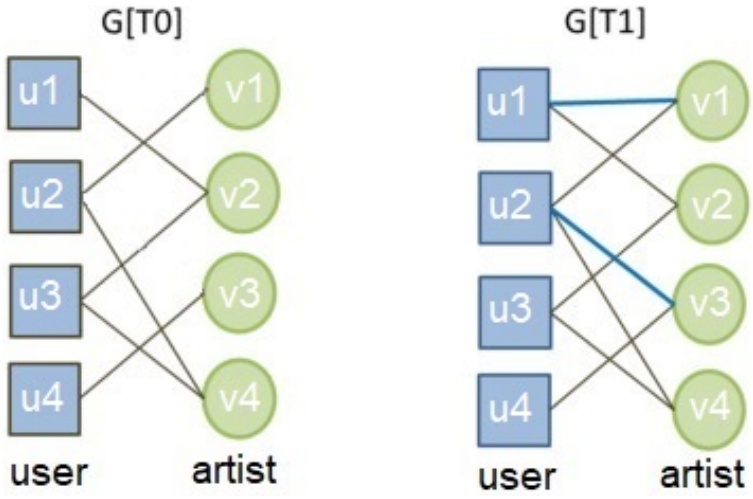



Figure 13: New edge occurrences

### 3.2.3 Preference ranking

To complete our dataset for binary classification, we will make use of ranking algorithms to obtain features. Rankings of artists for each user are calculated from $G[T0]$, and this score will be assigned to the user-artist pair. Intention is that positively labelled pairs will have higher score than negatively labelled pairs, hence there will be a clear trend that can be captured in binary classifiers.

#### ItemRank

First feature will be based on a random-walk based scoring algorithm ItemRank, introduced by authors in [7]. Essential component of ranking procedure is a correlation matrix $C$. It portrays the correlation between artists based on number of the listeners artists share. Naturally, the matrix is symmetric and eventually normalized to a stochastic matrix in its final form.

ItemRank algorithm is a modified biased version of PageRank, which was mentioned in section 1.4. Therefore it is also iterative an algorithm and resulting update rule acquires the following form:

$$IR_{u_i} = \alpha C IR_{u_i} + (1 - \alpha) d_{u_i} \tag{3.1}$$

where $IR_{u_i}$ is a vector of item preference scores for $i-th$ user, $C$ a is normalized correlation matrix. Static score distribution vector $d_{u_i}$, is a normalized vector that stores known preferences of a user. Adjustment of this vector allows us to obtain items ranking scores for each user. Notice that it has a similar role to initial vectors in BiRank algorithm and AEW. In addition, hyperparameter $\alpha$, called decay factor, will be set to $\alpha = 0.85$, which is a common choice. Using this equation, we were able to calculate an artist preference score for each user. In our code we used the python object from [12].

ItemRank proved to be very effective in research carried out in [13], when it comes to a user-item preference ranking. We will try to challenge ItemRank with BiRank.

### BiRank

In section 1.4.2, we presented BiRank, developed by authors in [8], as modification of HITS that returns ranking of both classes. However, is it possible to obtain a preference ranking of items for particular user? Yes, BiRank has the ability to model a collaborative filtering effect. The key is initial vector $v_0$, which serves as an embodiment of the user's known preference. For each user, an initial vector will have values only for artists he is connected to in $G[T0]$. This will cause that an impact of graph's structure will focus mainly on other users' nodes with the same preferences in the artist.

Example can be seen in figure 14. When our goal is to rank artists for user $u1$, we will construct an initial vector with a setting of value 8 for artist $v1$ and value 0 for other artists. As $v1$ is connected more strongly to $u2$ than $u3$, $u2$ will be given a higher score then $u3$. This will also reflect on the score of $v2$ and $v3$. BiRank will assign higher score to $v2$ than $v3$, meaning that $v2$ is more preferred candidate for recommendation.

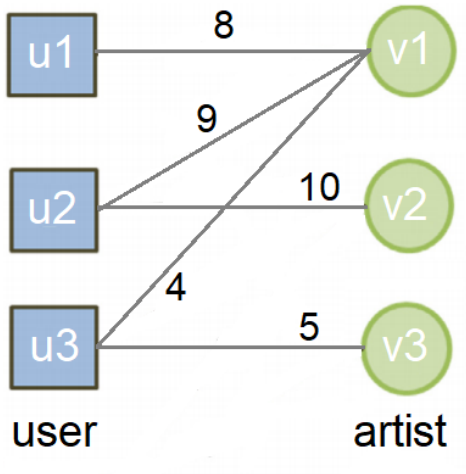



Figure 14: Collaborative filtering effect in BiRank

Consequently, we have obtained ItemRank scores and BiRank scores for each pair. In the next part we will build models using these features and experiment with their combination. Altogether our dataset consists of roughly 80k samples of which approximately 40% are positively labelled pairs. Our implementation of BiRank in python was inspirted by library documentation from [2].

### 3.2.4 Binary classification

Finally, we have all data that is necessary to build prediction models. Machine learning approach brings certain advantages into the whole procedure. Our aim is to predict occurrence of connections and we have kept information of real connections in positively labelled pairs. Performance of models is therefore a direct indicator of how measuring centralities in bipartite network is a useful approach in recommender systems. In addition, predicted probability from classification models present reasonable results that are easy to interpret.

Before fully engaging in model building, we will visualize our data in order to have a initial sense of what should be expected. Histogram in figure 15 showcases how successful each ranking algorithm was in capturing a target variable. It seems that BiRank did a better job and might be a more suitable algorithm.

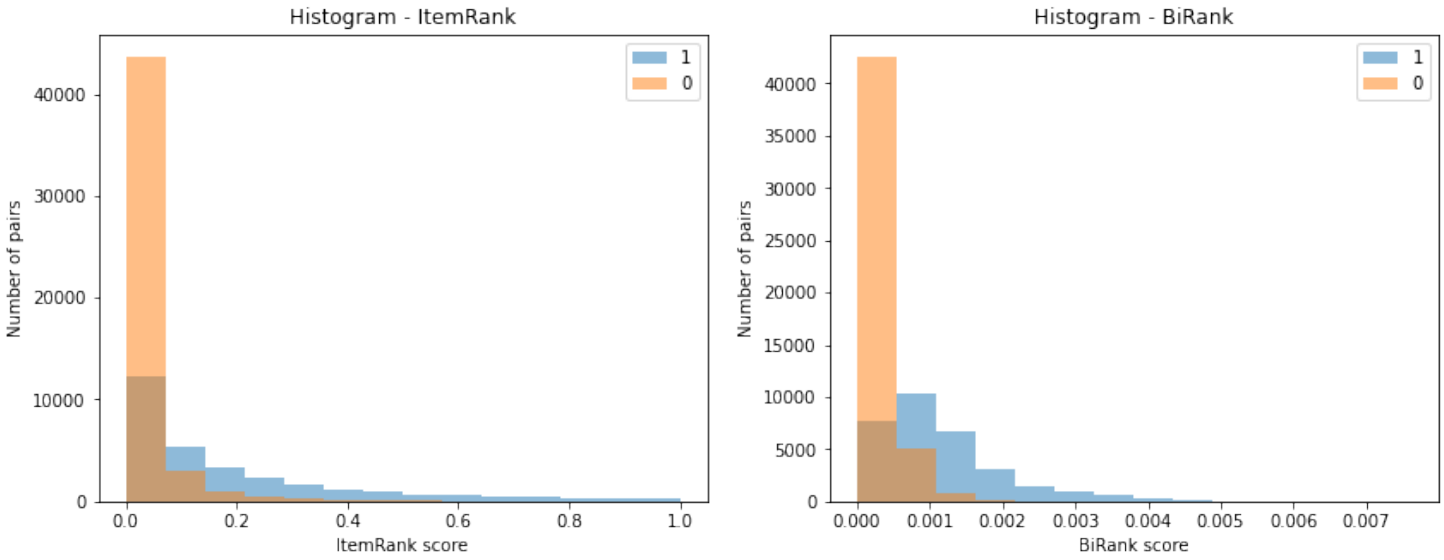



Figure 15: Ranking scores histograms

Next we separated the dataset into training and testing split, with ratio 4:1. As we mentioned before, we attempted to predict a link occurrence using various classification methods: logistic regression (LR), support vector classifier (SVC), random forest classifier (RFC) and extreme gradient boosting classifier (XGB). For each method we selected several hyperparameters that we tried to optimize for best performance. We decided to execute grid search for hyperparameter tuning. On some occasions, we zoomed in on the smaller interval of hyperparameter values. We executed 5-fold cross-validation and used

ROC AUC as a scoring metric. Based on these results we selected parameters used in the models.

