UCLouvain

École polytechnique de Louvain

# Tennis matches outcome prediction via Low-Rank approaches

Author: **Bastien MASSION**
Supervisors: **Pierre-Antoine ABSIL, Guillaume OLIKIER**
Readers: **Pierre-Antoine ABSIL, Guillaume OLIKIER, Julien HENDRICKX**
Academic year 2021–2022
Master [120] in Mathematical Engineering

# Abstract

This thesis tackles the problem of predicting the outcome of tennis matches using low-rank approaches. The work postulates the existence of a true winning probability matrix which generates the results of all matches. Three types of approaches are proposed in order to find this matrix back from the dataset. The first idea consists in setting up a low-rank matrix completion (LRMC) problem. Several classical LRMC techniques as well as new ones adapted to this problem are tested. We introduce second approach consisting in solving a maximum *a posteriori* (MAP) problem on the probability matrix while imposing a low-rank structure. The third novel formulation uses the famous Bradley-Terry-Luce (BTL) model in order to convert the probability guessing problem into a rating guessing problem. This idea reduces the number of constraints and allows for the inclusion of one more feature in the model such as the tournament in which a match is played. The last formulation develops a MAP formulation on the ratings constrained to be low-rank, and the probabilities of winning are computed afterwards via the BTL formula. The new techniques introduced in this work give similar or better results compared to classical LRMC techniques for this problem. Finally, an important statement about the winning probabilities is proved. Even if it is guessed that they should avoid being too large or too small, it turns out that in order to maximize the prediction accuracy, they need to be clipped to zero or one. This implies that, in the MAP framework, any prior distribution that could be chosen symmetric for normalization, is useless.

# Contents

# Acknowledgements

First of all, I would like to thank my supervisor Pr. Pierre-Antoine Absil for his availability every week during the semester, for the documentation he provided me, for his careful listening and deep interest in my findings, for his fruitful ideas to push the thesis further and for all his tips regarding the research process.

Guillaume Olikier, who co-supervised the thesis, was present at every meeting as well and involved in each step of the thesis. He also gave me important advice for the writing process and guided me through the continuation of my life as a researcher. For all of that, I am really thankful.

Then, I would like to acknowledge the last member of the Master's thesis Jury as a reader, Pr. Julien Hendrickx.

I have been lucky to partially share my thesis topic with Julien Herman. In addition to our interesting debriefing conversations after the meetings, our collaboration for data collecting and (pre)processing as well as for the writing of the introduction has been particularly time-saving.

Special thanks go to Julien Dewez, Marine Branders and Astrid Vekemans for their careful rereading of this work.

Finally, I am really thankful to my parents Laurence and Paul, my brother Roman and my girlfriend Mathilde for motivating me and supporting me throughout this thesis. Moreover, thank you Roman and Mathilde for helping me with the corrections of the final details. Above all, I need to thank them all for their involvement, interest and unfailing encouragement during my studies and for always pushing me to do my best.

# List of abbreviations

| | |
|---|---|
| ATP | Association of Tennis Professionals |
| BCD | Block Coordinate Descent (algorithm) |
| BMF | Bilinear Matrix Factorization |
| BTL | Bradley-Terry-Luce (model) |
| CM | Conditional Mean |
| FNM | Frobenius Norm Minimization |
| LASSO | Least Absolute Shrinkage and Selection Operator |
| LRMC | Low-Rank Matrix Completion |
| MAP | Maximum *A Posteriori* |
| MC | Matrix Completion |
| ML | Maximum Likelihood |
| MMSE | Minimum Mean Squared Error |
| MSE | Mean Squared Error |
| NMF | Nonnegative Matrix Factorization |
| NNM | Nuclear Norm Minimization |
| RMSE | Root Mean Squared Error |
| SVD | Singular Value Decomposition |
| WLASSO | Weighted Least Absolute Shrinkage and Selection Operator |
| WMSE | Weighted Mean Squared Error |
| WNNM | Weighted Nuclear Norm Minimization |
| WRMSE | Weighted Root Mean Squared Error |

# Introduction

The two first sections of this introduction are written in collaboration with Julien Herman as we have shared the same topic for our master's theses: predicting the outcome of tennis matches. Julien focused on dynamical and optimization aspects [Her22], while my work investigates low-rank approaches.

## Problem relevance

Competitive sports have always been a source of motivation, unwinding, passion and amazement for humans. As far as in Ancient Greece, professional players were competing to win the Olympic Games. People have been supporting their favourite teams and players since then, trying to guess if they will be able to win a match, a tournament or even a championship. For the fans, guessing the outcome of the games played by their favourite contestant and his rivals is part of their DNA.

Moreover, it seems to be in the human nature to make predictions. Everyday, we try to guess the weather, how the day's meetings will unfold or which gifts will receive at our birthday. Of course, this is also true for the sports games of the week. Maybe this common behaviour could be due to the powerful feeling of being able to predict the future?

Besides the supportive, sportive and satisfactory aspects, the prediction of matches has another bigger motivation: money. Indeed, high-level sport has always worked in pair with financial incentive. In particular, bets make up a huge part of competitive sports. They already existed during the Antiquity [fra20] and they continue to have huge financial prospects. The extension of online betting has been a new driving force for the sector. In 2020 in France, the market for online betting was worth 5.3 billion euros and continues to grow every year [ANdJ21].

However, the main driving force of the researchers who have already given a try to this problem is the intellectual challenge: the problem carries a lot of difficulties, which are explained below. So, fortunately for the fans, the betters and the players, the sport will

always keep its part of unpredictability and randomness and will always stay interesting and enjoyable to follow and to play. This thesis is another attempt at the match prediction problem. In particular, three low-rank approaches are developed. Top-level male tennis is used as a case study.

## Challenges

The problem of guessing the result of a match when knowing the results of previous matches presents several difficulties. First of all, there is a variety of tools from completely different fields that could be used to state and model the problem. The choice of the hypotheses and of the angle of attack plays a crucial role in the final results. Unfortunately, there seems to be no absolute correct approach. Then, the quantity of information taken into account highly influences the solution. Features can be as varied as the outcome of the last games, the importance of the tournaments, the court's surface, the scores of previous games, the meteorological conditions, the financial incentives, the tiredness of players, and so on. The number of features that the model can handle can thus be a limiting factor, as well as the selection of the most pertinent ones. Another concern about the data is the uncertainty: some features could be subject to outliers or to noise. Finally, the consideration of prior knowledge and prejudice about the problem's variables can lead to significantly different results. How to model this prior knowledge wisely in order to improve results?

In addition to all those data-related concerns, once the modelling part is done, other issues appear about the solving part. The characteristics of the method can also influence the results. For example, convexity, size and smoothness of the optimization problem are critical. Moreover, the tuning of the hyper-parameters of the method could have a huge impact. Finally, defining correct quality criteria and error metrics is also tricky. Some metrics could be appropriate for some methods but not for others.

## Related works

Sports events based on past results are hard to predict in general [Buu]. Several techniques have already been used in order to predict the outcome of a tennis game. In 2015, a prediction of the outcome based on point-by-point data and stochastic models such as Markov chains was proposed by Martin Bevc. This approach is designed to predict the winner of a match at a given moment of the game. He showed that predicting the correct outcome of a game is increasingly more difficult the further from the end of the match we are since there are fewer data available [Bev15].

In 2017, Robin Praet investigated the possibility of predicting the outcome of sports events based on recommendation techniques and machine learning. He focused on tennis during his master's thesis and found that players are more predictable in Grand Slam tournaments compared to other competitions. The conclusion admits that predicting sports results remains a complicated area since the algorithms are not able to exceed the border of 70% [Pra]. In 2020, predicting tennis matches using machine learning was again explored. The main result is that the logistic regression model greatly outperforms the official ATP (Association of Tennis Professionals) ranking in terms of accuracy [DS20].

In 2019, a new ranking model combining the BTL model and nonnegative matrix factorization was proposed to predict the outcome of tennis games between top players. It shows that the surface is a key determinant of the performances of male players, but the effects of the surface are attenuated for females [XTFF19]. In 2019, different measures for forecasting tennis matches outcome were explored, such as ATP ranking, betting odds and Elo rating system. The paper also uses adjusted Elo ratings to predict the outcome of matches, which takes the skill of the player on specific surfaces and betting odds into account. They found that betting odds perform well on forecasting and adjusted Elo ratings are a better predictor for higher-ranked players [WLG19]. A network-based approach using Long-Short Term Memory was also explored, to predict the result of the next match by using teams' historical match data [ZZH+21]. Based on this technique, the accuracy of the predictions of the five next games of football teams was around 70% on average.

## Overview of the thesis

This thesis tackles the problem of matches outcome prediction with the study case of top male tennis. Tennis has some comfortable properties: there is no tie possibility in the outcome, it is an individual sport and therefore offers less variability through the years (in contrast to a team sport where the team composition is changing every year), and lots of matches data are easily available. The ATP Tour is composed each year of twenty or so tournaments, with the most important ones being the four famous Grand Slams [ATP22]. The dataset gathers 46 652 matches spanning from 2000 to 2016.

During this thesis, an important assumption is made: the existence of a true winning probability matrix, from which the dataset was created through sampling. The thesis develops low-rank approaches in order to estimate this matrix. The low-rank hypothesis means that there exists a hidden structure in the matrix which limits its number of degrees of freedom. The hope is to reveal some main types of players, or type of tournaments, which should give the same kind of results. The thesis is divided into 4 main chapters.

Chapter 1 of this thesis states the game outcome prediction problem that it seeks to tackle. Building on the main assumption of the existence of a true probability matrix generating the whole dataset, the problem is explained in general terms: finding back the matrix of winning probabilities between a pair of players $P$ from this generated dataset. Then, mathematical notations for all important objects are established. They will be needed to set up properly the different formulations coming in the following chapters. In the last part of the first chapter, the error metrics used throughout the work are described. The most important one is the prediction accuracy. It appears that in order to maximize this value, probabilities need to be clipped to zero or one.

Chapter 2 develops the low-rank matrix completion approach to solve the problem. The low-rank structure hypothesis is crucial for this work. It turns out that low-rank matrix completion is a domain which has been extensively researched. However, the main difficulty appears to be the building the incomplete matrix from the dataset, that will be completed afterwards. Several estimator choices are presented: maximum likelihood (ML), maximum *a posteriori* (MAP) and condition mean (CM). MAP and CM assume a prior distribution on the probabilities, some possibilities are detailed. It turns out that choosing a symmetric prior does not affect the prediction accuracy. Then, new matrix completion formulations are introduced, inspired by the state of the art and imposing the constraints inherent to the probability-based prediction problem. The last part of this chapter shows the results of experiments using classical and new low-rank matrix completion techniques.

Chapter 3 tackles the prediction problem directly, without creating an intermediate incomplete matrix. The approach used is the MAP estimation. The general formulation is discussed, as well as specific choices for the prior distribution. As this thesis always postulates a low-rank structure for the matrix of interest, two novel low-rank MAP formulations for the prediction problem are proposed. Only one of them has been successfully implemented, and is tested in the last section of this chapter.

Chapter 4 first presents a common model in order to make winning predictions: the Bradley-Terry-Luce model. This model assigns a rating $E$ to each player, from where it is possible to compute winning probabilities. This last chapter develops a way to extract those ratings from the dataset, based on a low-rank MAP approach on those $E$ ratings. What's more, these ratings can be differentiated by tournament to include more information about the problem. This novel formulation extends an ML technique on alternative nonnegative $S$ ratings proposed in [XTFF19], by including prior knowledge and removing nonnegativity constraints. Some concerns about the interpretability of the low-rank structure of $E$ are expressed. A block-coordinate descent (BCD) algorithm to solve this formulation is stated as well. Lastly, all methods and formulations mentioned above are tested and compared in the final section.

Appendices A to E gather all secondary, complicated or boring mathematical developments that are used throughout the thesis.

All Python codes used in this thesis are available here: [mas22].

# Chapter 1

# Problem statement

## 1.1 Basic assumptions

In order to tackle the problem of match prediction, this thesis is built upon basic simplifying assumptions:

1. There exists a winning probability matrix $\widetilde{P}$, called the *true probability matrix*, which contains the probabilities of players winning against other players. This matrix is constant, unknown and complete.

2. The dataset $\mathcal{D}$ was produced by sampling this true probability matrix $\widetilde{P}$ independently for each match.

These assumptions have the following consequences. Firstly, the outcome of each match from $\mathcal{D}$ (played between players $i$ and $j$) was determined only by $\widetilde{P_{ij}}$, the probability of $i$ being the winner (or equivalently by $P_{ji} = 1 - P_{ij}$, the probability of $j$ being the winner). In particular, it is not influenced by a lot of external parameters that could feel intuitively important. For example, it is independent of the surface on which they play, the importance of the tournament, the weather, the financial incentives, the age of players, the ranking of players, and so on. Secondly, as all matches are independent, the result of a match is not influenced by the previous matches they played, and it does not influence the outcome of future games. This means that the players' current state of fitness and their former confrontations are not taken into account. Implicitly, it neglects any notion of time as well. In summary, the only determining factors for the match outcome are the players' indices $i$ and $j$, because they fully determine $\widetilde{P_{ij}}$.

## 1.2 Big picture

Based on the mentioned assumptions, the goal is of course to recover this somewhat magical matrix $\widetilde{P}$ on the basis of the dataset $\mathcal{D}$, in other words, to find the matrix $P$ which best approximates (or ideally matches) the true matrix $\widetilde{P}$, given $\mathcal{D}$. Indeed, if we succeed, then we can predict in the best way possible the outcome of a new match according to the assumptions:

$$P \approx \widetilde{P}.$$

There are still two things to address. On the one hand, how to define what *best* means in this case: error metrics are therefore detailed in Sections 1.4 to 1.6. On the other hand, how to set up an optimization problem to find this best approximation, maybe using extra assumptions in the process. This leads to two types of problem that will be explored in this work: matrix completion (Chapter 2) and maximum *a posteriori* (MAP) estimation (Chapters 3 and 4).

## 1.3 Notation

We first set up some notations in order to properly define and derive the different problem formulations we want to tackle.

- $\mathcal{N}$ is the set of all players.
- $n = |\mathcal{N}| \in \mathbb{N}$ is the total number of players.
- $P \in \mathbb{R}^{n \times n}$ is the *probability matrix* variable that we want to find. The element $P_{ij}$ contains the predicted probability of winning for player $i$ over player $j$. This matrix is constrained by 3 *probability constraints*:

  1. *Bounds constraints.* $P_{ij}$ are bounded by definition of a probability:

  $$0 \leqslant P_{ij} \leqslant 1 \qquad \forall (i,j) \in (\mathcal{N} \times \mathcal{N}). \tag{1.1}$$

  2. *Winner constraints.* There is always a winner between $i$ and $j$, i.e. there is no draw possibility. This condition means that diag-symmetric entries always add up to 1:

  $$P_{ij} + P_{ji} = 1 \qquad \forall (i,j) \in (\mathcal{N} \times \mathcal{N}). \tag{1.2}$$

3. *Diagonal constraints.* We deduce the value of $P_{ii}$ for consistency, even if it is useless as a player will never play against himself:

$$P_{ii} = \frac{1}{2} \qquad \forall i \in \mathcal{N}. \tag{1.3}$$

Those conditions can be formulated in matrix form, with 0 and 1 being matrices of size $n \times n$ full of zeros and ones respectively:

$$\begin{aligned} P &\geqslant 0_{n\times n} \\ P &\leqslant 1_{n\times n} \\ P + P^\top &= 1_{n\times n}. \end{aligned}$$

- $\mathcal{P} \subset \mathbb{R}^{n\times n}$ is the set of all acceptable probability matrices, i.e. all matrices respecting the probability constraints. Therefore, $P \in \mathcal{P}$:

$$\begin{aligned} \mathcal{P} &= \left\{ P \in \mathbb{R}^{n\times n} : 0 \leqslant P_{ij} \leqslant 1, P_{ij} + P_{ji} = 1, P_{ii} = \frac{1}{2}, \forall (i,j) \in (\mathcal{N} \times \mathcal{N}) \right\} \\ &= \left\{ P \in \mathbb{R}^{n\times n} : 0_{n\times n} \leqslant P \leqslant 1_{n\times n}, P + P^\top = 1_{n\times n} \right\}. \end{aligned} \tag{1.4}$$

- $\widetilde{P} \in \mathcal{P}$ is the *true probability matrix*, complete but unknown, that we want to approach.