For LR we experimented with the value of regularization parameter $C$. Parameter $C$ controls the penalty for misclassified training examples. We tried to optimize a SVC with tuning of hyperparameters such as regularization parameter $C$ and $\gamma$. In addition, we tried both the Gaussian and linear kernel. In RFC we focused on maximal depth of trees, where shallower trees performed better. Moreover, we optimized the number of decision trees in the forest. For XGB, we optimized learning rate, maximum depth, hyperparameters gamma and lambda.

### 3.2.5 Evaluation and Results

We mentioned that by using machine learning approach, beneficial performance measurements are included. Evaluation was executed on test set and we decided to use following metrics:

- **Accuracy** = $\frac{\text{True positives + True negatives}}{\text{Total number of samples}}$
- **Sensitivity** = $\frac{\text{True positives}}{\text{True positives + False negatives}}$
- **Specificity** = $\frac{\text{True negatives}}{\text{True negatives + False positives}}$
- **AUC** - Area under ROC curve, that plots true positive rate (sensitivity) against false positive rate (1 - specificity)

We developed models for both ItemRank and BiRank individually and also combined, to see whether it may yield improved results. Tables containing evaluation results can be seen below.

Table 3: **Performance values by using ItemRank**

| Classification models | LR | SVC | RFC | XGB |
|---|---|---|---|---|
| Accuracy | 0.771 | 0.772 | 0.790 | 0.767 |
| Sensitivity | 0.507 | 0.511 | 0.681 | 0.495 |
| Specificity | 0.940 | 0.939 | 0.860 | 0.943 |
| ROC AUC | 0.842 | 0.842 | 0.842 | 0.842 |

| Table 4: **Performance values by using BiRank** | | | | |
|---|---|---|---|---|
| Classification models | LR | SVC | RFC | XGB |
| Accuracy | 0.798 | 0.823 | 0.823 | 0.830 |
| Sensitivity | 0.551 | 0.677 | 0.677 | 0.735 |
| Specificity | 0.957 | 0.918 | 0.858 | 0.89 |
| ROC AUC | 0.890 | 0.890 | 0.889 | 0.889 |

| Table 5: **Performance values by using ItemRank and BiRank** | | | | |
|---|---|---|---|---|
| Classification models | LR | SVC | RFC | XGB |
| Accuracy | 0.827 | 0.826 | 0.832 | 0.832 |
| Sensitivity | 0.687 | 0.683 | 0.733 | 0.732 |
| Specificity | 0.918 | 0.919 | 0.896 | 0.897 |
| ROC AUC | 0.891 | 0.891 | 0.889 | 0.892 |

Evaluation results showed that machine learning models were able to perform at a similar pace. Differences in sensitivity and specificity only mean a difference in threshold. Nevertheless, ROC curves are almost the same. In this case of more than 2k artists available for recommendation, an adequate approach would be to select higher values of thresholds. This would translate into fewer and more confident quality recommendations.

When it comes to performance of ranking algorithms, BiRank proved to be superior. Combination of both features resulted in very slight improvement, and we might say algorithms complement each other in a certain way, but it is negligible. To sum up, the presented approach that utilizes bipartite networks seems to be useful and performs well.

## 3.3 Course recommendation

Most of the time students at universities have liberty to some extent to construct their study plan from subjects of their interest or specialization. However, when entering a study program, they might not have a full understanding of all courses and their relations. It is common that faculties are trying to organize introduction days, where students have the possibility to interact with older students and graduates and ask them questions. This fact offers an opportunity to use study plans of graduated students as a source of information for the recommendations.

In the previous part, we established applicability and potential of bipartite networks utilization in the recommender systems. Now, we are going to apply this approach with an aim to help students with a selection of courses. We will not go into such details, but rather focus on model deployment and actual recommendation examples.

### 3.3.1 Enrollment data

Course recommendations will be focused on students of Mathematics of Economy, Finance and Modelling at the Faculty of mathematics, physics and informatics of the Comenius university in Slovakia. Duration of a study program is 2 years and study plan is available at [6]. We have collected dummy enrollment data of 3 classes of 2021,2020 and 2019, which consist of a total of 95 students. In addition, we have ratings of those courses by students, as feedback is collected each year. Altogether the dataset contains 1473 tuples: [CourseName, CourseID, CourseRating, StudentID].
A network created from this data has a density of 51.68 %, meaning that the average student enrolled in approximately 15 courses. This density is much higher than in the usual data structure used for recommendation.

Study programs have some compulsory courses. We will be focusing on 30 courses: showed in 16. There are compulsory optional courses, that are divided into two blocks: economical and financial modeling and mathematical modeling. Students are expected to choose one of these directions and are expected to reach a certain amount of credits, but they can still enroll into courses from the opposite block. Optional courses are courses that are most popular for students of the study program outside two mentioned blocks. Figure 17 showcases a distribution of enrollments.

| Mathematical modeling | Economical and financial modeling |
|---|---|
| Cluster analysis and data classification | Computer statistics |
| Multivariate statistical analyses | Pensions and pension funds |
| Convex optimization | Political economics |
| Data dimensionality reduction | Dynamic macroeconomics |
| Databases - SQL | Quantitative methods in risk management |
| Databases and data analysis | Social network analysis |
| Digital signal processing | Stochastic methods of operational analysis |
| Principles of mathematical modelling in science and engineering | Theory of non-cooperative games |
| Optimal control classes | Time series analysis |
| Modelling biological processes | Macroeconomic development and economic policies in Slovakia |
| Stochastic simulation methods | Economics seminar |
| Integral transforms and special functions | Financial derivatives |
| **Optional** | Insurance theory |
| Markov processes (1) | |
| Nonparametric statistic | |
| Economics of information | |
| Stochastic optimization methods | |
| Insurance theory classes | |

Figure 16: List of courses

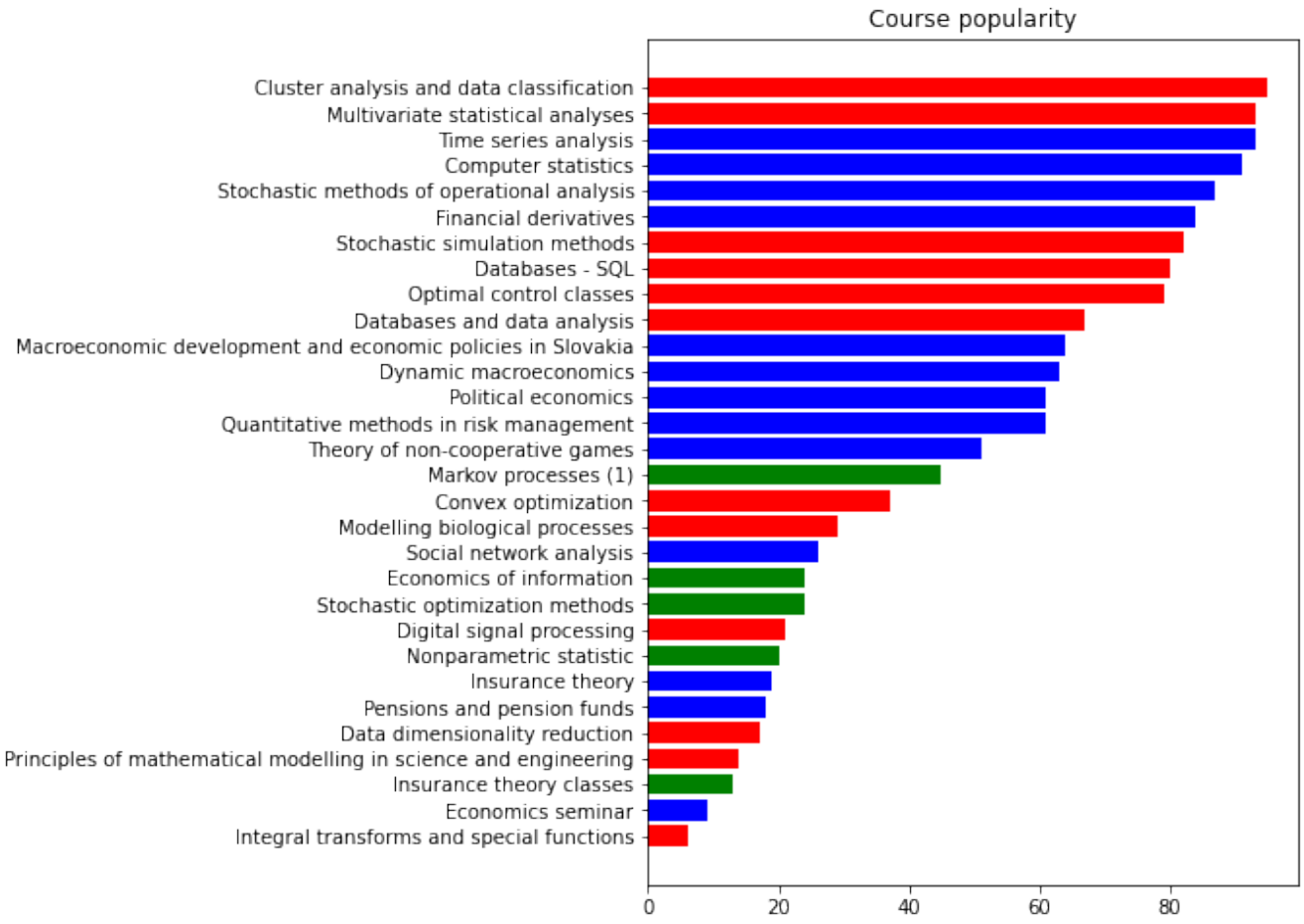



Figure 17: Number of enrollments per course

### 3.3.2 Ranking courses

Applying same principles as in the artist application, we will construct graph $G[T0]$ from 800 student-course relations. Remaining 673 connections are used as positively labelled student-course pairs. In addition, we generate 905 non-connected pairs that are negatively labelled.

Within $G[T0]$ we can calculate both ItemRank and BiRank. Figure B.1 in appendix B shows impact of hyperparameters $\alpha$ in ItemRank and $\alpha, \beta$ in BiRank. We also included 800 pairs, which existed within the network and labelled them - 2.

As parameter values increase, so does impact of graph's topology. Both positively and negatively labelled pairs are approaching known connections. Positively labelled pairs seem to do it slightly faster. Question is, whether it helps with classification. We will compare the results of models for different hyperparameter values in the next part.

### 3.3.3 Model development

Our aim is to select the best model for final recommendations. We will be using machine learning techniques employed in the artist recommendation. Models will be selected using a combination of grid search for hyperparameters and 5-fold cross-validation on the fraction of data amounting to 75%. Remaining share of 25% serves as a test set for evaluation.

In results, which are shown in the tables 6,7 and 8, we observe that hyperparameters only cause no differences or neglectable differences for both ItemRank and BiRank based models. Interestingly enough, both ranking algorithms matched evenly and their combination is again performing slightly better. Relationship between response variable and features is not complex, therefore all classification techniques had satisfactory performance.

For final recommendations, we will be working with 2 logistic regression models, with ItemRank score and BiRank score as features, respectively. Hyperparameters are set to $\alpha = 0.9$ in ItemRank and $\alpha, \beta = 0.9$ in BiRank. These models learned on the whole dataset and are now ready to offer recommendations.

| Table 6: **ROC AUC scores by using ItemRank** | | | | |
|---|---|---|---|---|
| Classification models | LR | SVC | RFC | XGB |
| $\alpha = 0.9$ | 0.884 | 0.884 | 0.883 | 0.876 |
| $\alpha = 0.7$ | 0.882 | 0.882 | 0.880 | 0.877 |
| $\alpha = 0.5$ | 0.879 | 0.879 | 0.878 | 0.872 |

| Table 7: **ROC AUC scores by using BiRank** | | | | |
|---|---|---|---|---|
| Classification models | LR | SVC | RFC | XGB |
| $\alpha, \beta = 0.9$ | 0.883 | 0.883 | 0.879 | 0.882 |
| $\alpha, \beta = 0.7$ | 0.886 | 0.886 | 0.884 | 0.891 |
| $\alpha, \beta = 0.5$ | 0.887 | 0.887 | 0.885 | 0.880 |

| Table 8: **ROC AUC scores by using ItemRank and BiRank** | | | | |
|---|---|---|---|---|
| Classification models | LR | SVC | RFC | XGB |
| $\alpha_{IR}, \alpha_{BR}, \beta = 0.9$ | 0.891 | 0.891 | 0.892 | 0.893 |
| $\alpha_{IR}, \alpha_{BR}, \beta = 0.7$ | 0.892 | 0.893 | 0.894 | 0.892 |
| $\alpha_{IR}, \alpha_{BR}, \beta = 0.5$ | 0.890 | 0.891 | 0.893 | 0.892 |

### 3.3.4 Model deployment

To observe how actual recommendations come into existence, we will create an example student called Joe, who is trying to select course for his study plan. Suppose that Joe is entering his first year of applied mathematics master studies and is picking his optional courses. Joe has already selected the following 6 courses he will enroll into: Cluster analysis and data classification, Multivariate statistical analyses, Modelling biological processes, 'Stochastic optimization methods, Optimal control classes and; Quantitative methods in risk management. Joe is not sure about other courses, however, it is necessary for him to pick several additional course to have enough credits.

To obtain recommendation, we will add Joe into the network, which consists of all previous students enrollment connections and ratings of courses. Based on generated graph, we utilize both ItemRank an BiRank to obtain Joe's preference score for remaining 24 courses available for him. Calculated scores serve as an input for trained models and probabilities that particular connection occurs are acquired.

Recommendation presented to Joe is a list of courses ordered from the mostly recommended ones as shown in figure 18, where 10 courses were chosen for recommendation.

| ItemRank recommendation | Birank recommendation |
|---|---|
| Time series analysis | Computer statistics |
| Computer statistics | Time series analysis |
| Stochastic methods of operational analysis | Stochastic methods of operational analysis |
| Financial derivatives | Stochastic simulation methods |
| Stochastic simulation methods | Financial derivatives |
| Databases - SQL | Databases - SQL |
| Databases and data analysis | Databases and data analysis |
| Dynamic macroeconomics | Macroeconomic development and economic policies in Slovakia |
| Political economics | Political economics |
| Macroeconomic development and economic policies in Slovakia | Dynamic macroeconomics |

Figure 18: Joe's course recommendation

Selecting a fixed number of items for recommendation is one of the possible options. An alternative approach is to observe how probabilities change for each course. Based on figure 19 we decided that top 10 courses would be an adequate selection as we can notice significant change in probabilities. One can argue that change at 19th course is more significant, therefore only 6 courses should be recommended.

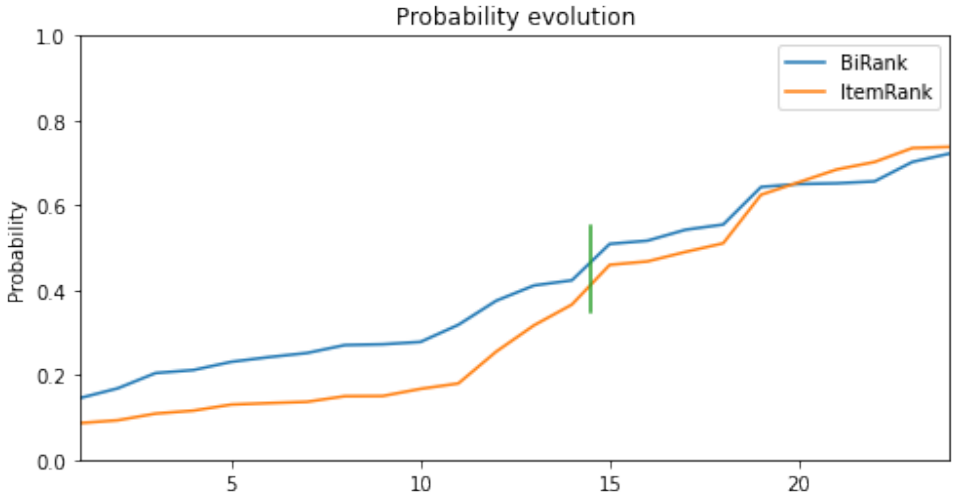



Figure 19: Joe's course recommendation

Models have agreed on top 10 courses for Joe. However, recommendations slightly differ in their order. Full list of courses with probabilities can be found in appendix B figures B.2, B.3.

For Joe we have intentionally decided to select a majority of courses from the mathematical block. Does a recommendation differ when we start with courses from economical and financial block?

We will create a second example of a student called Jane. Her prior selection consists mostly of economical and financial courses: Time series analysis, Dynamic macroeconomics, Stochastic methods of operational analysis, Databases - SQL, Stochastic optimization methods, Quantitative methods in risk management.