- $\mathcal{D}$ is the *dataset* containing all results of matches. $\mathcal{D}$ is randomly divided into two disjoint subsets:

$$\mathcal{D} = \mathcal{D}_{\text{tr}} \uplus \mathcal{D}_{\text{te}}.$$

- $\alpha_{\text{te}} \in ]0,1[$ is the *testing percentage*, i.e. the fraction of the dataset kept for testing purposes.

- $\mathcal{D}_{\text{tr}} \subset \mathcal{D}$ is the *training dataset* that is used to solve the problem and find $P$.

- $\mathcal{D}_{\text{te}} \subset \mathcal{D}$ is the *testing dataset* that is used to measure the quality of the constructed $P$.

- $d_{ij,k} \in \mathcal{D}$ is the result of the $k^{\text{th}}$ match played between player $i$ and player $j$ in the dataset $\mathcal{D}$. $d_{ij,k}$ can be interpreted as a realization of the random variable $D_{ij,k}$ as well. From assumption 2, we consider $D_{ij,k}$ as a Bernouilli random variable with parameter $\widetilde{P_{ij}}$, equal to 1 when player $i$ wins, and to 0 when player $j$ wins:

$$D_{ij,k} \sim \text{Ber}(\widetilde{P_{ij}}) \iff \begin{cases} \mathbb{P}[D_{ij,k} = 1] = \widetilde{P_{ij}}, \\ \mathbb{P}[D_{ij,k} = 0] = 1 - \widetilde{P_{ij}}. \end{cases}$$

We can notice that $D_{ji,k} = 1 - D_{ij,k} \sim \text{Ber}(1 - \widetilde{P_{ij}}) = \text{Ber}(\widetilde{P_{ji}})$ by symmetry.

- $\mathcal{D}_{ij} \subset \mathcal{D}$ is the set of all matches played between players $i$ and $j$:

$$\mathcal{D}_{ij} = \bigcup_k \{d_{ij,k} \in \mathcal{D}\}.$$

- $m_{ij} \in \mathbb{N}$ is the *number of matches* played between player $i$ and player $j$: $m_{ij} = |\mathcal{D}_{ij}| \geqslant 0$. Obviously, $m_{ij} = m_{ji}$ by symmetry. If $m_{ij} = 0$, we are in the case where no match has been played between players $i$ and $j$. Obviously, $m_{ii} = 0$ as a player cannot play against himself.

- $W \in \mathbb{R}^{n \times n}$ is the matrix called *confrontation matrix*. It is a nonnegative integer matrix that contains the number of wins by player $i$ over player $j$ in the element $w_{ij}$:

$$w_{ij} = \sum_{k=1}^{m_{ij}} d_{ij,k}.$$

  Trivially, we have $0 \leqslant w_{ij} \leqslant m_{ij}$ by definition, $w_{ji} = m_{ij} - w_{ij}$ by symmetry and $w_{ii} = 0$.

- $\Omega \subset (\mathcal{N} \times \mathcal{N})$ is the set of all pairs of players $(i, j)$ that have already played one against the other in the dataset $\mathcal{D}$:

$$\Omega = \{(i, j) \in (\mathcal{N} \times \mathcal{N}) : m_{ij} > 0\}.$$

- $\Omega^{\mathrm{c}} = (\mathcal{N} \times \mathcal{N}) \setminus \Omega$ is the complement set of $\Omega$. It contains all pairs of players that have never played one against the other. It could equivalently be defined as follows:

$$\Omega^{\mathrm{c}} = \{(i, j) \in (\mathcal{N} \times \mathcal{N}) : m_{ij} = 0\}.$$

## 1.4 Prediction accuracy metric $\mathcal{A}$

The first way of defining how well a method performs at predicting the outcome of matches consists in making predictions for the matches from the testing set and comparing it with the real data. This is the most intuitive metric, and it is the main metric used in this thesis.

For the prediction of the match $d_{ij,k} \in \mathcal{D}$ (the $k^{\text{th}}$ game between players $i$ and $j$ coming from the dataset $\mathcal{D}$), let us define the new binary random variable $C_{ij,k}$. It equals 1 if the prediction is correct and 0 otherwise. $c_{ij,k}$ is a realisation of this random variable $C_{ij,k}$. It is obvious that the prediction made depends on $P_{ij}$, and therefore $C_{ij,k}$ also depends on $P_{ij}$. The global prediction accuracy $\mathcal{A}$ on a dataset $\mathcal{D}$ is then defined as the ratio of the number of

correctly guessed games over the total number of matches. In order to not count any match twice, we impose $i > j$. This accuracy depends on $P$:

$$\mathcal{A}(P) = \frac{\#\text{ good guesses}}{\#\text{ matches}} = \frac{\displaystyle\sum_{\substack{(i,j)\in\Omega \\ i>j}} \sum_{k=1}^{m_{ij}} c_{ij,k}}{\displaystyle\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}}.$$

Obviously, this accuracy $\mathcal{A}(P)$ needs to be maximized by finding $P$ as optimally as possible. Intuitively, one could argue that if we find $P_{ij}$ for all $(i,j)$, which hopefully are close to the true $\widetilde{P_{ij}}$, then we should predict the victory of player $i$ with a probability $P_{ij}$ and his defeat with a probability $1 - P_{ij}$.

It turns out that this predictive pattern is suboptimal, even in the case of perfect reconstruction ($P_{ij} = \widetilde{P_{ij}}$). Note that all details concerning the developments and proofs about this prediction accuracy metric $\mathcal{A}$ are described in Appendix A. Let us first examine optimal solution on average, then the optimal solution experimentally.

The optimal solution on average consists of predicting that the winner is always the player with the highest victory probability, and never betting (even with a small probability) on the supposed weakest player. To make it formal, the *clipping operator* $\mathcal{C}$ on a probability $P_{ij}$ or on the whole probability matrix $P$ (by element-wise application) is defined as:

$${P_{ij}}^* = \left[\mathcal{C}(\widetilde{P})\right]_{ij} = \mathcal{C}(\widetilde{P_{ij}}) = \begin{cases} 0 & \text{if} \quad 0 \leqslant \widetilde{P_{ij}} < \frac{1}{2} \\ \frac{1}{2} & \text{if} \quad \widetilde{P_{ij}} = \frac{1}{2} \\ 1 & \text{if} \quad \frac{1}{2} < \widetilde{P_{ij}} \leqslant 1 \end{cases} \qquad \forall (i,j) \in (\mathcal{N} \times \mathcal{N}).$$

The optimal $P$ in order to maximize the experimental accuracy has a different expression, i.e. how should we choose $P$ such that the prediction accuracy of a particular dataset $\mathcal{D}_{\text{te}}$. It is linked to the maximum likelihood (ML) estimator $\overline{P_{ij}}^{\text{ML}} \approx \frac{w_{ij}}{m_{ij}}$, which will be developed in Section 2.5.2. This result indicates that the ML estimator is natural and even desirable. The optimal solution is given by:

$${P_{ij}}^* = \mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right) = \mathcal{C}\left(\overline{P_{ij}}^{\text{ML}}\right) \qquad \forall (i,j) \in \Omega.$$

Let us remark that in practice, the prediction accuracy is computed on $\mathcal{D}_{\text{te}}$, while $P$ is computed using $\mathcal{D}_{\text{tr}}$. Thus, the optimal solution ${P_{ij}}^* = \mathcal{C}\left(\overline{P_{ij_{\text{te}}}}^{\text{ML}}\right) = \mathcal{C}\left(\frac{w_{ij_{\text{te}}}}{m_{ij_{\text{te}}}}\right)$ is in reality not directly accessible. Hence, using the ML estimator on the training set is not necessarily enough. In order to maximize $\mathcal{A}(P)$ in practice, we need to find $P_{ij_{\text{tr}}}$ for all $(i,j)$ in $\Omega_{\text{te}}$ (at

least) such that:

$$\mathcal{C}\left(P_{ij_{\text{tr}}}\right) = \mathcal{C}\left(\overline{P_{ij_{\text{te}}}}^{\text{ML}}\right) \qquad \forall (i,j) \in \Omega.$$

Obviously, we can hope that $\mathcal{C}\left(\overline{P_{ij_{\text{tr}}}}^{\text{ML}}\right) = \mathcal{C}\left(\overline{P_{ij_{\text{te}}}}^{\text{ML}}\right)$ since $\overline{P_{ij_{\text{te}}}}^{\text{ML}}$ and $\overline{P_{ij_{\text{tr}}}}^{\text{ML}}$ are both estimators of $\widetilde{P_{ij}}$, which would motivate the use of the ML estimator nevertheless. Though, this desired equality is not guaranteed. Moreover, it does not apply when $(i,j) \notin \Omega_{\text{tr}} \cap \Omega_{\text{te}}$, therefore we necessarily need some clever method to guess those missing entries.

It is possible to show (see Appendix A) that there exists two fundamental upper bounds on the accuracy reachable in this framework, one in the average case, one for a given experiment:

$$\mathbb{E}\left[\mathcal{A}(P)\right] \leqslant \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(\frac{1}{2} + \left|\frac{1}{2} - \widetilde{P_{ij}}\right|\right)$$

$$\mathcal{A}(P) \leqslant \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(\frac{1}{2} + \left|\frac{1}{2} - \frac{w_{ij}}{m_{ij}}\right|\right).$$

It is often true that $w_{ij}$ and $m_{ij}$ for $(i,j) \in \Omega$ are small ($\leqslant 2$) in a real dataset $\mathcal{D}$. Therefore, the ratio $\frac{w_{ij}}{m_{ij}}$ is subject to a large variance and is biased towards 0 or 1. This allows the experimental accuracy to be better than the expected one.

From this analysis, a question comes in mind: why are professional betters then still betting on outsiders, is this not a losing strategy? The reason is that betters do not want to be right as often as possible, they aim to maximize their gains. Namely, the gain is higher if the weakest player wins as his odds are higher. Therefore, if the expected gain is higher when betting on the weakest player, they have every interest in betting on the weakest player.

## 1.5 Matrix completion metrics

The following metrics are standard but can only be applied when there is a matrix $\overline{P}$ to compare to: MSE, RMSE, Weighted MSE and Weighted RMSE. In other words, it is valid only for matrix completion methods from Chapter 2, not for the MAP approaches (Chapters 3 and 4).

### 1.5.1 MSE

The MSE is defined as the Mean Square Error. Because of the square, MSE does not have the same units as the variable it measures. For a dataset $\mathcal{D}$, it is defined as follows:

$$\text{MSE} = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left(\overline{P_{ij}} - P_{ij}\right)^2.$$

### 1.5.2 RMSE

The RMSE is defined as the Root Mean Square Error. Unlike the MSE, this metric shares the same units as the variable it measures (a probability measure). For a dataset $\mathcal{D}$, it is defined as follows:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left(\overline{P_{ij}} - P_{ij}\right)^2}.$$

### 1.5.3 Weighted MSE

The MSE and RMSE metrics have a flaw: each element $P_{ij}$ with $(i,j) \in \Omega$ contributes the same amount in the metric. However, in the dataset, the number of matches played by a pair of players $m_{ij}$ largely varies from pair to pair. Intuitively, the pairs which played more games should be predicted better. Therefore, the weighted MSE (WMSE) is defined in order to put more weight the more a pair had played together:

$$\text{WMSE} = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\Omega} m_{ij} \left(\overline{P_{ij}} - P_{ij}\right)^2.$$

### 1.5.4 Weighted RMSE

The weighted RMSE (WRMSE) is the square root of WMSE. WRMSE has the advantage over WMSE to share units of a probability.

$$\text{WRMSE} = \sqrt{\text{WMSE}} = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\Omega} m_{ij} \left(\overline{P_{ij}} - P_{ij}\right)^2}. \tag{1.5}$$

## 1.6 Other metrics

### 1.6.1 Rank

The rank of a matrix $A$, denoted $\text{rank}(A)$, provides important information about the matrix $A$, in particular the number of "degrees of freedom" of the matrix. Mathematically,

it is defined as the number of linearly independent columns (or rows) of the matrix $A$. This consideration is especially critical in the context of low-rank approaches developed in this thesis (see Section 2.3). This low-rank structure implies that the matrix has either its rank minimized, either its rank bounded, or its rank fixed, as explained in Section 2.4. It is important to verify if those constrained are satisfied or not.

### 1.6.2 Singular values distribution

Knowing the distribution of singular values $\sigma_i(A) \geqslant 0$ yields even more information than the rank alone. Indeed, via the SVD decomposition, the matrix can be seen as a weighted sum of rank 1 matrices orthogonal to each other, and singular values tell about the amplitude of those rank 1 components. Those components can be seen as the degrees of freedom of the matrix. The rank can even be defined as the number of non-zero singular values: $\text{rank}(A) = |\{\sigma_i(A) > 0\}|$ [Cle21]. What is more, the distribution of $\sigma_i(A)$ informs about "how much low-rank" a matrix is. For example, if there exists a huge gap in size between the first and second half of the singular values (all are strictly positive), the matrix still carries a low-rank structure, despite being full-rank.

In order to study the distribution of the $P$ matrix's singular values, it is useful to separate $P$ into its symmetric and skew-symmetric parts: $P = \frac{1}{2}1_{n\times n} + P'$. $1_{n\times n}$ is the ones matrix of size $n \times n$, it is of rank 1 and has its unique singular value equal to $\frac{n}{2}$. Suppose $r = \text{rank}(P')$. As the matrix $P'$ is skew-symmetric, its singular values come in pairs and its SVD decomposition can be written as a sum of $\frac{r}{2}$ rank 2 skew-symmetric components: $P' = \sum_{k=1}^{\frac{r}{2}} \sigma_{2k}(u_{2k}{v_{2k}}^\top - v_{2k}u_{2k}^\top)$. All these facts are proven in Appendix B.

Combining the singular values distributions of $P'$ and $\frac{1}{2}1_{n\times n}$, we have that $P$ has $\sigma_1 \approx \frac{n}{2}$ coming from its symmetric shift, and the rest has globally the same shape as the distribution of $P'$.

### 1.6.3 Running time

This metric is self-explanatory. Each method tried takes some time in order to compute a solution. Depending on the complexity of the method and the structure of the optimization problem, the running time can vary a lot.

### 1.6.4 Violation of constraints

There are 4 types of constraints that need to be verified in order for the solution of any method to be acceptable. The idea is to simply compute the percentage of those constraints

that are violated (or relaxed) by the solution. During experiments, some constraints are often imposed afterwards in order to be able to compute other metrics such as $\mathcal{A}$.

1. Bounds constraints:

$$0 \leqslant P_{ij} \leqslant 1 \qquad \forall (i,j) \in (\mathcal{N} \times \mathcal{N}).$$

2. Winner constraints:

$$P_{ij} + P_{ji} = 1 \qquad \forall (i,j) \in (\mathcal{N} \times \mathcal{N}).$$

3. Diagonal constraints:

$$P_{ii} = \frac{1}{2} \qquad \forall i \in \mathcal{N}.$$

4. Rank constraint:

$$\text{rank}(P) \leqslant r.$$

   If the method imposes a limit $r$ on the rank, then the solution should also respect this constraint. In most cases, the clipping operation $\mathcal{C}$ destroys this rank constraint.