Jane's recommendations are shown in figure 18. Full list of courses with probabilities can be found in appendix B figures B.4, B.5. Again we selected recommended courses based on the probability distribution (21).

| ItemRank recommendation | Birank recommendation |
|---|---|
| Cluster analysis and data classification | Computer statistics |
| Multivariate statistical analyses | Cluster analysis and data classification |
| Computer statistics | Multivariate statistical analyses |
| Financial derivatives | Stochastic simulation methods |
| Stochastic simulation methods | Financial derivatives |
| Optimal control classes | Optimal control classes |
| Databases and data analysis | Databases and data analysis |
| Political economics | Macroeconomic development and economic policies in Slovakia |
| Macroeconomic development and economic policies in Slovakia | Political economics |

Figure 20: Jane's course recommendation

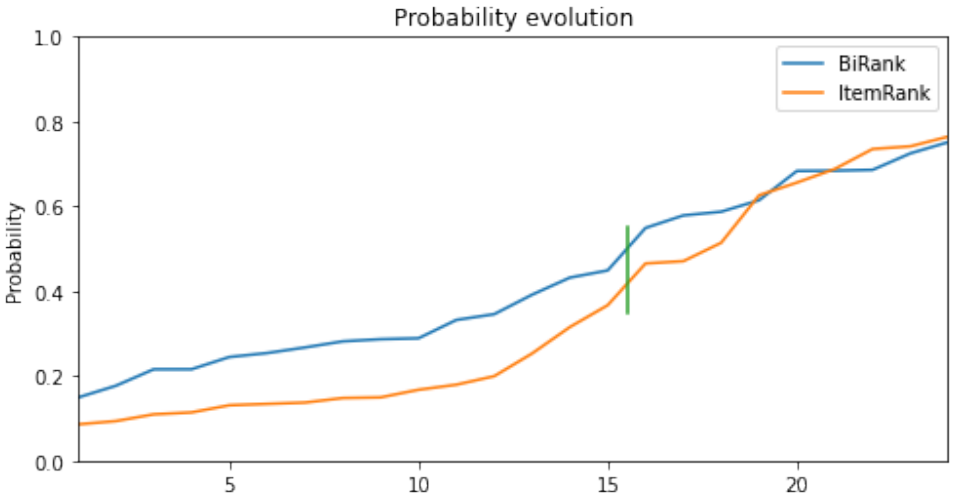



Figure 21: Jane's course recommendation

An initial selection does not seem to have any impact. Notice that courses with the highest preference scores are the courses with most overall enrolments in the data set. This is expected, as the most popular items should be the ones being recommended. However, in our case, a group of courses are dominating the network and almost every student has enrolled into a majority of these courses. Remember, there is 95 students overall and notice distribution of the enrollments in figure 17. This causes that even when students are focusing on a one specific block, they tend to enrol into popular courses from the opposite block. In fact, a share of 98% of students enrolled into at least 7 from a group of 9 most popular courses. Therefore, a dominating group of courses will always be the most recommended.

For this study program majority of optional courses are still contained in the collection which is compulsory. Interesting expansion of this application would be a different study program, where a collection of courses would be larger. Popular courses there would not automatically translate into your recommendation. If your prior selection would associate you with a small group of students, of which the majority did not attend popular courses, recommendation would look different. However, this was not the case for study program presented in our thesis.

# Conclusion

This thesis addresses the use of centralities in bipartite networks and its main goal was to expand applicational use of bipartite networks in not so common fields such as sport analysis and course recommendation. In addition, we presented possible enhancements of undertaken approaches, with hope to motivate further research on bipartite networks applicability in out of ordinary applications.

First chapter of the thesis was an introduction to the basic concepts of bipartite networks. We began with a description of graphs ([15]), which translate networks into mathematical nature and allow analytical possibilities. Next, we focused on centrality properties ([5],[3]) and appertaining ranking algorithms ([8],[14]), which had a major role throughout the whole thesis.

Next chapter was dedicated to a topic that does not receive much attention - sport analysis in networks. Main goal was to obtain ranking of NBA players, based on their team contributions. We began with analysis of the problem and presented reasons why implementation of known ranking algorithms is inappropriate. We introduced ranking algorithm AEW, i.e. average edge weight that we developed for bipartite networks. This approach was an enhancement of the original idea of transforming two-mode graph into one-mode, which was presented by authors of the topic in [17]. We showcased a satisfying results, as obtained ranking was successfully compared with official quality measures used in the sport analysts community.

In the last chapter, we used bipartite networks to create recommendations. At first, we outlined how link prediction via machine learning workflow creates recommendations ([13]). We tested this approach on a common recommender systems dataset and explained the feature role of ranking algorithms in detail. Based on the performance result, we discovered that BiRank seems to be more adequate than ItemRank. Results also proved usefulness and quality of machine learning based link prediction method. Therefore, we introduced a course recommendation model to help students with decision making in the enrollment period. With deployment of two final recommendations models, we interpreted and analysed appertaining results and findings.

As the main contribution of our thesis, we consider dealing with unconventional uses of bipartite networks. To reach desired results, we developed new ranking algorithm that

produced satisfactory results when ranking NBA players based on their team contribution. Created algorithm is suitable for networks where quality of links is deciding while number of connections is not important. Such application could be ranking of movies, where bias for popular movies produced in Hollywood would be suppressed and model would uncover potential of less known movies. For recommender systems, we showcased the effectiveness of BiRank in machine learning based approach and introduced application with potential to help students in decision making. We hope that the thesis has motivated the reader to engage in modelling using bipartite networks, and experiment with uncommon fields of study.

# Bibliography


[1] BASKETBALL REFERENCE: *2020 NBA Playoffs Stats: Advanced*, available at https://www.basketball-reference.com/playoffs/NBA_2020_advanced.html (03.01.2021)

[2] *API documentation of birankpy*, available at https://github.com/BrianAronson/birankr/tree/master/birankpy (05.02.2021)

[3] BORGATTI, S.P., HALGIN, D.S.: *Analyzing Affiliation Networks*, LINKS Center for Social Network Analysis, 2011, available at https://www.researchgate.net/publication/238797464_Analyzing_Affiliation_Networks (15.01.2021)

[4] DE WERRA, D.: *Some Models of Graphs for Scheduling Sports Competitions* , Discrete Applied Mathematics 21, 47-65, 1998, available at https://www.sciencedirect.com/science/article/pii/0166218X88900339 (25.04.2021)

[5] FAUST, K.: *Centrality in affiliation networks*, Social Networks 19, 157-191, 1997, available at https://www.sciencedirect.com/science/article/pii/S0378873396003000 (15.01.2021)

[6] FMFI UK: *Mathematical Economics, Finance and Modelling*, Study plan, available at https://sluzby.fmph.uniba.sk/infolist/en/sp_mEMM.html (27.04.2021)

[7] GORI, M., PUCCI, A.: *ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines*, available at https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-444.pdf (11.04.2021)

[8] HE, X., GAO, M., KAN, M.-Y., WANG, D.: *BiRank: Towards Ranking on Bipartite Graphs*, IEEE Transactions on Knowledge and Data Engineering 29(1), 57–71, 2017, available at https://www.computer.org/csdl/journal/tk/2017/01/07572089/13rRUxD9h5K (09.01.2021)

[9] *hetrec2011-lastfm-2k description*, available at http://files.grouplens.org/datasets/hetrec2011/hetrec2011-lastfm-readme.txt (10.04.2021)

[10] *hetrec2011-lastfm-2k download*, available at https://github.com/yaxue1123/data-analysis-of-last.fm (10.04.2021)

[11] HOSHINO, R., KAWARABAYASHI, K.: *Scheduling Bipartite Tournaments to Minimize Total Travel Distance* , Journal of Artifical Intelligence Research 42, 91-124, 2011, available at https://arxiv.org/ftp/arxiv/papers/1401/1401.3909.pdf (25.04.2021)

[12] *ItemRank.py*, available at https://github.com/arashkhoeini/itemrank (16.04.2021)

[13] KART, O., ULUCAY, O., BINGOL, B., ISIK, Z.: *A machine learning-based recommendation model fir bipartite neworks*,Physica A: Statistical Mechanics and its Applications 553, 2020, available at https://www.sciencedirect.com/science/article/pii/S0378437120300844 (10.04.2021)

[14] KLEINBERG, J.M.: *Authoritative Sources in a Hyperlinked Environment*, Journal of the ACM 46(5), 604–632, 1999, available at https://dl.acm.org/doi/abs/10.1145/324133.324140 (15.01.2021)

[15] KUNEGIS, J.: *Exploiting the Structure of Bipartite Graphs for Algebraic and Spectral Graph Theory Applications*, Internet Mathematics 11(3), 2014, available at https://www.researchgate.net/publication/269711244_Exploiting_The_Structure_of_Bipartite_Graphs_for_Algebraic_and_Spectral_Graph_Theory_Applications (15.01.2021)