# Chapter 2

# Low-rank matrix completion on probability matrix

## 2.1 Big picture

A first way to tackle the prediction problem is to set up a matrix completion problem. The idea is to assume that the fundamental matrix $\widetilde{P}$ has an inner structure, which means that its entries are correlated. Often, like in this thesis, the inner structure is considered to be a low-rank structure. Then, an incomplete matrix $\overline{P} \in \mathcal{P}$ is computed directly from $\mathcal{D}$ via some chosen function $g(\mathcal{D})$, derived from some estimators. The incompleteness of $\overline{P}$ comes from the fact that in general, some players have not played against each other in $\mathcal{D}$. For the entries where we have data (corresponding to $(i,j) \in \Omega$), we expect $\overline{P_{ij}} \approx \widetilde{P_{ij}}$ as $\mathcal{D}$ is assumed to be sampled from $\widetilde{P}$. $\overline{P_{ij}}$ entries have to be computed carefully because most of them rely on a really small set of data ($m_{ij}$ is small), so they are subject to a lot of uncertainty. Afterwards, we try to *complete* the matrix $\overline{P}$, giving the matrix $P$. For $(i,j) \in \Omega$, the completion process aims to find $P_{ij} \approx \overline{P_{ij}}$ (and thus $P_{ij} \approx \widetilde{P_{ij}}$). For $(i,j) \in \Omega^c$, the method guesses $P_{ij}$ by exploiting the inner structure of $\widetilde{P}$ which $\overline{P}$ and $P$ should inherit, hoping that $P_{ij} \approx \widetilde{P_{ij}}$ as well. Afterwards, $P_{ij}$ are clipped to $\mathcal{C}(P_{ij})$ to maximize the prediction accuracy.

## 2.2 Notation

In order to properly define the completion matrix problem, we need some extra notation.

- $\overline{P} \in \mathcal{P}$ is the *incomplete probability matrix* that we can partially compute and that we want to complete. The function $g(\mathcal{D})$ need to be designed. Several estimators are proposed in Section 2.5. An intuitive choice is the maximum likelihood (ML) estimator

$\overline{P_{ij}}^{\mathrm{ML}} = g_{ij}(\mathcal{D})^{\mathrm{ML}} = \frac{w_{ij}}{m_{ij}}, \forall (i,j) \in \Omega$. For $(i,j) \in \Omega^{\mathrm{c}}$, although having no real influence, a value need to be chosen. To make the parallel with the coding part of this work, $\overline{P_{ij}} = \mathrm{Nan}$, were Nan is not a number, a way to store a hole in Python:

$$\overline{P_{ij}} = \begin{cases} g_{ij}(\mathcal{D}) & \text{if } (i,j) \in \Omega, \\ \mathrm{Nan} & \text{if } (i,j) \in \Omega^{\mathrm{c}}. \end{cases}$$

- $B_\Omega \in \{0,1\}^{n \times n}$ is a binary mask for the known entries:

$$[B_\Omega]_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \Omega, \\ 0 & \text{if } (i,j) \in \Omega^{\mathrm{c}}. \end{cases}$$

- $\mathrm{P}_\Omega$ is called the *projection operator*. It takes a matrix $A$ as input and it outputs the same matrix with 0 at the entries in $\Omega^{\mathrm{c}}$. This projection operation is linear as it can be rewritten using a linear element-wise matrix (Hadamard) product $\odot$ with the binary mask $B_\Omega$:

$$[\mathrm{P}_\Omega(A)]_{ij} = [B_\Omega \odot A]_{ij} = \begin{cases} A_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{if } (i,j) \in \Omega^{\mathrm{c}}. \end{cases}$$

  This reformulation is useful for the implementation. Notice that we define $0.\mathrm{Nan} = 0$.

## 2.3 Low-rank structure

In order to guess the missing entries, we necessarily need to assume that they are somehow related to the known entries. If it was not the case, then the completion process would not do any better than random guessing. This is encoded into an inner structure of the matrix.

During this thesis, this inner structure is assumed to be a low-rank structure. Let $\mathrm{rank}(A)$ be the rank of the matrix $A \in \mathbb{R}^{m \times n}$. It is defined as the number of linearly independent columns (or rows) of the matrix. The rank is considered low:

$$\mathrm{rank}(A) \ll \min\{m,n\}.$$

Mathematically, it means that all columns of $A$ are linear combinations of $\mathrm{rank}(A)$ linearly independent columns. The low-rank assumption is the main structure used for matrix completion and has given rise to the low-rank matrix completion (LRMC) field. This area has been active for over 15 years, therefore low-rank matrix completion is well understood by the scientific community.

An additional advantage of the low-rank structure is that it carries some physical sense in plenty of real applications. In our case, the low-rank structure of $P$ is somewhat hard to interpret. There is intuitively the feeling that some types of players exist (strong on clay courts, strong on hard courts, offensive players, defensive players, left-handed players,...) and that each player is a combination of those *meta-players*. Thus, a guess is that it should somehow be reflected in the winning probabilities. However, it stays unclear why this idea should translate into a low-rank structure for this matrix $\widetilde{P}$. It would be nice to be able to verify this low-rank assumption by examining the singular values distribution of $\widetilde{P}$. Unfortunately, assuming this matrix truly exists, it will forever stay unknown. Therefore, it is not possible to confirm that this low-rank assumption is correct.

### 2.3.1 Rank fixed

A first way to represent this low-rank matrix $A$ is to consider it as an element of the set of all matrices with a given rank $r$. This set is known as the *Riemannian manifold of fixed-rank matrices* [AAM14]:

$$\mathbb{R}_r^{m\times n} = \{A \in \mathbb{R}^{m\times n} : \operatorname{rank}(A) = r\}.$$

A downside of this viewpoint is that this manifold is not a convex set and thus not a vector space. Therefore, optimization on this set cannot be tackled with the results of convex optimization and requires specific tools such as Riemannian optimization.

A nice implication of the low-rank property with known rank $r$ is that we can factorize the matrix $A$ in two full-rank $r$ low-dimension matrices, see Appendix B.5. If we define $U \in \mathbb{R}_r^{m\times r}$ and $V \in \mathbb{R}_r^{n\times r}$, then the Bilinear Matrix Factorization (BMF) reads:

$$A = UV^\top.$$

So, the manifold of fixed rank matrices can be equivalently defined by its factorized form:

$$\mathbb{R}_r^{m\times n} = \{UV^\top : U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{n\times r}, \operatorname{rank}(U) = r, \operatorname{rank}(V) = r\}.$$

This factorization is not unique. It is defined up to a invertible matrix of size $r$, see Appendix B.6. Therefore, some conditions can be imposed on $U$ or on $V$ if needed in our formulations without losing this factorization, as long as there exists an invertible matrix transforming a valid factorization into a new one which satisfies the additional constraints. In particular, $U$ can be imposed to have orthogonal columns: $U^\top U = I_r$. Notice that this specific factorization is not unique either, see Appendix B.7.

### 2.3.2 Rank bounded

The rank constraint can be relaxed in order to handle a more manageable set for the optimization: $\text{rank}(A) \leqslant r$. Indeed, its low-dimensional BMF $UV^\top$ also exists, but this time, the factors are not constrained. Even if $U$ and $V$ both live in vector spaces which are convex sets ($\mathbb{R}^{m\times r}$ and $\mathbb{R}^{n\times r}$), the subset of $\mathbb{R}^{m\times n}_{\leqslant r}$ containing $A$ is not convex. For the sake of precision, we use the term *determinantal variety* [OGA22]:

$$\mathbb{R}^{m\times n}_{\leqslant r} = \{A \in \mathbb{R}^{m\times n} : \text{rank}(A) \leqslant r\} = \{UV^\top : U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{n\times r}\}.$$

This decomposition is often used in the optimization methods for matrix completion as the factors have a smaller size than the initial matrix and because the bounded rank constraint naturally arises while optimizing in conventional vector spaces.

## 2.4 Literature review

Matrix completion is a topic that has been deeply studied for several years since the famous Netflix Prize spanning from 2006 to 2009 [Net09]. The goal of this competition was to predict the rating (integer between 1 and 5) that users would give to films they have never seen before. Precisely, the goal was to minimize the RMSE [Net22]. This problem is the prototype of matrix completion. Indeed, the goal is to complete the score matrix where $s_{ij}$ is the rating given by the user $i$ to the film $j$, knowing a part of the entries. Moreover, it seems to be a case for Low-rank Matrix Completion (LRMC). There is a clear low-rank structure: users should act like a composition of meta-users and therefore give the same kind of notes to similar films. For example, users who like romance and action films should note films in the same way. In the best case scenario, the low-rank structure should make the meta-players appear.

For this overview of the literature of LRMC, let $X \in \mathbb{R}^{m\times n}$ be the true matrix that we want to find back. Let $\text{P}_\Omega(X) \in \mathbb{R}^{m\times n}$ be the incomplete matrix with some entries known from the dataset. Let $M \in \mathbb{R}^{m\times n}$ be the variable matrix which is reconstructed during the matrix completion procedure.

Low-rank matrix completion techniques are summarized in several literature reviews on which this review is based [CW22] [LHSZ19] [NKS19] [RYL+18]. Here is a non-exhaustive summary of the techniques preponderant in the literature and/or relevant to this work. They can be split into three main categories: techniques with no rank information (leading to rank minimization), techniques with a bound on the rank and techniques with a rank fixed. One last type of completion worth mentioning here is the nonnegative matrix completion, where the matrix is additionally constrained to be nonnegative.

### 2.4.1 No rank information

**Rank minimization**. The most basic idea in order to work in the low-rank framework is to simply minimize the rank of the reconstructed matrix $M$, which require no prior information about it. To set up the problem correctly, entries $M_{ij}$ with $(i,j) \in \Omega$ are imposed to match the known entries [CR08]:

$$\begin{aligned} &\min_{M} \ \operatorname{rank}(M) \\ &\mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X). \end{aligned} \tag{2.1}$$

The constraint $\mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X)$ is coming back later but can be a bit too restrictive. On the one hand, values $[\mathrm{P}_\Omega(X)]_{ij}$ can be altered by some noise, so it is not necessarily wanted to have exact equality. On the other hand, strict equalities can heavily restrict the minimum rank. Therefore, a relaxation of this constraint can be used in all formulations mentioned in this Section:

$$\|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_F \leqslant \delta.$$

This relaxation uses the Frobenius norm $\|A\|_F = \sqrt{\operatorname{tr}(A^*A)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|A_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$ and a small $\delta > 0$ that needs to be fixed, which is in itself a hard problem.

Another problem with formulation 2.1 is that the objective function $\operatorname{rank}(M)$ is not convex, unlike the constraints. Convex optimization results can thus not be used. What is more, it has been proven to be an NP-hard problem [Faz02].

**Nuclear norm minimization (NNM)**. The solution is to consider the convex envelope of the rank as objective function. It is called the nuclear norm [Faz02]:

$$\|A\|_* = \operatorname{tr}\left((A^*A)^{\frac{1}{2}}\right) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(A).$$

It is interesting to define the Schatten $p$-norm, as they generalize the nuclear and Frobenius norms:

$$\|A\|_{S_p} = \left(\sum_{i=1}^{\min\{m,n\}} \sigma_i(A)^p\right)^{\frac{1}{p}}.$$

In particular, $\|A\|_* = \|A\|_{S_1}$ and $\|A\|_F = \|A\|_{S_2}$ [Sch22]. This will be useful for later formulations.

Here is the new NNM convex formulation:

$$\min_{M} \|M\|_* \tag{2.2}$$

$$\mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X).$$

**Semidefinite programming NNM**. Problem 2.2 can be rewritten as a semidefinite program, for which special algorithms exist [CR08]. It uses two additional positive semidefinite matrices $W_1 \in \mathbb{R}^{m\times m}$ and $W_2 \in \mathbb{R}^{n\times n}$:

$$\begin{aligned} \min_{M,W_1,W_2} \ & \mathrm{tr}(W_1) + \mathrm{tr}(W_2) \\ & \mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X) \\ & \begin{pmatrix} W_1 & X \\ X^\top & W_2 \end{pmatrix} \succcurlyeq 0 \\ & W_1 \succcurlyeq 0 \\ & W_2 \succcurlyeq 0. \end{aligned} \tag{2.3}$$

**Iteratively Reweighted Least Squares (IRLS)**. Some papers propose to rewrite the problem 2.2 in order to tackle it with IRLS schemes [FRW11] [MF12]:

$$\begin{aligned} \min_{M,W} \ & \|W^{\frac{1}{2}} M\|_F^2 \\ & W = (MM^*)^{-\frac{1}{2}} \\ & \mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X). \end{aligned} \tag{2.4}$$

**NNM with regularization**. A standard technique to avoid overfitting is to add a regularization term to the NNM problem 2.2, for example, the Frobenius norm of the matrix [CCS08]. The regularization term is chosen such that the problem stays convex. We define a trade-off parameter $\tau > 0$. If $\tau \to \infty$, then the formulation reduces to problem 2.2. It reads:

$$\begin{aligned} \min_{M} \ & \frac{1}{2}\|M\|_F^2 + \tau\|M\|_* \\ & \mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X). \end{aligned} \tag{2.5}$$

A common method to solve such a constrained problem is to minimize the unconstrained lagrangian, with the Langrangian multiplier $Y$ and where $\langle A, B\rangle = \sum_{i=1}^m \sum_{j=1}^n A_{ij}B_{ij}$ designate the Frobenius inner product:

$$\min_{M,Y} \mathcal{L}(M, Y) = \min_{M,Y} \ \frac{1}{2}\|M\|_F + \tau\|M\|_* + \langle Y, \mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\rangle.$$

In [CCS08], the lagrangian formulation is solved using the Singular Value Thresholding (SVT) algorithm.

**LASSO relaxation**. The strong constraint $\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)$ from problem 2.2 can be shifted to the objective as a penalty function:

$$\min_M \frac{1}{2}\|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_F^2 + \tau\|M\|_*. \tag{2.6}$$

This relaxation is solved for example by the SoftImpute algorithm [MHT10]. It is called LASSO matrix completion because it is similar to the standard least absolute shrinkage and selection operator (LASSO) [TY10] [MGC11]. Recall what the LASSO regression for finding the solution to the linear system $Ax = b$ is: $\min_x \ \frac{1}{2}\|Ax - b\|_{\ell_2}^2 + \tau\|x\|_{\ell_1}$, where $\|x\|_{\ell_p} = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ is the standard vector $p$-norm [LAS22]. The parallel with the LASSO regression is now obvious: the standard $\ell_p$ norms have been replaced with the Schatten $S_p$ norms. The Frobenius norm term produces a least squares regression while the nuclear norm term acts as a shrinkage regularizer, reducing the (convex envelope of the) rank and therefore limiting the number of degrees of freedom.

**Robust Principal Component Analysis (RPCA)**. The rest of the techniques without rank information try to tackle the completion problem in presence of noise on the entries $\mathrm{P}_\Omega(X)$. The problem 2.2 can be equivalently reformulated in order to resemble the usual RPCA problem for complete perturbed matrices [LCM13]. It introduces a noise matrix $S \in \mathbb{R}^{m\times n}$ which is imposed to be sparse:

$$\begin{aligned} &\min_{M,S} \|M\|_* \\ &M + S = \mathrm{P}_\Omega(X) \\ &\mathrm{P}_\Omega(S) = 0. \end{aligned} \tag{2.7}$$

**Weighted NNM (WNNM)**. The idea behind [GXM+17] is to add weights on the singular values in order to control how they shrink. The weighted nuclear norm is defined with a given positive vector $w \geqslant 0$:

$$\|A\|_{w,*} = \sum_{i=1}^{\min\{m,n\}} w_i\sigma_i(A).$$

This leads to the next formulation to tackle the robust matrix completion problem:

$$\begin{aligned} &\min_{M,S} \|M\|_{w,*} \\ &M + S = \mathrm{P}_\Omega(X) \\ &\mathrm{P}_\Omega(S) = 0. \end{aligned} \tag{2.8}$$

### 2.4.2 Bound on the rank

**Frobenius Norm Minimization (FNM)**. If an upper bound $r$ is given for the rank of the matrix to reconstruct, then it becomes a constraint of our problem and is therefore removed from the objective function. A new valid objective function is the error on known entries, computed with a Frobenius norm. This formulation is known as Frobenius Norm Minimization (FNM) or as the minimum rank approximation problem [LB09]:

$$\begin{aligned} &\min_{M} \frac{1}{2}\|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_F^2 \\ &\text{rank}(M) \leqslant r. \end{aligned} \tag{2.9}$$

A method to solve this problem is called IterativeSVD [TCS+01].

**Weighted FNM (WFNM)**. In some cases, elements of $\mathrm{P}_\Omega(X)$ are known with a higher confidence than others, i.e. they are less subject to noise. This leads to the idea that those more certain entries should be more important to reproduce precisely during the completion process than the less certain ones. This is called *weighted matrix completion*. A weight matrix $W \in \mathbb{R}^{m\times n}$ with $W_{ij} > 0$ for $(i,j) \in \Omega$ is defined in order to create the weighted Frobenius norm: $\|A\|_{F,W}^2 = \|W^{(1/2)} \odot A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n W_{ij} A_{ij}^2$, where $W^{(1/2)}$ is the element-wise square root matrix of $W$. This leads to the following formulation [FNP+19]:

$$\begin{aligned} &\min_{M} \frac{1}{2}\|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_{F,W}^2 \\ &\text{rank}(M) \leqslant r. \end{aligned} \tag{2.10}$$

**Bilinear Matrix Factorization (BMF)**. The factorization explained in Section 2.3 allows to write $M = UV^\top$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ have by construction their ranks constrained [HH09]. Problem 2.9 can be rewritten as problem 2.11. Even if this avoids the constraint $\text{rank}(M) \leqslant r$ in the optimization problem, it deteriorates the objective function as it becomes non-convex in $(U, V)$ while still block-component-wise convex:

$$\min_{U,V} \frac{1}{2}\|\mathrm{P}_\Omega(UV^\top) - \mathrm{P}_\Omega(X)\|_F^2. \tag{2.11}$$

The solution to this non-convexity is to optimize alternatively on $U$ and $V$. Indeed, if all but one factor are fixed, the objective stays convex (block-component-wise convexity). Some methods exist, such as the Alternating Minimization for matrix Completion (AltMinComplete) [JNS12] and the Alternating Steepest Descent (ASD) [TW16].

The BMF formulation deteriorates the problem in another way. It has been shown that all local minima of formulation 2.9 are stationary points of this formulation 2.11. However, the opposite is not necessarily true, meaning that there could exist stationary points of 2.11

towards which the mentioned algorithms could converge that are not local minimizers of 2.9 [HLB20].

A similar problem is solved by another alternating method, the LMaFit method [WYZ12]:

$$\begin{aligned} &\min_{U,V,Z} \frac{1}{2}\|UV^\top - Z\|_F^2 \\ &\mathrm{P}_\Omega(Z) = \mathrm{P}_\Omega(X). \end{aligned} \tag{2.12}$$

**Mixed NNM and BMF**. The NNM and BMF approaches are combined in [CDlTCB13]. It uses the variational definition of the nuclear norm: $\|Z\|_* = \min_{\substack{U,V \\ Z=UV^\top}} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$. For the trade-off parameter $\tau > 0$, the problem then reads:

$$\min_{U,V} \frac{1}{2}\|\mathrm{P}_\Omega(UV^\top) - \mathrm{P}_\Omega(X)\|_F^2 + \frac{\tau}{2}\left(\|U\|_F^2 + \|V\|_F^2\right). \tag{2.13}$$

**$\ell_p$-norm minimization**. There is *a priori* no reason to stick to the Frobenius norm. On the contrary, it is well-known that regression with $\ell_p$ (pseudo-)norms for $0 < p \leqslant 2$ are more robust against outliers. Let us recall that $\ell_p$ norms on matrices (also called $L_{p,p}$ norms) are defined as : $\|A\|_p = \left(\sum_{i=1}^m \sum_{j=1}^n |A_{ij}|^p\right)^{\frac{1}{p}}$. The norm is convex for $1 \leqslant p \leqslant 2$. This leads to the following formulation [ZS18], adaptation of problem 2.11:

$$\min_{U,V} \frac{1}{2}\|\mathrm{P}_\Omega(UV^\top) - \mathrm{P}_\Omega(X)\|_p^p. \tag{2.14}$$

**Alternating Projections (AP)**. This projection technique does not use the bilinear matrix factorization. The idea is instead to project alternatively on two sets to which the solution should belong. The algorithm stops when it finds their intersection as the solution should belong to both sets simultaneously. On the one hand to the determinantal variety ($M \in S_r = \mathbb{R}_{\leqslant r}^{m\times n}$) and on the other hand to the fidelity constraint set ($M \in S_p$, where $\delta_p$ needs to be chosen):

$$\begin{aligned} S_r &= \mathbb{R}_{\leqslant r}^{m\times n} = \{A \in \mathbb{R}^{m\times n} : \mathrm{rank}(A) \leqslant r\} \\ S_p &= \{A \in \mathbb{R}^{m\times n} : \|\mathrm{P}_\Omega(A) - \mathrm{P}_\Omega(X)\|_p^p \leqslant \delta_p\}. \end{aligned} \tag{2.15}$$

**Truncated Nuclear Norm Regularization (TNNR)**. NNM reduces all singular values. However, it is in some applications appropriate to not focus on reducing the first $r$ singular values, but only the smaller ones. We then define the truncated nuclear norm:

$$\|A\|_r = \sum_{i=r+1}^{\min\{m,n\}} \sigma_i(A) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(A) - \sum_{i=1}^{r} \sigma_i(A) = \|A\|_* - \sum_{i=1}^{r} \sigma_i(A).$$

By the SVD decomposition of $A$ we have $A = U\Sigma V^\top$ and thus $\Sigma = U^\top AV$. Moreover, $\sum_{i=1}^{r} \sigma_i(A)$ is equal to the trace of the truncated diagonal matrix of singular values $\Sigma_r$. Yet, $\Sigma_r = U_r^\top AV_r$, where the orthogonal matrices $U_r$ and $V_r$ are the matrices $U$ and $V$ truncated after the $r^{\text{th}}$ column. We have therefore the equality $\sum_{i=1}^{r} \sigma_i(A) = \max_{U,V} \operatorname{tr}(U^\top MV)$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ such that $U^\top U = V^\top V = I_r$. In the end, the minimization of the truncated nuclear norm is [HZY+13]:

$$\begin{aligned} &\min_{M} \left( \|M\|_* - \max_{U,V} \operatorname{tr}(U^\top MV) \right) \\ &\mathrm{P}_\Omega(M) = \mathrm{P}_\Omega(X). \end{aligned} \tag{2.16}$$

### 2.4.3 Rank fixed

**Optimization over the manifold of fixed-rank matrices**. When the rank is fixed, then matrix $M$ belongs to the manifold of fixed-rank matrices $\mathbb{R}_r^{m\times n}$. The easiest formulation consists in adapting 2.9:

$$\min_{M\in\mathbb{R}_r^{m\times n}} \|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_F^2. \tag{2.17}$$

This problem can be tackled using tools from Riemannian optimization [AMS08]. For example, [Van12] uses the SVD factorization of any matrix in $\mathbb{R}_r^{m\times n}$ in order to apply a Riemannanian non-linear conjugate gradients algorithm.

**Optimization over the Grassmann manifold**. The manifold of orthogonal matrices, also called the Stiefel manifold, is a subset of the manifold of fixed-rank matrices: $\mathrm{St}(m,r) = \{Q \in \mathbb{R}^{m\times r} : Q^\top Q = I_r\}$ [BZA20]. As discussed in Section 2.3, any matrix $M \in \mathbb{R}_r^{m\times n}$ can be factorized $M = QR^\top$, with $Q \in \mathrm{St}(m,r)$ and $R \in \mathbb{R}^{n\times r}$.

Problem 2.17 can be expressed using this orthogonal decomposition as follows [DM10]:

$$\min_{Q\in\mathrm{St}(m,r)} \left( \min_{R\in\mathbb{R}^{n\times r}} \|\mathrm{P}_\Omega(QR^\top) - \mathrm{P}_\Omega(X)\|_F^2 \right). \tag{2.18}$$

We observe that the function $f(Q) = \min_{R\in\mathbb{R}^{n\times r}} \|\mathrm{P}_\Omega(QR^\top) - \mathrm{P}_\Omega(X)\|_F^2$ has a minimum when $f(Q) = 0$. Therefore, the goal of this formulation is to find $Q \in \mathrm{St}(m,r)$ such that $f(Q) = \min_{R\in\mathbb{R}^{n\times r}} \|\mathrm{P}_\Omega(QR^\top) - \mathrm{P}_\Omega(X)\|_F^2 = 0$. This root-finding problem is unconstrained on the Stiefel manifold, but unfortunately, the solution is not unique due to the non-uniqueness of the $M = QR^\top$ decomposition.

This is reflected in the fact that for any $C \in \mathbb{R}^{r\times r}$ orthogonal ($C^\top C = CC^\top = I_r$), we have $f(Q) = f(QC)$. This leads to the realization that the value of $f$ only depends on $\mathrm{span}(Q)$, the space spanned by the columns of $Q$. By definition, this set $\mathrm{span}(Q)$ is a

subspace of dimension $r$ of $\mathbb{R}^m$. The Grassmann manifold is then introduced: $\mathrm{Gr}(m, r)$ is set of all $r$-dimensional linear subspaces in $\mathbb{R}^m$. A shift of perspective allows to search for the right subspace $\mathrm{span}(Q) \in \mathrm{Gr}(m, r)$ such that $f(Q) = 0$. The problem is then unconstrained on the Grassmann manifold [DM10].

### 2.4.4 Nonnegative matrix completion

**Nonnegative LASSO**. If the matrix $M$ is known to have only nonnegative entries, then it can be useful to add this convex constraint into the optimization problem [ZCW18]. For example, adapting 2.6 keeps a convex problem:

$$\begin{aligned} \min_{M} \ & \frac{1}{2}\|\mathrm{P}_\Omega(M) - \mathrm{P}_\Omega(X)\|_F^2 + \tau\|M\|_* \\ & M \geqslant 0. \end{aligned} \tag{2.19}$$

**Based on Nonnegative Matrix Factorization (NMF)**. Another well-researched topic is the Nonnegative Matrix Factorization problem. It consists in finding a decomposition of any nonnegative matrix $A \in \mathbb{R}^{m\times n}$ such that $A = UV^\top$, where $U \in \mathbb{R}^{m\times r}$, $U \geqslant 0$ and $V \in \mathbb{R}^{n\times r}$, $V \geqslant 0$. It can also be denoted $U \in \mathbb{R}_+^{m\times r}$ and $V \in \mathbb{R}_+^{n\times r}$. $U$ and $V$ are two nonnegative factors and where $r \leqslant \min\{m, n\}$ is the (nonnegative) rank. Even if the NMF is not unique, it has the asset to be easily interpretable: $U$ is a dictionary and $V$ is a coefficient matrix [XYWZ12].

This NMF is very similar to the BMF and can be exploited for nonnegative matrix completion formulations with a given rank $r$:

$$\begin{aligned} \min_{U,V} \ & \frac{1}{2}\|\mathrm{P}_\Omega(UV^\top) - \mathrm{P}_\Omega(X)\|_F^2 \\ & U \geqslant 0 \\ & V \geqslant 0. \end{aligned} \tag{2.20}$$

## 2.5 Estimators for the matrix to complete

### 2.5.1 Motivation of estimators

The entries $\overline{P_{ij}}$ need to be computed $\forall (i, j) \in \Omega$ in order to set-up the matrix completion problem properly. This is linked with the choice of the functions $g_{ij}(\mathcal{D})$. The field of statistics has developed specific tools in order to estimate as best as possible an unknown variable from some data samples: estimators. This step can thus be understood as finding wisely the estimator $\overline{P_{ij}} = g_{ij}(\mathcal{D})$ of the true value $\widetilde{P_{ij}}$ from $\mathcal{D}$.

It is clear that any estimator $\overline{P_{ij}}$ should respect the probability constraints for $(i,j) \in \Omega$ ($\overline{P} \in \mathcal{P}$, see equation 1.4), so the functions $g_{ij}$ need to agree with those constraints.

A key observation useful to compute estimators is that the element $\widetilde{P_{ij}}$ only influences the outcomes of the matches $d_{ij,k} \in \mathcal{D}_{ij}$. The only games that could possibly be linked with $\widetilde{P_{ij}}$ and on which $\widetilde{P_{ij}}$ could depend are the ones in $\mathcal{D}_{ij}$. Therefore, it seems logical that $\overline{P_{ij}}$ should only be a function of those matches only:

$$\overline{P_{ij}} = g_{ij}(\mathcal{D}) = g_{ij}(\mathcal{D}_{ij}).$$

### 2.5.2 Maximum likelihood (ML) estimator

The likelihood function $\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij})$ of $\mathcal{D}_{ij}$ is defined as the probability of generating the dataset $\mathcal{D}_{ij}$ assuming the model $\widetilde{P_{ij}}$: $\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij}) = \mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}]$. This function depends on $\widetilde{P_{ij}}$. The maximum likelihood (ML) estimator for $\widetilde{P_{ij}}$ corresponds to the value of $\widetilde{P_{ij}}$ that maximizes the likelihood:

$$\begin{aligned}\overline{P_{ij}}^{\mathrm{ML}} &= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmax}} \mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij}) \\ &= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmin}} -\ln(\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij})).\end{aligned} \tag{2.21}$$

Let us compute the likelihood function in the context of matches prediction. Using the assumptions of Section 1.1, matches are supposed independent. By also remembering that $D_{ij,k}$ are considered as Bernouilli random variables:

$$\begin{aligned}\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij}) &= \mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}] \\ &= \mathbb{P}\left[\bigcap_{d_{ij,k}\in\mathcal{D}_{ij}} d_{ij,k}|\widetilde{P_{ij}}\right] \\ &= \prod_{d_{ij,k}\in\mathcal{D}_{ij}} \mathbb{P}\left[d_{ij,k}|\widetilde{P_{ij}}\right] \\ &= \prod_{\substack{k=1\\ d_{ij,k}=1}}^{w_{ij}} \mathbb{P}\left[d_{ij,k}|\widetilde{P_{ij}}\right] . \prod_{\substack{k=w_{ij}+1\\ d_{ij,k}=0}}^{m_{ij}} \mathbb{P}\left[d_{ij,k}|\widetilde{P_{ij}}\right] \\ &= \widetilde{P_{ij}}^{w_{ij}} \left(1-\widetilde{P_{ij}}\right)^{m_{ij}-w_{ij}}.\end{aligned} \tag{2.22}$$

ML estimation is maybe the most common form of estimation because it requires no information about prior distribution and it is often easy and straightforward to compute. Moreover, it has sometimes intuitive interpretation, which is the case in the framework of the game outcome prediction problem. The natural idea to approximate the probability of

winning by the ratio of matches won over matches played has solid mathematical foundations: it corresponds to the ML estimator $\overline{P_{ij}}^{\text{ML}} = \frac{w_{ij}}{m_{ij}}$. This is shown in Appendix D.2. Finally, as explained in Section 1.4, the ML estimator is the most efficient way to approximate $\widetilde{P_{ij}}$ in order to maximize the prediction accuracy $\mathcal{A}$.