[16] MAJHER, J.: *GitHub Repository for data and code used in these thesis*, available at https://github.com/JanMajher/Node-Centralities-in-Bipartite-Networks (01.05.2021)

[17] MISRA, S.: *Player Centrality on NBA Teams* , Standford University, 2016, available at http://snap.stanford.edu/class/cs224w-2017/projects/cs224w-88-final.pdf (09.01.2021)

[18] NBA: *NBA Advanced Stats*, 2020, available at https://www.nba.com/stats/lineups/advanced/?Season=2019-20&SeasonType=Playoffs&sort=MIN&dir=1 (16.01.2021)

[19] TAHERI, S.M., MAHAYAR, H., FIROUZI, M., GHALEBI, E., GROSU, R., MOVAGHAR, A.: *HellRank: a Hellinger-based centrality measure for bipartite social networks*, Social Network Analysis and Mining 7, Article No. 22, 2017, available at https://link.springer.com/article/10.1007/s13278-017-0440-7 (15.01.2021)

[20] WÄSCHE, H., DICKSON, G., WOLL, A., BRANDES, U.: *Social network analysis in sport research: an emerging paradigm* , European Journal for Sport and Society 14, 138-165, 2017, available at https://www.researchgate.net/publication/316709317_Social_network_analysis_in_sport_research_an_emerging_paradigm (25.04.2021)

[21] ZHANG, X.: *Networks Effects in NBA Teams*, Thesis, Arizona State University, USA, 2017, available at https://repository.asu.edu/attachments/191191/content/Zhang_asu_0010N_17334.pdf (25.04.2021)

# Appendix

## A Players ranking

| Rank | Player | Ranking score | 59 | Serge Ibaka | 0.008337 |
|---|---|---|---|---|---|
| 1 | Donovan Mitchell | 0.015655 | 60 | Joe Ingles | 0.008303 |
| 2 | Kawhi Leonard | 0.013316 | 61 | Jarrett Allen | 0.008241 |
| 3 | Anthony Davis | 0.012808 | 62 | Tyler Johnson | 0.008184 |
| 4 | LeBron James | 0.012772 | 63 | Jae Crowder | 0.008093 |
| 5 | Jayson Tatum | 0.012669 | 64 | Eric Bledsoe | 0.008079 |
| 6 | Joel Embiid | 0.012596 | 65 | Carmelo Anthony | 0.008051 |
| 7 | Luka Dončić | 0.012270 | 66 | Royce O'Neale | 0.008044 |
| 8 | James Harden | 0.011857 | 67 | Jerami Grant | 0.007954 |
| 9 | Giannis Antetokounmpo | 0.011798 | 68 | Danny Green | 0.007945 |
| 10 | Jamal Murray | 0.011743 | 69 | Duncan Robinson | 0.007925 |
| 11 | Jaylen Brown | 0.011702 | 70 | Michael Porter Jr. | 0.007924 |
| 12 | Paul George | 0.011679 | 71 | Rajon Rondo | 0.007869 |
| 13 | Mike Conley | 0.011554 | 72 | Dwight Howard | 0.007869 |
| 14 | Nikola Jokić | 0.011263 | 73 | Jusuf Nurkić | 0.007803 |
| 15 | Kemba Walker | 0.011157 | 74 | Trey Burke | 0.007800 |
| 16 | Nikola Vučević | 0.011146 | 75 | Kyle Kuzma | 0.007798 |
| 17 | Lou Williams | 0.011082 | 76 | Danuel House | 0.007651 |
| 18 | Rudy Gobert | 0.010845 | 77 | Shai Gilgeous-Alexander | 0.007593 |
| 19 | Jordan Clarkson | 0.010801 | 78 | Norman Powell | 0.007539 |
| 20 | Caris LeVert | 0.010720 | 79 | Robert Covington | 0.007385 |
| 21 | Jimmy Butler | 0.010710 | 80 | Evan Fournier | 0.007377 |
| 22 | Joe Harris | 0.010703 | 81 | Danilo Gallinari | 0.007271 |
| 23 | Gordon Hayward | 0.010516 | 82 | Jeff Green | 0.007217 |
| 24 | Montrezl Harrell | 0.010314 | 83 | JaVale McGee | 0.007198 |
| 25 | Damian Lillard | 0.010307 | 84 | Markelle Fultz | 0.007179 |
| 26 | Kristaps Porziņģis | 0.010295 | 85 | Paul Millsap | 0.007172 |
| 27 | Khris Middleton | 0.010211 | 86 | Dorian Finney-Smith | 0.007113 |
| 28 | CJ McCollum | 0.010035 | 87 | Alex Caruso | 0.007105 |
| 29 | Goran Dragić | 0.010031 | 88 | Landry Shamet | 0.007088 |
| 30 | Bam Adebayo | 0.009959 | 89 | Markieff Morris | 0.007066 |
| 31 | Georges Niang | 0.009930 | 90 | OG Anunoby | 0.006994 |
| 32 | Enes Kanter | 0.009919 | 91 | Gary Harris | 0.006968 |
| 33 | Malcolm Brogdon | 0.009845 | 92 | Wesley Matthews | 0.006963 |
| 34 | Timothé Luwawu-Cabarrot | 0.009654 | 93 | Al Horford | 0.006894 |
| 35 | Marcus Smart | 0.009508 | 94 | Gary Trent Jr. | 0.006684 |
| 36 | T.J. Warren | 0.009473 | 95 | Terrence Ross | 0.006679 |
| 37 | Fred VanVleet | 0.009356 | 96 | Patrick Beverley | 0.006631 |
| 38 | Josh Richardson | 0.009323 | 97 | Kelly Olynyk | 0.006535 |
| 39 | Marcus Morris | 0.009250 | 98 | Monte Morris | 0.006506 |
| 40 | JaMychal Green | 0.009223 | 99 | P.J. Tucker | 0.006478 |
| 41 | Kyle Lowry | 0.009218 | 100 | Justin Holiday | 0.006325 |
| 42 | Brook Lopez | 0.009095 | 101 | Marc Gasol | 0.006267 |
| 43 | Tobias Harris | 0.009076 | 102 | Maxi Kleber | 0.006246 |
| 44 | Tim Hardaway Jr. | 0.008998 | 103 | Torrey Craig | 0.006099 |
| 45 | Victor Oladipo | 0.008928 | 104 | Kendrick Nunn | 0.006073 |
| 46 | Reggie Jackson | 0.008926 | 105 | Steven Adams | 0.006056 |
| 47 | Tyler Herro | 0.008884 | 106 | Luguentz Dort | 0.005978 |
| 48 | Daniel Theis | 0.008871 | 107 | Robert Williams | 0.005958 |
| 49 | Chris Paul | 0.008833 | 108 | Hassan Whiteside | 0.005948 |
| 50 | Eric Gordon | 0.008808 | 109 | James Ennis | 0.005940 |
| 51 | Shake Milton | 0.008753 | 110 | D.J. Augustin | 0.005886 |
| 52 | Garrett Temple | 0.008663 | 111 | Andre Iguodala | 0.005576 |
| 53 | Kentavious Caldwell-Pope | 0.008627 | 112 | Gary Clark | 0.005243 |
| 54 | Ivica Zubac | 0.008581 | 113 | Grant Williams | 0.005011 |
| 55 | Pascal Siakam | 0.008515 | 114 | Mason Plumlee | 0.004312 |
| 56 | Dennis Schröder | 0.008514 | 115 | Derrick Jones Jr. | 0.004056 |
| 57 | Myles Turner | 0.008432 | 116 | Khem Birch | 0.003803 |
| 58 | Russell Westbrook | 0.008346 | 117 | Wesley Iwundu | 0.003655 |