Note that we could also want to maximize the probability of the total number of wins of $i$ over $j$ $\mathbb{P}\left[w_{ij}|\widetilde{P_{ij}}\right]$, instead of each game independently $\mathbb{P}\left[\mathcal{D}_{ij}|\widetilde{P_{ij}}\right]$. However, this would lead to the exact same formulation. This is shown in Appendix D.

### 2.5.3 Maximum *a posteriori* (MAP) estimator

The idea of the MAP estimator is to find the $\widetilde{P_{ij}}$ that could have best produced the matches $\mathcal{D}_{ij}$. Concretely, MAP estimation maximizes the posterior probability distribution of $\widetilde{P_{ij}}$, which is defined as the distribution of the parameter $\widetilde{P_{ij}}$ given the dataset $\mathcal{D}_{ij}$: $f_{\widetilde{P_{ij}}|\mathcal{D}_{ij}}(\widetilde{P_{ij}})$. The Bayes formula [Bay22] applies and allows to express the posterior distribution in terms of the likelihood of the observations $\mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}]$, the prior belief distribution $f_{\widetilde{P_{ij}}}(\widetilde{P_{ij}})$ on $\widetilde{P_{ij}}$ and a normalizing constant $\mathbb{P}[\mathcal{D}_{ij}]$.

$$\begin{aligned}
\overline{P_{ij}}^{\text{MAP}} &= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmax}} f_{\widetilde{P_{ij}}|\mathcal{D}_{ij}}(\widetilde{P_{ij}}) \\
&= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmax}} \frac{\mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}].f_{\widetilde{P_{ij}}}(\widetilde{P_{ij}})}{\mathbb{P}[\mathcal{D}_{ij}]} \\
&= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmax}} \mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}].f_{\widetilde{P_{ij}}}(\widetilde{P_{ij}}) \\
&= \underset{0\leqslant\widetilde{P_{ij}}\leqslant 1}{\operatorname{argmin}} -\ln\left(\mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}]\right) - \ln\left(f_{\widetilde{P_{ij}}}(\widetilde{P_{ij}})\right). \qquad (2.23)
\end{aligned}$$

We can observe that the MAP estimator depends directly on the choice of the prior distribution of $\widetilde{P_{ij}}$. In other words, the MAP estimator needs the assumption of some knowledge on $\widetilde{P_{ij}}$ prior to the discovery of $\mathcal{D}_{ij}$. Table 4.1 gives an overview of some prior distributions that could be used for $\widetilde{P_{ij}}$, such as the Beta distribution and the Beta-like distribution. A particular case happens when the prior is uniform, because the MAP estimator becomes equivalent to the ML estimator: $\overline{P_{ij}}^{\text{MAP}} = \overline{P_{ij}}^{\text{ML}} = \frac{w_{ij}}{m_{ij}}$. Indeed, the objective function of 2.23 reduces to the one of the ML estimation 2.21 as the prior term is constant, thus achieving the same optimal point. Conceptually, a uniform prior does not give any useful information about the variable, thus it is equivalent to assuming no prior knowledge.

A big default of an uniform prior distribution is that often $m_{ij}$ is small ($\leqslant 2$), which forces the estimator towards extreme probabilities zero or one. However, it seems impossible to have $\widetilde{P_{ij}} = 0$ or $\widetilde{P_{ij}} = 1$ in reality: each player has some non-zero chances to win any

game. It is possible to use prior distribution to tackle this issue. For example, choosing a Beta prior distribution $\text{Beta}(b,b)$ with $b > 1$ ($f_{\widetilde{P_{ij}}}(x) = \frac{x^{b-1}(1-x)^{b-1}}{\text{B}(b,b)}$, see Appendix E.2) solves the problem because $0 < \frac{b-1}{m_{ij}+2b-2} \leqslant \overline{P_{ij}}^{\text{MAP}} \leqslant \frac{m_{ij}+b-1}{m_{ij}+2b-2} < 1$. Prior distribution has a *normalization role*: it prevents $\overline{P_{ij}}$ to get unrealistic values by normalizing it towards $\frac{1}{2}$. Considering only this normalization role, it seems logical for the prior distribution to be identical for all $(i,j) \in \Omega$. Associated with the winner constraint, it implies that the prior is a *symmetric distribution* around $\frac{1}{2}$. However, it turns out that this idea is insufficient in order to improve results, this is described in Section 2.7.3 and in Appendix D.6.

For some prior choices, the MAP estimator could either be really hard or even impossible to express explicitly. In general, a small optimization subproblem needs to be solved iteratively in order to find $\overline{P_{ij}}^{\text{MAP}}$. To guarantee global optimality and fast convergence, a natural requirement is to ask the negative log-prior distribution to be convex. This condition will come back later in Section 3.1. This subproblem simply re-writes the problem 2.23 by including the expression of the likelihood function 2.22:

$$\overline{P_{ij}}^{\text{MAP}} = \underset{0 \leqslant \widetilde{P_{ij}} \leqslant 1}{\text{argmin}} -w_{ij} \ln\left(\widetilde{P_{ij}}\right) - (m_{ij} - w_{ij}) \ln\left(1 - \widetilde{P_{ij}}\right) - \ln\left(f_{\widetilde{P_{ij}}}(\widetilde{P_{ij}})\right). \tag{2.24}$$

### 2.5.4 Conditional mean (CM) estimator

Another commonly used estimator is the conditional mean (CM) estimator: $\overline{P_{ij}}^{\text{CM}} = \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right]$. Assuming a prior distribution $f_{\widetilde{P_{ij}}}(\widetilde{p_{ij}})$, we have the following integral expression for the CM estimator:

$$\begin{aligned}
\overline{P_{ij}}^{\text{CM}} &= \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] \\
&= \int_0^1 \widetilde{p_{ij}}.f_{\widetilde{P_{ij}}|\mathcal{D}_{ij}}(\widetilde{p_{ij}})\, d\widetilde{p_{ij}} \\
&= \int_0^1 \widetilde{p_{ij}}.\frac{\mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}].f_{\widetilde{P_{ij}}}(\widetilde{p_{ij}})}{\mathbb{P}[\mathcal{D}_{ij}]} d\widetilde{p_{ij}} \\
&= \int_0^1 \widetilde{p_{ij}}.\frac{\widetilde{p_{ij}}^{w_{ij}}(1-\widetilde{p_{ij}})^{m_{ij}-w_{ij}}.f_{\widetilde{P_{ij}}}(\widetilde{p_{ij}})}{\mathbb{P}[\mathcal{D}_{ij}]} d\widetilde{p_{ij}} \\
&= \int_0^1 \frac{\widetilde{p_{ij}}^{w_{ij}+1}(1-\widetilde{p_{ij}})^{m_{ij}-w_{ij}}.f_{\widetilde{P_{ij}}}(\widetilde{p_{ij}})}{\int_0^1 \widetilde{p_{ij}}^{w_{ij}}(1-\widetilde{p_{ij}})^{m_{ij}-w_{ij}}.f_{\widetilde{P_{ij}}}(\widetilde{p_{ij}})d\widetilde{p_{ij}}} d\widetilde{p_{ij}}.
\end{aligned} \tag{2.25}$$

The CM estimator has also a normalizing effect on $\widetilde{P_{ij}}$, regardless of the chosen prior distribution. For example, considering a uniform prior distribution, we find that $\overline{P_{ij}}^{\text{CM}} = \frac{w_{ij}+1}{m_{ij}+2}$, see Appendix D.3. This estimator avoids to predict extreme results 0 and 1 and pushes the estimate towards $\frac{1}{2}$. There is even an easy bound in the uniform prior case: $0 < \frac{1}{m_{ij}+2} \leqslant \overline{P_{ij}}^{\text{CM}} \leqslant 1 - \frac{1}{m_{ij}+2} < 1$.

It has the very desirable property to minimize the MSE, and is therefore called Minimum MSE (MMSE) estimator: $\overline{P_{ij}}^{\text{MMSE}} = \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] = \overline{P_{ij}}^{\text{CM}}$. The proof is in Appendix D.4.

### 2.5.5 Summary table

| Prior $f_{\widetilde{P_{ij}}}(x)$ | Posterior $f_{\widetilde{P_{ij}}\|\mathcal{D}_{ij}}(x)$ | $\overline{P_{ij}}^{\text{ML}}$ | $\overline{P_{ij}}^{\text{MAP}}$ | $\overline{P_{ij}}^{\text{CM}}$ |
|---|---|---|---|---|
| $\text{Uni}(0,1)$ | $\text{Beta}(w_{ij}, m_{ij} - w_{ij})$ | $\dfrac{w_{ij}}{m_{ij}}$ | $\dfrac{w_{ij}}{m_{ij}}$ | $\dfrac{w_{ij}+1}{m_{ij}+2}$ |
| $\text{Beta}(b,b)$ | $\text{Beta}(w_{ij}+b, m_{ij}-w_{ij}+b)$ | / | $\dfrac{w_{ij}+b-1}{m_{ij}+2b-2}$ | $\dfrac{w_{ij}+b}{m_{ij}+2b}$ |
| $f_{\widetilde{P_{ij}}}(x)$ | $\dfrac{\mathbb{P}[\mathcal{D}_{ij}\|\widetilde{P_{ij}}].f_{\widetilde{P_{ij}}}(x)}{\mathbb{P}[\mathcal{D}_{ij}]}$ | / | 2.24 | 2.25 |

Table 2.1: Table of estimators in function of prior distributions

Table 2.1 summarizes different possible choices of estimators $\overline{P_{ij}}$, in function of the prior distribution chosen. For the Beta prior distribution, the proofs are found in Appendix D.5. Note that, for all estimators, it can be verified that the probability constraints are verified: $\overline{P_{ij}} = 1 - \overline{P_{ji}}$, and $0 \leqslant \overline{P_{ij}} \leqslant 1$ for all $(i,j) \in \Omega$.

## 2.6 Low-rank matrix completion formulations on $P$

Based on classical methods, let us introduce two new LRMC methods adapted to the winning probability guessing problem. These methods compute the probability matrix $P$ by completing the incomplete probability matrix $\overline{P}$ (and constructed with some chosen estimators), in order to approximate the true probability matrix $\widetilde{P}$.

`LASSO`. One formulation dedicated to the matrix completion of the probability matrix is a restriction of the LASSO formulation 2.6 by adding the constraints of $P \in \mathcal{P}$:

$$\begin{aligned} \min_P \ & \frac{1}{2}\|\mathrm{P}_\Omega(P) - \mathrm{P}_\Omega(\overline{P})\|_F^2 + \tau\|P\|_* \\ & P \geqslant 0_{n\times n} \\ & P \leqslant 1_{n\times n} \\ & P + P^\top = 1_{n\times n}. \end{aligned} \tag{2.26}$$

`WLASSO`. Formulation 2.26 could be improved by using the weighted Frobenius norm, which is an idea developed with problem 2.10. A natural weight matrix to use is the confrontation matrix $W$: $\|P\|_{F,W}^2 = \sum_{i=1}^n \sum_{j=1}^n w_{ij} P_{ij}^2$. This leads to a new weighted LASSO

(WLASSO) formulation:

$$\begin{aligned} \min_P \frac{1}{2}\|\mathrm{P}_\Omega(P) - \mathrm{P}_\Omega(\overline{P})\|_{F,W}^2 + \tau\|P\|_* \\ P \geqslant 0_{n\times n} \\ P \leqslant 1_{n\times n} \\ P + P^\top = 1_{n\times n}. \end{aligned} \tag{2.27}$$

## 2.7 Results

Several completion methods have been tested in order find $P$ by matrix completion:

- `SDP_NNM`: solution to problem 2.3, implementation with [cvx] by [Fel].
- `SoftImpute`: solution to problem 2.5 with trade-off parameter $\tau$, implementation of the SVT algorithm by [Fel].
- `iterative_SVD`: solution to problem 2.9 with rank $r$ (for missing entries), implementation of the IterativeSVD algorithm by [Fel].
- `lmafit`: solution to problem 2.12 with rank $r$, implementation of the LMaFit algorithm by [DV].
- `LASSO`: solution to problem 2.26 with trade-off parameter $\tau$, implementation with [cvx] by myself.
- `WLASSO`: solution to to problem 2.27 with trade-off parameter $\tau$, implementation with [cvx] by myself.
- `kNN`: neighbourhood matrix completion technique (no low-rank assumption) with parameter $k$, implementation of $k$-Nearest Neighbours ($k$-NN) algorithm by [Fel].
- `columns_mean`: neighbourhood matrix completion technique, naive method, implementation of the column-averaging algorithm by myself.
- `rows_mean`: neighbourhood matrix completion technique, naive method, implementation of the row-averaging algorithm by myself.

The `LASSO` and `WLASSO` methods have probability constraints build-in. On the contrary, the other methods mentioned above are direct implementations of low-rank matrix completion techniques or neighbourhood matrix completion techniques, without adding the constraints of $P$ being a matrix of probabilities. This is problematic as the prediction accuracy $\mathcal{A}$ can

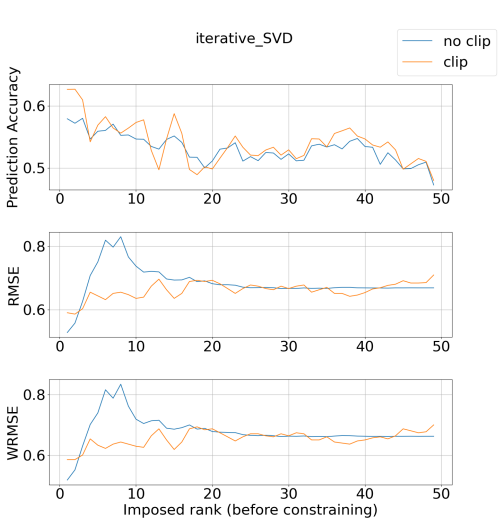
only be computed from a matrix $P$ respecting the constraints of a probability matrix. A small procedure to impose the constrains afterwards consists of the following steps:

1. Project all $P_{ij} < 0$ to 0 and all $P_{ij} > 1$ to 1 such that $0 \leqslant P_{ij} \leqslant 1$ for all $i$ and $j$.
2. Scale values in order to have $P_{ij} + P_{ji} = 1$ with the formula $P_{ij} \leftarrow \frac{P_{ij}}{P_{ij}+P_{ji}}$.
3. Impose $P_{ii} = \frac{1}{2}$.

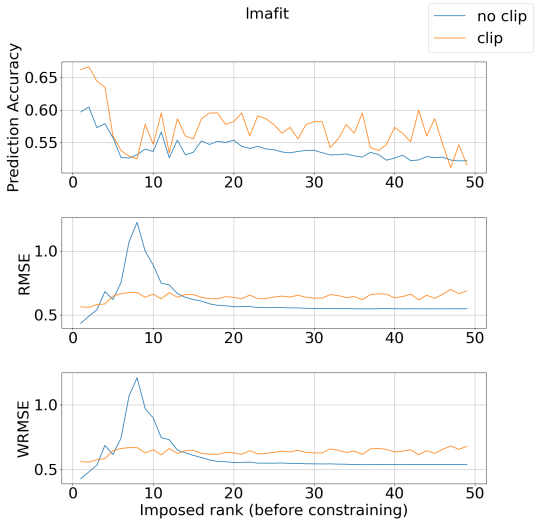
Throughout this section, results are presented for the LRMC techniques introduced in this chapter. First, optimal methods' parameters are found. Then, the singular values distribution of $P$ coming from the different methods is studied. After, the influence of the choices of estimator and prior distribution are discussed. Finally, the variability of the testing dataset due to its random selection is demonstrated. Note that comparisons against other methods are done in the last chapter (Section 4.8.5).

### 2.7.1 Optimal parameters

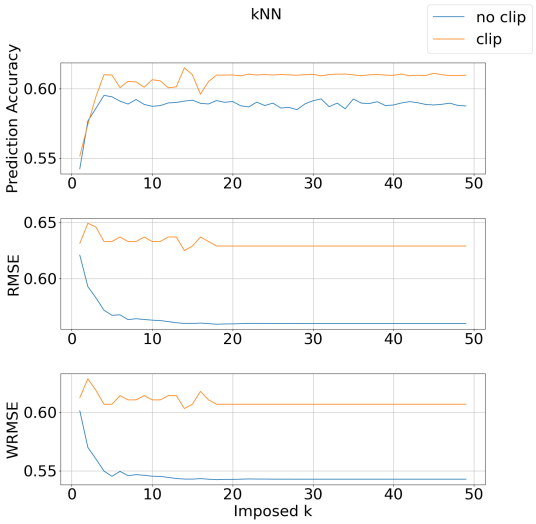
Let us first compute the optimal parameters for each method according to three criteria: prediction accuracy, RMSE and WRMSE.



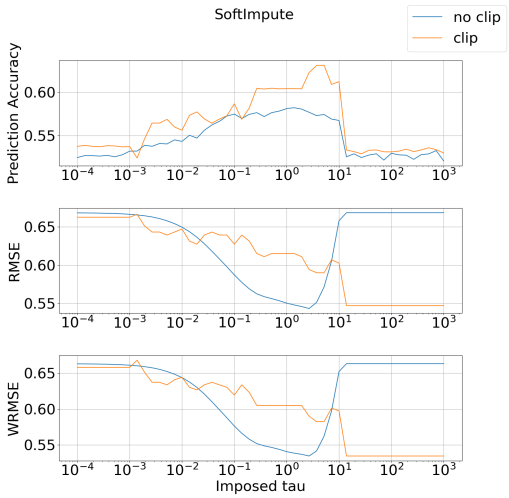

(a) `iterative_SVD`: $r_{\text{opt}} = 2$



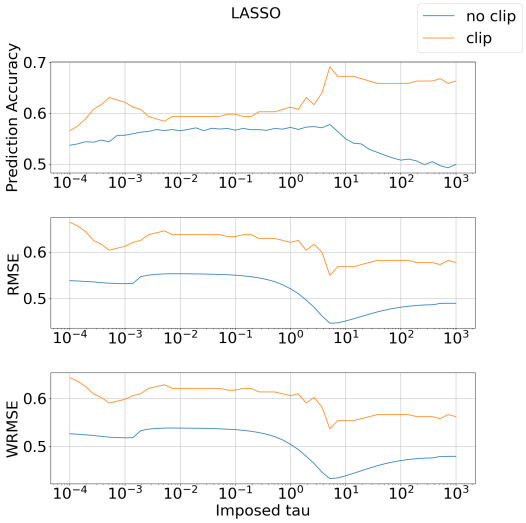

(b) `lmafit`: $r_{\text{opt}} = 2$



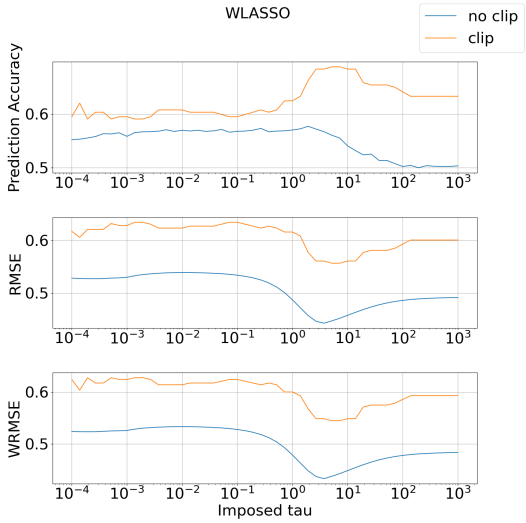

(c) `kNN`: $k_{\text{opt}} = 10$




(d) `SoftImpute`: $\tau_{\text{opt}} = 5.0$




(e) `LASSO`: $\tau_{\text{opt}} = 5.0$




(f) `WLASSO`: $\tau_{\text{opt}} = 5.0$

Figure 2.1: Finding best $r$ for `iterative_SVD` and `lmafit`, $k$ for `kNN`, $\tau$ for `SoftImpute`, `LASSO` and `WLASSO`. $n = 50$ players, season 2013, ML estimator, $\alpha_{\text{te}} = 0.3$

.

Several observations can be made from the Figure 2.1:

- Ranges of best values for the parameters according to the different metrics overlap nicely. It indicates that improving results on one metric also improves on the other ones.

- RMSE and WRMSE graphs are almost always parallel. This holds true for $P$ and $\mathcal{C}(P)$, and for all methods. This similarity is probably due to the fundamental resemblance of their definitions.

- WRMSE $<$ RMSE for all methods. This is indeed expected. From the way the training and testing sets are created, we expect $w_{ij,\text{te}} \approx \alpha_{\text{te}} w_{ij}$ and $w_{ij,\text{tr}} \approx (1-\alpha_{\text{te}}) w_{ij}$, so $w_{ij,\text{te}} \approx \frac{\alpha_{\text{te}}}{1-\alpha_{\text{te}}} w_{ij,\text{tr}}$. Therefore, the bigger $w_{ij,\text{te}}$, the bigger $w_{ij,\text{tr}}$. In other words, the more important an element is in the WRMSE ($(i,j) \in \Omega_{\text{te}}$), the more likely it is taken into consideration in the completion process ($(i,j) \in \Omega_{\text{tr}}$). If an element has not been trained on ($w_{ij,\text{tr}} = 0$), its reconstruction is probably less accurate. But if this element is nevertheless in $\Omega_{\text{te}}$, it is likely that $w_{ij,\text{te}}$ is really small (especially when $\alpha_{\text{te}}$ is small), and have therefore a small impact on WRMSE, while it would have a relatively higher impact on RMSE.

- RMSE($P$) $<$ RMSE($\mathcal{C}(P)$) and WRMSE($P$) $<$ WRMSE($\mathcal{C}(P)$) for most of methods. This seems logical since $P$ should be the matrix such that $P_{ij} \approx \widetilde{P_{ij}}$. As the clipping operator sends every entry to zero or to one, it should only increase the distance with $\widetilde{P_{ij}}$.

- $\mathcal{A}(\mathcal{C}(P)) > \mathcal{A}(P)$, which is expected from the analysis in Section 1.4 and was moreover the motivation of the definition of the clipping operator.

- For trade-off methods (`SoftImpute`, `LASSO`, `WLASSO`), there exists a sweet spot for the regularization parameter $\tau$, implying that the regularization approach works for this problem.

- A low-rank structure seems to appear for this problem. By looking at methods using a maximum rank $r$ (`iterative_SVD`, `lmafit`), it is clear that the best results are reached when $r$ is small, even close to one. Let us choose $r = 2$ as optimal rank. This is a nice sign that the low-rank approach for this matches prediction problem makes sense and could be fruitful.

- The imposed rank methods (`iterative_SVD`, `lmafit`) seem to perform badly between $r = 5$ and $r = 12$ concerning RMSE and WRMSE. This could be because the probability constraints are not imposed during the optimization process, but this has to be confirmed.

- `kNN` method quickly reaches a plateau in all three metrics, around $k \geqslant 10$.

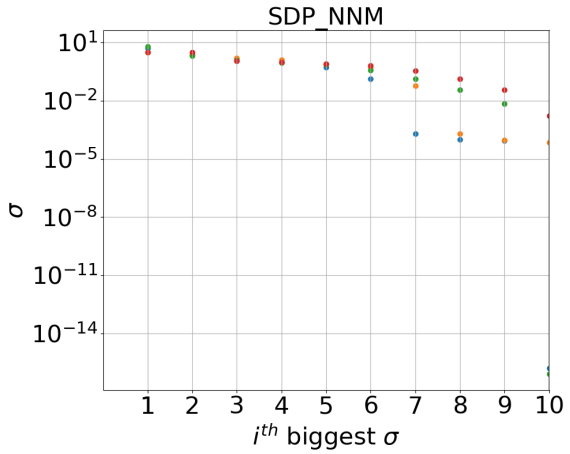
- `LASSO` and `WLASSO` seem to give the best results, proving that these new methods are motivated. This improvements could be due to the constraints integrated by definition

into the problems, while the other methods have their output matrix modified in order to satisfy the constraints and be able to compute the prediction accuracy metric.

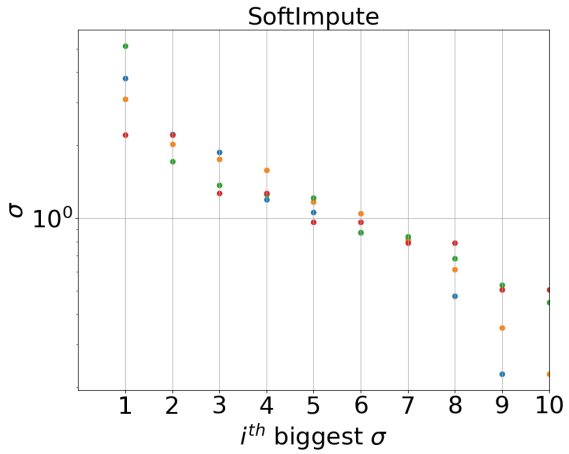
### 2.7.2 Singular values distribution

Let us have a look at the distributions of the singular values of $P$ for different methods in order to better understand the possibly low-rank structure of the problem. Some interesting ideas are developed in Section 1.6.2.

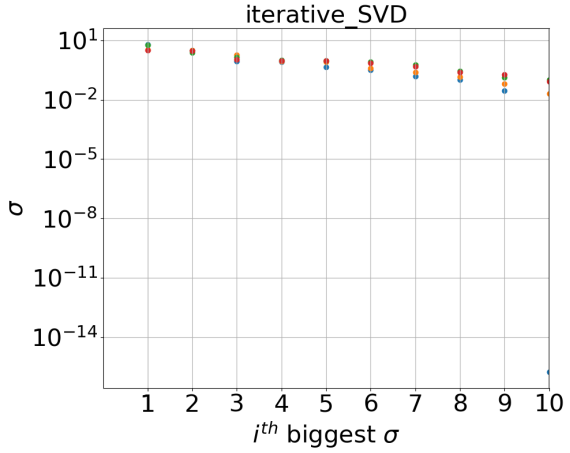
On Figure 2.2, the distribution of singular values of $P$ computed via different methods are represented. Singular values of $P' = P - \frac{1}{2}1_{n\times n}$ and of their clipped versions $\mathcal{C}(P)$ and $\mathcal{C}(P')$ are shown as well.



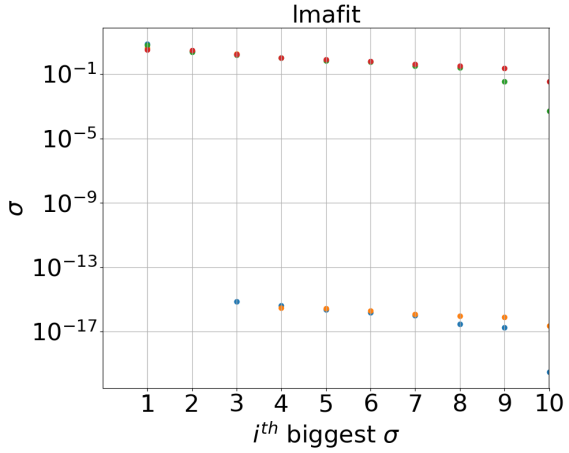

(a) `SDP_NNM`



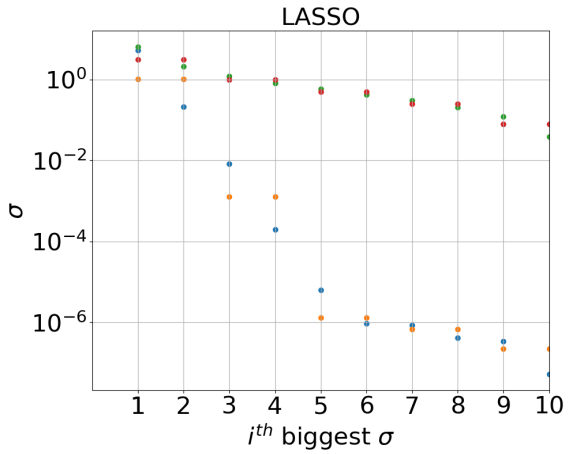

(b) `SoftImpute`



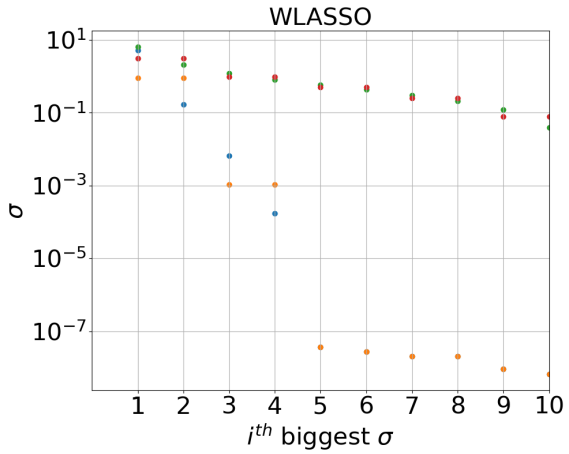

(c) `iterative_SVD`



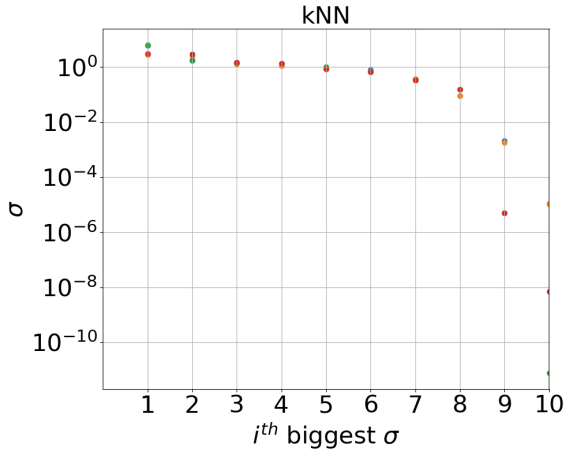

(d) `lmafit`



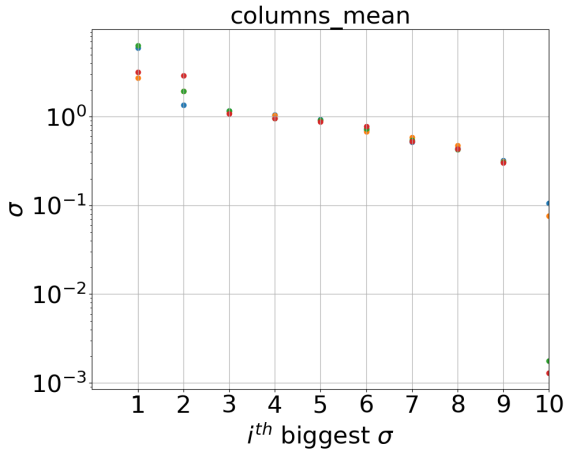

(e) `LASSO`



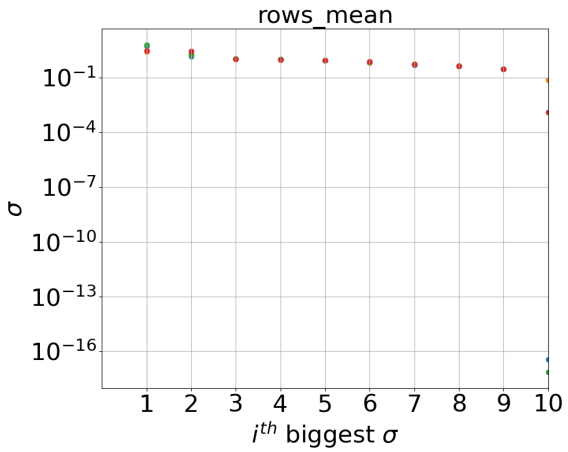

(f) `WLASSO`




(g) `kNN`




(h) `columns_mean`




(i) `rows_mean`

Figure 2.2: Singular values distributions of different methods before and after clipping, for $P$ and for $P' = P - \frac{1}{2}1_{n\times n}$ (centered): blue - original, orange - centered, green - clipped, red - clipped and centered. $n = 10$ players, season 2013, ML estimator, $\alpha_{\text{te}} = 0.3$

.