Figure A.1: OR

| Rank | Player | Ranking score | 59 | Serge Ibaka | 0.008153 |
|---|---|---|---|---|---|
| 1 | Donovan Mitchell | 0.015400 | 60 | Tyler Johnson | 0.008147 |
| 2 | Kawhi Leonard | 0.013949 | 61 | Garrett Temple | 0.008087 |
| 3 | LeBron James | 0.013852 | 62 | Markelle Fultz | 0.007978 |
| 4 | Luka Dončić | 0.013759 | 63 | Joe Ingles | 0.007963 |
| 5 | James Harden | 0.013423 | 64 | JaMychal Green | 0.007876 |
| 6 | Giannis Antetokounmpo | 0.013250 | 65 | Danuel House | 0.007822 |
| 7 | Anthony Davis | 0.012970 | 66 | Carmelo Anthony | 0.007811 |
| 8 | Jayson Tatum | 0.012702 | 67 | Enes Kanter | 0.007783 |
| 9 | Joel Embiid | 0.012481 | 68 | Dorian Finney-Smith | 0.007702 |
| 10 | Caris LeVert | 0.012400 | 69 | Evan Fournier | 0.007687 |
| 11 | Jamal Murray | 0.012286 | 70 | Jeff Green | 0.007674 |
| 12 | Nikola Vučević | 0.012000 | 71 | Danilo Gallinari | 0.007659 |
| 13 | Nikola Jokić | 0.011764 | 72 | Kentavious Caldwell-Pope | 0.007658 |
| 14 | Paul George | 0.011601 | 73 | Jae Crowder | 0.007637 |
| 15 | Jimmy Butler | 0.011535 | 74 | D.J. Augustin | 0.007579 |
| 16 | Malcolm Brogdon | 0.011404 | 75 | Steven Adams | 0.007562 |
| 17 | Khris Middleton | 0.011166 | 76 | Trey Burke | 0.007524 |
| 18 | Jaylen Brown | 0.011127 | 77 | Reggie Jackson | 0.007488 |
| 19 | Kemba Walker | 0.011030 | 78 | Duncan Robinson | 0.007472 |
| 20 | Mike Conley | 0.010936 | 79 | Kyle Kuzma | 0.007330 |
| 21 | Lou Williams | 0.010920 | 80 | Royce O'Neale | 0.007308 |
| 22 | Damian Lillard | 0.010598 | 81 | Robert Covington | 0.007223 |
| 23 | Rudy Gobert | 0.010385 | 82 | Jerami Grant | 0.007220 |
| 24 | Fred VanVleet | 0.010286 | 83 | Terrence Ross | 0.007191 |
| 25 | Bam Adebayo | 0.010277 | 84 | Michael Porter Jr. | 0.007145 |
| 26 | CJ McCollum | 0.010234 | 85 | Alex Caruso | 0.007130 |
| 27 | Chris Paul | 0.010155 | 86 | OG Anunoby | 0.007111 |
| 28 | Kristaps Porziņģis | 0.010108 | 87 | Al Horford | 0.007030 |
| 29 | Jordan Clarkson | 0.010061 | 88 | Norman Powell | 0.006996 |
| 30 | Gordon Hayward | 0.009937 | 89 | Danny Green | 0.006904 |
| 31 | Goran Dragić | 0.009894 | 90 | P.J. Tucker | 0.006881 |
| 32 | Tobias Harris | 0.009864 | 91 | Wesley Matthews | 0.006781 |
| 33 | Kyle Lowry | 0.009848 | 92 | Paul Millsap | 0.006470 |
| 34 | T.J. Warren | 0.009660 | 93 | Gary Trent Jr. | 0.006425 |
| 35 | Dennis Schröder | 0.009637 | 94 | Landry Shamet | 0.006386 |
| 36 | Joe Harris | 0.009627 | 95 | Dwight Howard | 0.006377 |
| 37 | Marcus Smart | 0.009335 | 96 | Maxi Kleber | 0.006369 |
| 38 | Josh Richardson | 0.009331 | 97 | Gary Harris | 0.006302 |
| 39 | Russell Westbrook | 0.009312 | 98 | Monte Morris | 0.006292 |
| 40 | Eric Gordon | 0.009183 | 99 | Patrick Beverley | 0.006282 |
| 41 | Brook Lopez | 0.009082 | 100 | Marc Gasol | 0.006219 |
| 42 | Eric Bledsoe | 0.009082 | 101 | Luguentz Dort | 0.006190 |
| 43 | Timothé Luwawu-Cabarrot | 0.009008 | 102 | Markieff Morris | 0.006114 |
| 44 | Pascal Siakam | 0.008983 | 103 | James Ennis | 0.006088 |
| 45 | Montrezl Harrell | 0.008869 | 104 | Kelly Olynyk | 0.006080 |
| 46 | Tim Hardaway Jr. | 0.008841 | 105 | JaVale McGee | 0.006062 |
| 47 | Rajon Rondo | 0.008813 | 106 | Justin Holiday | 0.005977 |
| 48 | Jarrett Allen | 0.008790 | 107 | Robert Williams | 0.005974 |
| 49 | Marcus Morris | 0.008788 | 108 | Gary Clark | 0.005695 |
| 50 | Tyler Herro | 0.008764 | 109 | Hassan Whiteside | 0.005564 |
| 51 | Georges Niang | 0.008713 | 110 | Kendrick Nunn | 0.005430 |
| 52 | Victor Oladipo | 0.008691 | 111 | Torrey Craig | 0.005269 |
| 53 | Shai Gilgeous-Alexander | 0.008652 | 112 | Andre Iguodala | 0.005006 |
| 54 | Jusuf Nurkić | 0.008587 | 113 | Khem Birch | 0.004865 |
| 55 | Shake Milton | 0.008483 | 114 | Grant Williams | 0.004489 |
| 56 | Myles Turner | 0.008473 | 115 | Mason Plumlee | 0.004332 |
| 57 | Ivica Zubac | 0.008280 | 116 | Wesley Iwundu | 0.004187 |
| 58 | Daniel Theis | 0.008240 | 117 | Derrick Jones Jr. | 0.003221 |

Figure A.2: PT48

| Rank | Player | Ranking score | 59 | Royce O'Neale | 0.008270 |
|---|---|---|---|---|---|
| 1 | Myles Turner | 0.021335 | 60 | Shai Gilgeous-Alexander | 0.008185 |
| 2 | Robert Covington | 0.016959 | 61 | Kristaps Porziņģis | 0.008080 |
| 3 | Justin Holiday | 0.015999 | 62 | Al Horford | 0.008071 |
| 4 | T.J. Warren | 0.015371 | 63 | Marc Gasol | 0.008048 |
| 5 | Victor Oladipo | 0.014429 | 64 | Jerami Grant | 0.008016 |
| 6 | Kawhi Leonard | 0.014360 | 65 | Terrence Ross | 0.007854 |
| 7 | Anthony Davis | 0.012784 | 66 | Wesley Iwundu | 0.007854 |
| 8 | James Harden | 0.012383 | 67 | Goran Dragić | 0.007844 |
| 9 | Jimmy Butler | 0.012271 | 68 | Jae Crowder | 0.007825 |
| 10 | Brook Lopez | 0.012173 | 69 | Ivica Zubac | 0.007769 |
| 11 | Kyle Lowry | 0.012152 | 70 | Danuel House | 0.007747 |
| 12 | OG Anunoby | 0.011873 | 71 | Carmelo Anthony | 0.007656 |
| 13 | Joel Embiid | 0.011838 | 72 | Marcus Morris | 0.007455 |
| 14 | Jayson Tatum | 0.011773 | 73 | Garrett Temple | 0.007411 |
| 15 | Fred VanVleet | 0.011539 | 74 | Luguentz Dort | 0.007373 |
| 16 | Russell Westbrook | 0.011296 | 75 | Donovan Mitchell | 0.007328 |
| 17 | Jarrett Allen | 0.011178 | 76 | Jeff Green | 0.007310 |
| 18 | Jaylen Brown | 0.011145 | 77 | JaVale McGee | 0.007213 |
| 19 | LeBron James | 0.011109 | 78 | Jamal Murray | 0.007127 |
| 20 | Eric Bledsoe | 0.010917 | 79 | Norman Powell | 0.007042 |
| 21 | Paul George | 0.010626 | 80 | Lou Williams | 0.007010 |
| 22 | Gordon Hayward | 0.010610 | 81 | Grant Williams | 0.006884 |
| 23 | Danny Green | 0.010406 | 82 | Robert Williams | 0.006846 |
| 24 | Malcolm Brogdon | 0.010349 | 83 | Derrick Jones Jr. | 0.006667 |
| 25 | Marcus Smart | 0.010180 | 84 | Paul Millsap | 0.006647 |
| 26 | Chris Paul | 0.010068 | 85 | Michael Porter Jr. | 0.006640 |
| 27 | Hassan Whiteside | 0.009976 | 86 | Steven Adams | 0.006615 |
| 28 | Giannis Antetokounmpo | 0.009976 | 87 | Landry Shamet | 0.006607 |
| 29 | Evan Fournier | 0.009944 | 88 | Duncan Robinson | 0.006605 |
| 30 | Daniel Theis | 0.009921 | 89 | Dennis Schröder | 0.006486 |
| 31 | Bam Adebayo | 0.009820 | 90 | Timothé Luwawu-Cabarrot | 0.006470 |
| 32 | Patrick Beverley | 0.009637 | 91 | Dwight Howard | 0.006423 |
| 33 | P.J. Tucker | 0.009558 | 92 | Joe Harris | 0.006384 |
| 34 | Eric Gordon | 0.009558 | 93 | Montrezl Harrell | 0.006331 |
| 35 | Rudy Gobert | 0.009525 | 94 | Danilo Gallinari | 0.006301 |
| 36 | Mike Conley | 0.009519 | 95 | Josh Richardson | 0.006188 |
| 37 | Luka Dončić | 0.009469 | 96 | Kyle Kuzma | 0.006091 |
| 38 | Dorian Finney-Smith | 0.009469 | 97 | Markieff Morris | 0.005977 |
| 39 | Pascal Siakam | 0.009430 | 98 | Gary Trent Jr. | 0.005965 |
| 40 | Nikola Jokić | 0.009407 | 99 | Torrey Craig | 0.005947 |
| 41 | Alex Caruso | 0.009401 | 100 | Monte Morris | 0.005872 |
| 42 | Markelle Fultz | 0.009316 | 101 | Kelly Olynyk | 0.005801 |
| 43 | Khris Middleton | 0.009034 | 102 | Damian Lillard | 0.005773 |
| 44 | Wesley Matthews | 0.009034 | 103 | Jordan Clarkson | 0.005762 |
| 45 | Caris LeVert | 0.008980 | 104 | Joe Ingles | 0.005745 |
| 46 | Kemba Walker | 0.008948 | 105 | Mason Plumlee | 0.005686 |
| 47 | Serge Ibaka | 0.008946 | 106 | Tobias Harris | 0.005560 |
| 48 | Rajon Rondo | 0.008931 | 107 | Tyler Herro | 0.005433 |
| 49 | Gary Clark | 0.008899 | 108 | D.J. Augustin | 0.005343 |
| 50 | Maxi Kleber | 0.008841 | 109 | Khem Birch | 0.005343 |
| 51 | Nikola Vučević | 0.008688 | 110 | Tim Hardaway Jr. | 0.005075 |
| 52 | James Ennis | 0.008688 | 111 | Kendrick Nunn | 0.004870 |
| 53 | Kentavious Caldwell-Pope | 0.008419 | 112 | JaMychal Green | 0.004761 |
| 54 | Andre Iguodala | 0.008363 | 113 | Shake Milton | 0.004619 |
| 55 | Trey Burke | 0.008347 | 114 | Reggie Jackson | 0.004448 |
| 56 | Gary Harris | 0.008343 | 115 | Enes Kanter | 0.004260 |
| 57 | Jusuf Nurkić | 0.008284 | 116 | Tyler Johnson | 0.004044 |
| 58 | CJ McCollum | 0.008284 | 117 | Georges Niang | 0.002963 |