Some predictions were made in 1.6.2 and are verified:

- $\sigma_1(P)$ is always big, slightly over $\frac{n}{2}$.
- The singular values of $P'$ come in pairs as $P'$ is skew-symmetric.
- If the rank is bounded (such as in `lmafit`), or minimized via the nuclear norm (such as in `SDP_NNM`, `LASSO` and `WLASSO`), the smallest singular values of the non-clipped matrices drop close to 0. This represents the low-rank structure imposed by the methods.
- As expected, applying the clipping operator destroys the low-rankness. Indeed, $\mathcal{C}(P)$ is always full rank, for all methods.

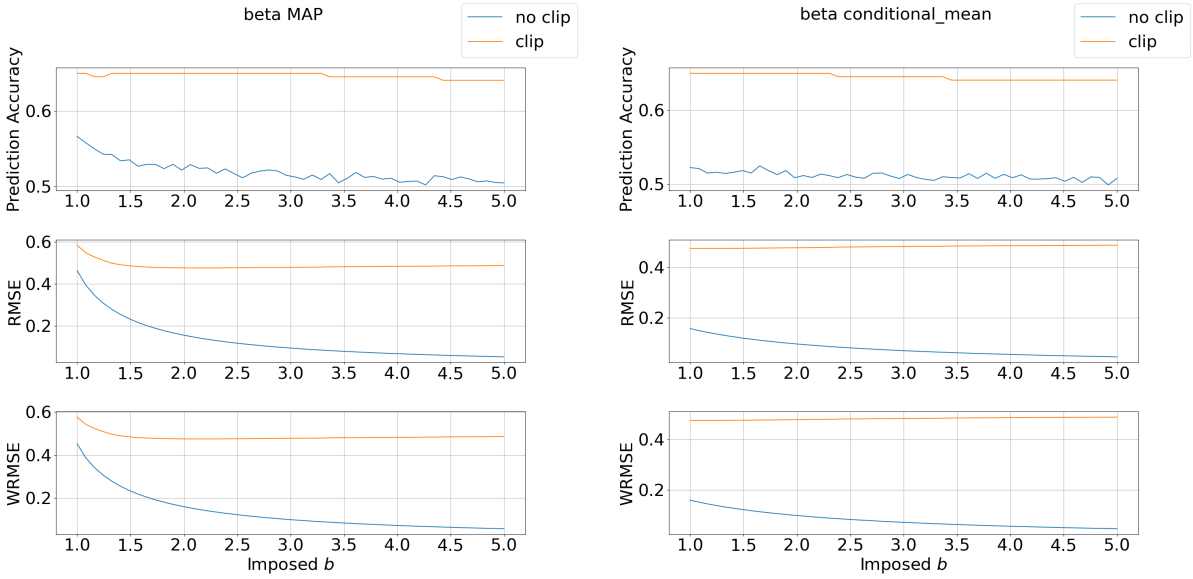
### 2.7.3 Estimators for $\widetilde{P_{ij}}$

As exposed in Section 2.5, in order to set up a matrix completion problem for guessing the outcome of tennis matches, the function $g_{ij}(\mathcal{D}_{ij})$ needs to be chosen in order to compute $\overline{P_{ij}}$ for $(i, j) \in \Omega$. The most logical way to do it is to use tools from statistics and in particular estimators.

Three types of estimators are studied in this thesis: ML (maximum likehihood), MAP (maximum *a posteriori*) and CM (conditional mean). The two last ones need to assume a prior distribution on $\widetilde{P_{ij}}$. If the prior is uniform, which is equivalent to considering no prior,

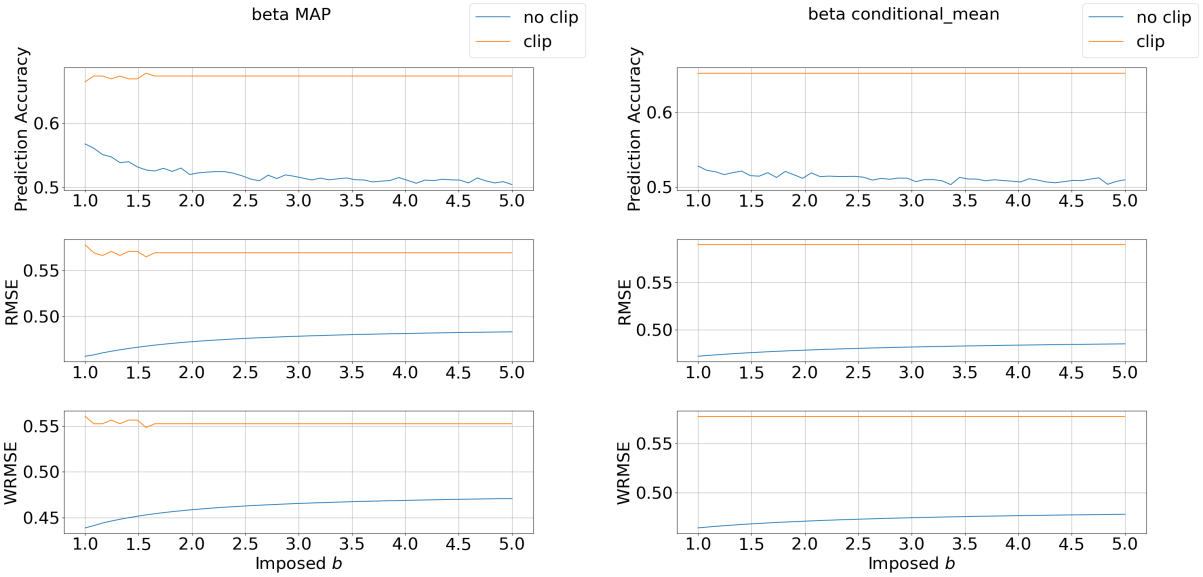
then the MAP estimator reduces to the ML estimator. In this section, the MAP and CM estimators are considered under the assumption of a Beta prior distribution: $\widetilde{P_{ij}} \sim \text{Beta}(b, b)$. The PDF is $f_{\widetilde{P_{ij}}}(x) = \frac{x^{b-1}(1-x)^{b-1}}{\text{B}(b,b)}$, with parameter $b \geqslant 1$ such that $-\ln(f_{\widetilde{P_{ij}}}(x))$ is convex. A special case is $b = 1$, because the beta distribution becomes the simple uniform distribution.

For the experiments showed in Figure 2.3, $\overline{P}_{\text{tr}}$ from the training set (for $(i, j) \in \Omega_{\text{tr}}$) is computed with MAP and CM estimators. For the comparisons with the testing sets by the RMSE and WRMSE metrics, $\overline{P}_{\text{te}}$ from the testing set (for $(i, j) \in \Omega_{\text{te}}$) is computed with the same estimator, but also with the ML estimator. Indeed, choosing twice the same MAP or CM estimator for training and testing sets leads irrevocably to a decrease to 0 for RMSE and WRMSE when choosing big values of $b$. Indeed, as shown in Appendix D.5, assuming a symmetric Beta prior distribution tends to shrink the range of possible values of $\overline{P_{ij}}_{\text{tr}}$ towards $\frac{1}{2}$, even more when $b$ increases. Therefore, the entries $P_{ij}$ found after optimization should also be closer to $\frac{1}{2}$. Choosing an estimator taking prior knowledge into account for the testing matrix would force $\overline{P_{ij}}_{\text{te}}$ towards $\frac{1}{2}$ as well, therefore decreasing the RMSE and WRMSE metrics. In order to avoid this phenomenon, the solution is to also choose an estimator independent of any prior distribution for the testing part, with a range of possible values from 0 to 1: the ML estimator.




(a) MAP for training and testing

(b) CM for training and testing




(c) MAP for training, ML for testing

(d) CM for training, ML for testing

Figure 2.3: Influence of parameter $b$ of the Beta$(b, b)$ prior distribution, for MAP (ML when $b = 1$) and CM estimators. $n = 50$ players, season 2013, `LASSO` method with $\tau = 5.0$, $\alpha_{\text{te}} = 0.3$

.

As expected, RMSE and WRMSE decrease to 0 (in the non-clipped case) for identical training and testing estimators when $b$ increases (see Figures 2.3a and 2.3b). On the contrary, they increase when compared with the ML estimator for testing (still in the non-clipped case, see Figures 2.3c and 2.3d). This is explained in the previous analysis as well. Besides, the ML estimator is known for often going to extreme probabilities (zero or one) because of the small size of the dataset compared to the number of unknowns. Using the ML estimator for testing (whose dataset is even smaller) means that a lot of testing values are zeros or ones, while on the contrary the training probabilities are pushed towards $\frac{1}{2}$. We could expect that the errors (RMSE, WRMSE) grow to 0.5 when $b$ increases, and this is indeed what happens.

Another important observation is that, perhaps surprisingly, adding prior knowledge never improves the prediction accuracy. This phenomenon is due to the symmetry of the prior distribution around $\frac{1}{2}$ mentioned in Section 2.5.3. Let us explain this assertion.

When the prior distribution is assumed symmetric and negative log-convex, then it is rather noteworthy that the clipped MAP estimator is equal to the clipped ML estimator: $\mathcal{C}\left(\overline{P_{ij}}^{\text{MAP}}\right) = \mathcal{C}\left(\overline{P_{ij}}^{\text{ML}}\right)$. This is shown in Appendix D.6. We could expect that the same kind of equation holds true for the reconstructed entries: $\mathcal{C}\left(P_{ij}{}^{\text{MAP}}\right) = \mathcal{C}\left(P_{ij}{}^{\text{ML}}\right)$. Therefore, no matter the prior assumed, as long as it is symmetric, the clipped version of the solutions coming from the ML and MAP frameworks should be identical. This is indeed what is

observed on Figures 2.3a and 2.3c.

What is more, this phenomenon has a physical interpretation. As explained in Section 1.4, in order to maximize the prediction accuracy, the winner should always be the player considered the strongest, i.e. the one with the highest winning probability: probabilities should be projected to zero or to one. This was by the way the reason behind the introduction of the clipping operator. Yet, by only using symmetric prior distribution to normalization purpose (this was motivated in Section 2.5.3), the probabilities during the training process are pushed towards $\frac{1}{2}$, but they always stay on the same side of this threshold. By clipping them afterwards for testing, the effect of the normalization just disappears, and thus the prior distribution has no impact at all on the prediction accuracy. Physically, the strongest player always stays the strongest player, even if his margin decreases. So, from the beginning, the idea of normalization was useless in terms of prediction accuracy. However, it can still be correctly motivated in terms of other metrics such as RMSE or WRMSE.

The same process is assumed to happen for the CM estimator: $\mathcal{C}\left(\overline{P_{ij}}^{\mathrm{CM}}\right) = \mathcal{C}\left(\overline{P_{ij}}^{\mathrm{ML}}\right)$ from where we can expect $\mathcal{C}\left(P_{ij}{}^{\mathrm{CM}}\right) = \mathcal{C}\left(P_{ij}{}^{\mathrm{ML}}\right)$. This relation is proved in the case of a Beta distribution, but it still lacks a proof in the general case (see Appendix D.7).

### 2.7.4 Variability of the testing set

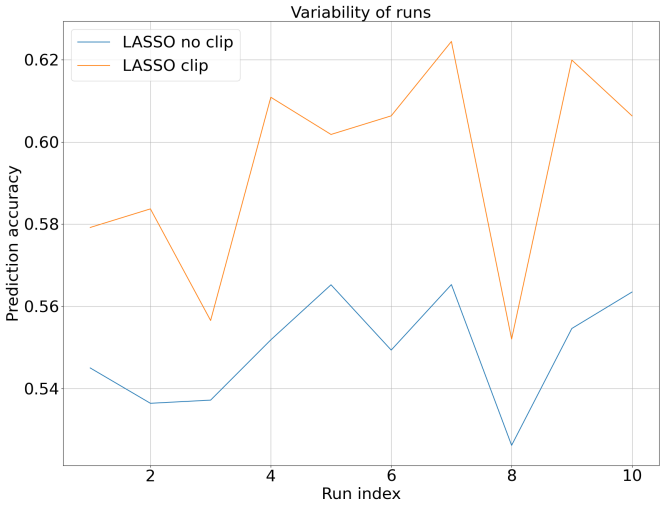



Figure 2.4: Different runs of `LASSO` method with $\tau = 0.5$. $n = 50$ players, season 2013, ML estimator, $\alpha_{\mathrm{te}} = 0.3$

.

For a given dataset $\mathcal{D}$, the testing set is composed of $\alpha_{\text{te}}.|\mathcal{D}|$ randomly selected matches within $\mathcal{D}$. The training set gathers all matches not selected in the testing set. A problematic feature comes from this randomness: some testing sets are more favourable than others for prediction. Indeed, the dataset over one year typically counts around 3000 matches, separated into training and testing sets of approximately 2100 and 900 matches ($\alpha_{\text{te}} = 0.3$). Meanwhile, for $n = 50$ players, the matrix $P$ has $\frac{n^2-n}{2} = 1225$ degrees of freedom. Even if the low-rank assumption aims to reduce this number, the training set stays relatively small compared with the number of variables. This leads to the overfitting phenomenon, which tells that the model is highly susceptible to variance. In this case, the variance is introduced by the randomness in the testing selection. Therefore, the prediction accuracy can strongly vary from run to run. Figure 2.4 shows exactly that phenomenon: the prediction accuracy can vary up to 6% for the same method and the same dataset $\mathcal{D}$. A solution to this flaw would be to introduce a new more reliable way of selecting the testing set, probably without using randomness.