Figure A.3: DR

| Rank | Player | Ranking score | 59 | Derrick Jones Jr. | 0.008116 |
|---|---|---|---|---|---|
| 1 | Donovan Mitchell | 0.014218 | 60 | Trey Burke | 0.008110 |
| 2 | LeBron James | 0.013820 | 61 | Steven Adams | 0.008062 |
| 3 | Giannis Antetokounmpo | 0.013807 | 62 | Norman Powell | 0.008018 |
| 4 | Anthony Davis | 0.013701 | 63 | Eric Gordon | 0.008008 |
| 5 | Kawhi Leonard | 0.013190 | 64 | Danny Green | 0.008006 |
| 6 | Mike Conley | 0.012578 | 65 | Mason Plumlee | 0.007961 |
| 7 | James Harden | 0.012269 | 66 | D.J. Augustin | 0.007950 |
| 8 | Kristaps Porziņģis | 0.012143 | 67 | Joe Ingles | 0.007898 |
| 9 | Luka Dončić | 0.011895 | 68 | Danuel House | 0.007898 |
| 10 | Enes Kanter | 0.011462 | 69 | Tyler Herro | 0.007889 |
| 11 | Serge Ibaka | 0.011239 | 70 | Justin Holiday | 0.007834 |
| 12 | Joel Embiid | 0.011219 | 71 | Tobias Harris | 0.007769 |
| 13 | Nikola Jokić | 0.011140 | 72 | Tim Hardaway Jr. | 0.007720 |
| 14 | Jimmy Butler | 0.011084 | 73 | Jusuf Nurkić | 0.007710 |
| 15 | Jayson Tatum | 0.011059 | 74 | Dorian Finney-Smith | 0.007691 |
| 16 | Jamal Murray | 0.010784 | 75 | Grant Williams | 0.007646 |
| 17 | Bam Adebayo | 0.010583 | 76 | Jeff Green | 0.007634 |
| 18 | Rudy Gobert | 0.010362 | 77 | Terrence Ross | 0.007631 |
| 19 | Montrezl Harrell | 0.010353 | 78 | Markieff Morris | 0.007609 |
| 20 | Nikola Vučević | 0.010311 | 79 | Jae Crowder | 0.007603 |
| 21 | Malcolm Brogdon | 0.010212 | 80 | Russell Westbrook | 0.007561 |
| 22 | Robert Williams | 0.010198 | 81 | CJ McCollum | 0.007478 |
| 23 | Myles Turner | 0.010125 | 82 | Kyle Kuzma | 0.007459 |
| 24 | Jaylen Brown | 0.010015 | 83 | OG Anunoby | 0.007449 |
| 25 | Kemba Walker | 0.009957 | 84 | Dennis Schröder | 0.007419 |
| 26 | JaVale McGee | 0.009929 | 85 | Alex Caruso | 0.007402 |
| 27 | Lou Williams | 0.009891 | 86 | Duncan Robinson | 0.007400 |
| 28 | Ivica Zubac | 0.009769 | 87 | Wesley Matthews | 0.007371 |
| 29 | Brook Lopez | 0.009690 | 88 | Shake Milton | 0.007363 |
| 30 | Rajon Rondo | 0.009605 | 89 | Shai Gilgeous-Alexander | 0.007135 |
| 31 | Paul George | 0.009437 | 90 | Pascal Siakam | 0.007098 |
| 32 | Jordan Clarkson | 0.009414 | 91 | Tyler Johnson | 0.007048 |
| 33 | Dwight Howard | 0.009411 | 92 | Danilo Gallinari | 0.007019 |
| 34 | Georges Niang | 0.009406 | 93 | Khem Birch | 0.006964 |
| 35 | Marcus Morris | 0.009305 | 94 | Josh Richardson | 0.006928 |
| 36 | Goran Dragić | 0.009289 | 95 | Paul Millsap | 0.006847 |
| 37 | JaMychal Green | 0.009281 | 96 | Andre Iguodala | 0.006839 |
| 38 | Caris LeVert | 0.009150 | 97 | Landry Shamet | 0.006802 |
| 39 | Jarrett Allen | 0.009150 | 98 | P.J. Tucker | 0.006790 |
| 40 | T.J. Warren | 0.009110 | 99 | Hassan Whiteside | 0.006764 |
| 41 | Robert Covington | 0.009108 | 100 | Jerami Grant | 0.006618 |
| 42 | Kelly Olynyk | 0.009051 | 101 | Al Horford | 0.006609 |
| 43 | Daniel Theis | 0.009023 | 102 | Gary Clark | 0.006420 |
| 44 | Chris Paul | 0.008932 | 103 | Evan Fournier | 0.006252 |
| 45 | Gordon Hayward | 0.008909 | 104 | Marc Gasol | 0.006251 |
| 46 | Patrick Beverley | 0.008901 | 105 | Royce O'Neale | 0.006245 |
| 47 | Joe Harris | 0.008817 | 106 | Kendrick Nunn | 0.006213 |
| 48 | Kyle Lowry | 0.008810 | 107 | Gary Harris | 0.006157 |
| 49 | Fred VanVleet | 0.008747 | 108 | Maxi Kleber | 0.006155 |
| 50 | Michael Porter Jr. | 0.008668 | 109 | Torrey Craig | 0.006080 |
| 51 | Khris Middleton | 0.008647 | 110 | Timothé Luwawu-Cabarrot | 0.006077 |
| 52 | Damian Lillard | 0.008580 | 111 | Markelle Fultz | 0.006049 |
| 53 | Monte Morris | 0.008555 | 112 | James Ennis | 0.005673 |
| 54 | Victor Oladipo | 0.008443 | 113 | Carmelo Anthony | 0.005594 |
| 55 | Eric Bledsoe | 0.008386 | 114 | Garrett Temple | 0.005526 |
| 56 | Marcus Smart | 0.008255 | 115 | Wesley Iwundu | 0.005515 |
| 57 | Reggie Jackson | 0.008208 | 116 | Gary Trent Jr. | 0.004974 |
| 58 | Kentavious Caldwell-Pope | 0.008196 | 117 | Luguentz Dort | 0.003893 |

Figure A.4: NR

# B Course recommendation

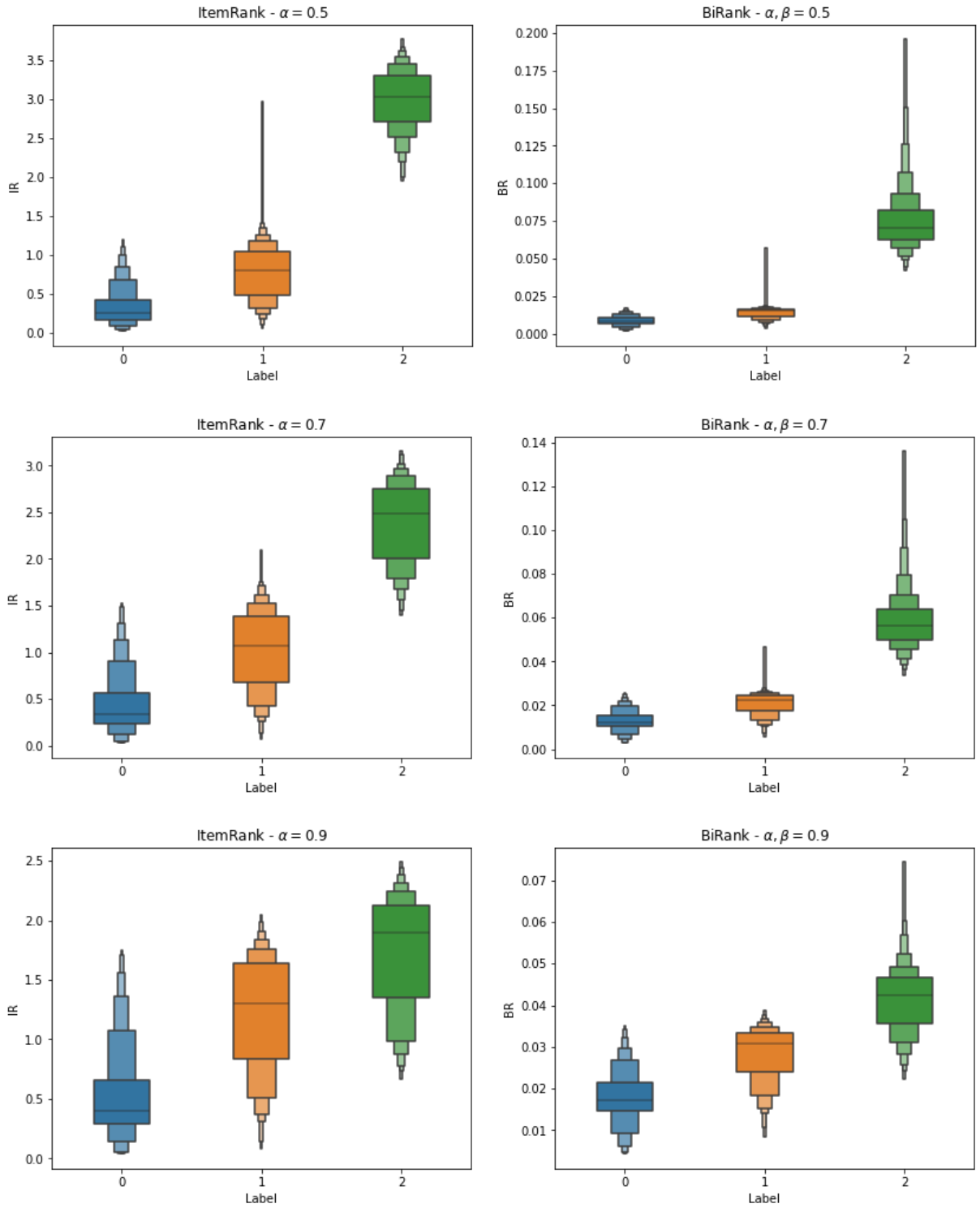



Figure B.1: Preference score distribution

| Probability | Course |
|---|---|
| 0.737682 | Time series analysis |
| 0.734983 | Computer statistics |
| 0.702078 | Stochastic methods of operational analysis |
| 0.684463 | Financial derivatives |
| 0.654405 | Stochastic simulation methods |
| 0.62483 | Databases - SQL |
| 0.510401 | Databases and data analysis |
| 0.490007 | Dynamic macroeconomics |
| 0.467732 | Political economics |
| 0.459605 | Macroeconomic development and economic policies in Slovakia |
| 0.365873 | Theory of non-cooperative games |
| 0.317073 | Markov processes (1) |
| 0.254974 | Convex optimization |
| 0.179962 | Social network analysis |
| 0.167198 | Economics of information |
| 0.150353 | Digital signal processing |
| 0.149914 | Nonparametric statistic |
| 0.136657 | Insurance theory |
| 0.133629 | Data dimensionality reduction |
| 0.130168 | Pensions and pension funds |
| 0.115901 | Principles of mathematical modelling in science and engineering |
| 0.10896 | Insurance theory classes |
| 0.093182 | Economics seminar |
| 0.086509 | Integral transforms and special functions |

Figure B.2: Joe's ItemRank recommendation

| Probability | Course |
|---|---|
| 0.722267 | Computer statistics |
| 0.702264 | Time series analysis |
| 0.65642 | Stochastic methods of operational analysis |
| 0.651816 | Stochastic simulation methods |
| 0.650143 | Financial derivatives |
| 0.643241 | Databases - SQL |
| 0.554632 | Databases and data analysis |
| 0.542295 | Macroeconomic development and economic policies in Slovakia |
| 0.516528 | Political economics |
| 0.508916 | Dynamic macroeconomics |
| 0.423071 | Theory of non-cooperative games |
| 0.410994 | Markov processes (1) |
| 0.374934 | Convex optimization |
| 0.318174 | Social network analysis |
| 0.277876 | Nonparametric statistic |
| 0.272318 | Economics of information |
| 0.270404 | Digital signal processing |
| 0.251719 | Insurance theory |
| 0.242093 | Data dimensionality reduction |
| 0.230946 | Pensions and pension funds |
| 0.211582 | Principles of mathematical modelling in science and engineering |
| 0.204525 | Insurance theory classes |
| 0.168459 | Economics seminar |
| 0.145484 | Integral transforms and special functions |

Figure B.3: Joe's BiRank recommendation

| Probability | Course |
|---|---|
| 0.764314 | Cluster analysis and data classification |
| 0.741416 | Multivariate statistical analyses |
| 0.735052 | Computer statistics |
| 0.687949 | Financial derivatives |
| 0.655909 | Stochastic simulation methods |
| 0.625927 | Optimal control classes |
| 0.514031 | Databases and data analysis |
| 0.470642 | Political economics |
| 0.465213 | Macroeconomic development and economic policie... |
| 0.366793 | Theory of non-cooperative games |
| 0.315278 | Markov processes (1) |
| 0.252863 | Convex optimization |
| 0.199428 | Modelling biological processes |
| 0.179411 | Social network analysis |
| 0.167411 | Economics of information |
| 0.149483 | Digital signal processing |
| 0.147925 | Nonparametric statistic |
| 0.137228 | Insurance theory |
| 0.134101 | Data dimensionality reduction |
| 0.130917 | Pensions and pension funds |
| 0.114034 | Principles of mathematical modelling in scienc... |
| 0.109527 | Insurance theory classes |
| 0.093579 | Economics seminar |
| 0.085928 | Integral transforms and special functions |

Figure B.4: Jane's ItemRank recommendation

| Probability | Course |
|---|---|
| 0.751262 | Computer statistics |
| 0.72482 | Cluster analysis and data classification |
| 0.685817 | Multivariate statistical analyses |
| 0.684014 | Stochastic simulation methods |
| 0.683334 | Financial derivatives |
| 0.614263 | Optimal control classes |
| 0.587219 | Databases and data analysis |
| 0.578521 | Macroeconomic development and economic policie... |
| 0.548932 | Political economics |
| 0.448797 | Theory of non-cooperative games |
| 0.431799 | Markov processes (1) |
| 0.391528 | Convex optimization |
| 0.345596 | Modelling biological processes |
| 0.33204 | Social network analysis |
| 0.288849 | Economics of information |
| 0.286682 | Nonparametric statistic |
| 0.281617 | Digital signal processing |
| 0.267198 | Insurance theory |
| 0.253973 | Data dimensionality reduction |
| 0.24462 | Pensions and pension funds |
| 0.215714 | Insurance theory classes |
| 0.215668 | Principles of mathematical modelling in scienc... |
| 0.176805 | Economics seminar |
| 0.149063 | Integral transforms and special functions |

Figure B.5: Jane's BiRank recommendation