# Chapter 3

# Low-Rank MAP formulations on the probability matrix

## 3.1 General MAP formulation

A different approach to matrix completion can be tried to tackle the match prediction problem. We call it the maximum *a posteriori* (MAP) approach. It sets up an optimization problem which has the following goal: finding the matrix $P$ that could have best produced the dataset $\mathcal{D}$, i.e. maximizing the posterior distribution of $P$ knowing $\mathcal{D}$, $f_{P|\mathcal{D}}(P)$. This kind of approach is often used in statistics and machine learning. In this MAP framework, the optimal matrix is found directly, in contrast to the matrix completion approach, where the matrix to complete was computed with statistical tools. Obviously, the mathematical developments of this whole section are really similar to the ones in Section 2.5.3, with the difference of working on $P$ and $\mathcal{D}$ instead of $\widetilde{P_{ij}}$ and $\mathcal{D}_{ij}$.

$$\begin{aligned} P^* &= \underset{P\in\mathcal{P}}{\operatorname{argmax}}\, f_{P|\mathcal{D}}(P) \\ &= \underset{P\in\mathcal{P}}{\operatorname{argmax}}\, \frac{\mathbb{P}[\mathcal{D}|P].f_P(P)}{\mathbb{P}[\mathcal{D}]} \\ &= \underset{P\in\mathcal{P}}{\operatorname{argmax}}\, \mathbb{P}[\mathcal{D}|P].f_P(P) \\ &= \underset{P\in\mathcal{P}}{\operatorname{argmin}} - \ln\left(\mathbb{P}[\mathcal{D}|P]\right) - \ln\left(f_P(P)\right). \end{aligned} \tag{3.1}$$

We recognize both terms in the last expression. The first one, $\mathbb{P}[\mathcal{D}|P] = \mathcal{L}(P|\mathcal{D})$ is known as the likelihood of the data $\mathcal{D}$ assuming the parameters $P$. Let us compute its expression with several remarks. First, using the assumptions of Section 1.1, we consider the matches to be independent: $\mathbb{P}[\mathcal{D}|P] = \prod_{\substack{(i,j)\in\Omega \\ i>j}} \mathbb{P}[\mathcal{D}_{ij}|P]$. The set $\mathcal{D}$ can be decomposed pair by pair, with each pair $(i,j)$ having played $m_{ij}$ matches. In order to not count any match

twice, we will only consider $i > j$. Then, also from assumptions of Section 1.1, matches between players $i$ and $j$ only depend on the probability $P_{ij}$: $\mathbb{P}[\mathcal{D}_{ij}|P] = \mathbb{P}[\mathcal{D}_{ij}|P_{ij}]$. Finally, $\mathbb{P}[\mathcal{D}_{ij}|\widetilde{P_{ij}}]$ is recognized as the likelihood $\mathcal{L}(P_{ij}|\mathcal{D}_{ij})$, and has been computed in equation 2.22. This gives:

$$\begin{aligned}
\mathcal{L}(P|\mathcal{D}) &= \mathbb{P}[\mathcal{D}|P] \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} \mathbb{P}[\mathcal{D}_{ij}|P] \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} \mathbb{P}[\mathcal{D}_{ij}|P_{ij}] \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} {P_{ij}}^{w_{ij}} \left(1 - P_{ij}\right)^{m_{ij}-w_{ij}} \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} {P_{ij}}^{w_{ij}} \prod_{\substack{(i,j)\in\Omega \\ i>j}} \left(1 - P_{ij}\right)^{m_{ij}-w_{ij}} \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} P_{ij}^{w_{ij}} \prod_{\substack{(i,j)\in\Omega \\ i>j}} {P_{ji}}^{w_{ji}} \\
&= \prod_{\substack{(i,j)\in\Omega \\ i>j}} {P_{ij}}^{w_{ij}} \prod_{\substack{(i,j)\in\Omega \\ j>i}} {P_{ij}}^{w_{ij}} \\
&= \prod_{(i,j)\in\Omega} {P_{ij}}^{w_{ij}}. \qquad (3.2)
\end{aligned}$$

Let us replace this development into problem 3.1:

$$\begin{aligned}
P^* &= \underset{P\in\mathcal{P}}{\operatorname{argmin}} - \ln\left(\prod_{(i,j)\in\Omega} P_{ij}^{w_{ij}}\right) - \ln\left(f_P(P)\right) \\
&= \underset{P\in\mathcal{P}}{\operatorname{argmin}} - \sum_{(i,j)\in\Omega} w_{ij} \ln\left(P_{ij}\right) - \ln\left(f_P(P)\right).
\end{aligned}$$

Then, the second term includes $f_P(P)$ the *prior* distribution of $P$. Obviously, the optimal matrix $P^*$ depends on the choice of its prior distribution, which is discussed in Section 3.2. In the end, the MAP estimation problem consists in minimizing the negative log-likelihood minus the log-prior. Notice that the problem is convex if and only if $-\ln\left(f_P(P)\right)$ is convex, as $-\ln(\mathcal{L})$ is always convex (see Appendix C.2).

$$\begin{aligned}
\min_P \; & - \sum_{(i,j)\in\Omega} w_{ij} \ln\left(P_{ij}\right) - \ln\left(f_P(P)\right) \qquad (3.3) \\
& P \geqslant 0_{n\times n} \\
& P \leqslant 1_{n\times n} \\
& P + P^\top = 1_{n\times n}.
\end{aligned}$$

## 3.2 Prior distribution of $P$

The MAP formulations need to account for the prior distribution of the matrix $P$: $f_P(P)$. Two main choices are possible and are detailed in this section. Firstly, one could assume a uniform prior distribution, which leads to an ML formulation. Secondly, a non-uniform prior can be considered. Each probability $P_{ij}$ requires its own prior distribution to be defined. Two important simplifying assumptions are then made ($P_{ij}$ mostly independent and identically distributed), leading to symmetric prior distributions.

### 3.2.1 Uniform prior (ML)

Assuming a uniform prior distribution on $P$ (equivalently assuming a uniform prior on each $P_{ij}$) is equivalent to consider no prior knowledge on $P$ (or on the elements $P_{ij}$). The MAP formulation then reduces to an ML formulation, similarly to what is described in Section 2.5.3:

$$\begin{aligned} \min_P \; & -\sum_{(i,j)\in\Omega} w_{ij} \ln\left(P_{ij}\right) \\ & P \geqslant 0_{n\times n} \\ & P \leqslant 1_{n\times n} \\ & P + P^\top = 1_{n\times n}. \end{aligned} \tag{3.4}$$

If no extra assumption is made, the problem is underdetermined and unusable in practice. Indeed, the elements $P_{ij} \in \Omega^c$ do not appear in the objective function and the constraints are not strong enough in order to fix them.

### 3.2.2 Non-uniform symmetric prior

In order to compute the expression of a non-uniform prior distribution, we need two significant assumptions.

The first main hypothesis about prior knowledge distributions in this work is to consider that all $P_{ij}$ are *a priori* independent. However, exceptions occur between pairs of players $P_{ij}$ and $P_{ji}$, as they are linked by the winner constraint $P_{ij} + P_{ji} = 1$, which can be rewritten as $f_{P_{ji}}(x) = f_{P_{ij}}(1-x)$. This implies:

$$f_P(P) = \prod_{(i,j)\in(\mathcal{N}\times\mathcal{N})} f_{P_{ij}}(p_{ij}) = \prod_{\substack{(i,j)\in(\mathcal{N}\times\mathcal{N}) \\ i>j}} f_{P_{ij}}(p_{ij}) f_{P_{ij}}(1-p_{ij}) \prod_{i\in\mathcal{N}} f_{P_{ii}}(p_{ii}).$$

The second assumption is even stronger: all $P_{ij}$ share the same distribution $d(x)$, $\forall (i,j) \in (\mathcal{N} \times \mathcal{N})$. This assumption extends for all pairs: $f_{P_{ij}}(x) = f_{P_{ji}}(x) = d(x)$. It implies directly that the common distribution $d(x)$ is symmetric around $\frac{1}{2}$: $f_{P_{ij}}(x) = f_{P_{ji}}(x) = f_{P_{ij}}(1-x) = d(x)$. We get:

$$f_P(P) = \prod_{(i,j)\in(\mathcal{N}\times\mathcal{N})} d(p_{ij}) = \prod_{(i,j)\in(\mathcal{N}\times\mathcal{N})} f_{P_{ij}}(p_{ij}).$$

This second assumption can be motivated as follows. On the one hand, since all entries represent the same kind of value (probabilities of a player $i$ winning against another player $j$), they should therefore be treated equally. On the other hand, it traduces symmetry, which is, in general, a good sign for a real problem.

Using these two main assumptions, the MAP problem 3.3 then becomes:

$$\begin{aligned} \min_P & - \sum_{(i,j)\in\Omega} w_{ij} \ln\left(P_{ij}\right) - \sum_{(i,j)\in(\mathcal{N}\times\mathcal{N})} \ln\left(f_{P_{ij}}(P_{ij})\right) \\ & P \geqslant 0_{n\times n} \\ & P \leqslant 1_{n\times n} \\ & P + P^\top = 1_{n\times n}. \end{aligned} \tag{3.5}$$

Several choices can be made for the prior distribution of $P_{ij}$, as long as $-\ln\left(f_{P_{ij}}(P_{ij})\right)$ is convex such that problem 3.5 stays convex, and symmetric around $\frac{1}{2}$. Different suitable prior distributions are described in Section 4.5.

This formulation 3.5 concerns all elements $P_{ij}$, even the ones with $(i,j) \in \Omega^c$. However, without additional constraints, the latter are all set to $\frac{1}{2}$, as the prior distribution is symmetric and negative log-convex (see Appendix C.5). Besides, for the $P_{ij}$ with $(i,j) \in \Omega$, the clipped value (optimal for maximizing prediction accuracy), is the same as the one in the ML problem. Indeed, the symmetric prior has only a normalization role (see Section 2.5.3), and this effect vanishes after clipping. Therefore, changing the prior does not affect the solution. This phenomenon has been fully explained in Section 2.7.3.

## 3.3 Low-rank MAP formulations on $P$

The MAP formulation 3.5 is convex as $-\ln\left(f_{P_{ij}}(P_{ij})\right)$ is chosen convex (see Table 4.1), which means that all tools and guarantees of convex optimization techniques can be applied. However, it has one big flaw: it does not include in any way the low-rankness of matrix $P$. In order to account for the low-rank feature while preserving a convex problem, two solutions already explored in the matrix completion framework are possible.

### 3.3.1 MAP with NNM

`MAP_NNM`. The first solution consists in adding a regularization term involving the nuclear norm, the convex envelope of the rank, with a trade-off parameter $\tau$:

$$\begin{aligned} \min_P & -\sum_{(i,j)\in\Omega} w_{ij} \ln\left(P_{ij}\right) - \sum_{(i,j)\in(\mathcal{N}\times\mathcal{N})} \ln\left(f_{P_{ij}}(P_{ij})\right) + \tau\|P\|_* \\ & P \geqslant 0_{n\times n} \\ & P \leqslant 1_{n\times n} \\ & P + P^\top = 1_{n\times n}. \end{aligned} \tag{3.6}$$

### 3.3.2 MAP with BMF

`MAP_BMF`. The second idea is to use the bilinear matrix factorization $P = UV^\top$ with given rank $r$, $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$. This means that $P_{ij} = \sum_{k=1}^r U_{ik}V_{jk}$:

$$\begin{aligned} \min_{U,V} & -\sum_{(i,j)\in\Omega} w_{ij} \ln\left(\sum_{k=1}^r U_{ik}V_{jk}\right) - \sum_{(i,j)\in(\mathcal{N}\times\mathcal{N})} \ln\left(f_{P_{ij}}\left(\sum_{k=1}^r U_{ik}V_{jk}\right)\right) \\ & UV^\top \geqslant 0_{n\times n} \\ & UV^\top \leqslant 1_{n\times n} \\ & UV^\top + VU^\top = 1_{n\times n}. \end{aligned} \tag{3.7}$$

This problem is not convex anymore, but it is still block-component-wise convex. In order to tackle this last formulation, a natural try is thus to use convex optimization by alternating minimization on $U$ and $V$. However, this is really hard or even sometimes impossible to do due to the last equality matrix constraint. During the following explanations concerning this fact, we suppose that $U$ is fixed and that $V$ is the optimization variable, but the reasoning holds in the opposite situation as well due to symmetry. Then, the constraint $UV^\top + VU^\top = 1_{n\times n}$ gets really problematic. Let us denote the number of independent equalities as $e(n)$. Noticing the symmetry, there are at most $\frac{n^2+n}{2}$ independent equalities: $e(n) \leqslant \frac{n^2+n}{2}$. Yet, each component has only $rn$ degrees of freedom and the low-rank framework particularly asks for small $r$. So, most of the time, there are too many equalities compared to the number of variables: $rn < e(n)$. For example, if $e(n) = \frac{n^2+n}{2}$, then this happens when $rn < \frac{n^2+n}{2} \iff r < \frac{n+1}{2}$. This implies that the feasible set $\mathcal{F}$ of $V$ (not even satisfying the other inequalities) can be empty, which makes this problem infeasible. More precisely, it depends on $U$ and thus on how many of the $\frac{n^2+n}{2}$ equations are independent.

For some $U$, this feasible set is not empty. An easy example is $U = \frac{1_{n\times r}}{\sqrt{2r}}$. However, this implies that $V = \frac{1_{n\times r}}{\sqrt{2r}}$ as well. The feasible set is a singleton, there is no room for

optimization. Moreover, $V$ is of rank 1, which is not interesting as the hope was that the optimization reached the upper bound for the rank $r$. More crucially, this imposes $P_{ij} = \frac{1}{2}$ for all $(i, j)$, which is a completely useless solution.

The only example of $U$ I could come up with in this thesis which allows some freedom to $V$ is the following: $U = (\frac{1_{n\times 1}}{\sqrt{2}}|0_{n\times r-1})$ and $V = (\frac{1_{n\times 1}}{\sqrt{2}}|V')$. We can easily verify that the equality constraints are verified, while the submatrix $V'$ of size $n \times r - 1$ is free. However, by definition of $U$, $V'$ disappears in the objective function and becomes irrelevant, while $P_{ij} = \frac{1}{2}$ for all $(i, j)$ once again.

To summarize about `MAP_BMF`, the low-rank problem 3.7 can seemingly not be tackled via alternating convex minimization because the probability constraints are too restrictive. Most values of $U$ (or $V$) make the problem infeasible. An especially critical step is therefore the initialisation of those matrices. It is an open question to determine the conditions on $U$ (or on $V$) such that the feasible set is non-empty, and even further such that this set is not a singleton. Note that other optimization techniques could potentially solve this problem 3.7.

## 3.4 Results

In this section, we present results for the `MAP_NNM` method introduced in this chapter. In particular, we look at the influence of the method's parameter and of the prior distribution choice. Comparisons against other methods are done in the last chapter (Section 4.8.5).

### 3.4.1 Optimal parameters

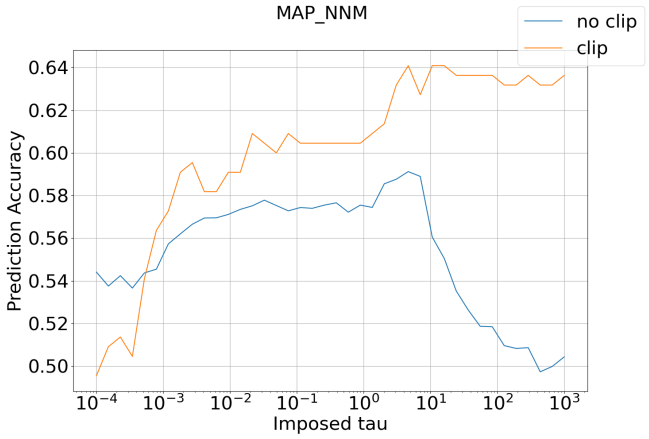



Figure 3.1: `MAP_NNM`: $\tau = 5.0$. Finding best $\tau$ for `MAP_NNM`. $n = 50$ players, season 2013, $f_P(P)$ uniform, $\alpha_{\text{te}} = 0.3$.

Figure 3.1 tells that the optimal parameter for the `MAP_NNM` method is $\tau = 5.0$ according to the prediction accuracy metric. Recall that RMSE and WRMSE criteria are not applicable in the MAP context. Surprisingly, the clipped version keeps a high prediction accuracy even when the non-clipped version dramatically decreases ($\tau > 10$).

### 3.4.2 Prior distribution

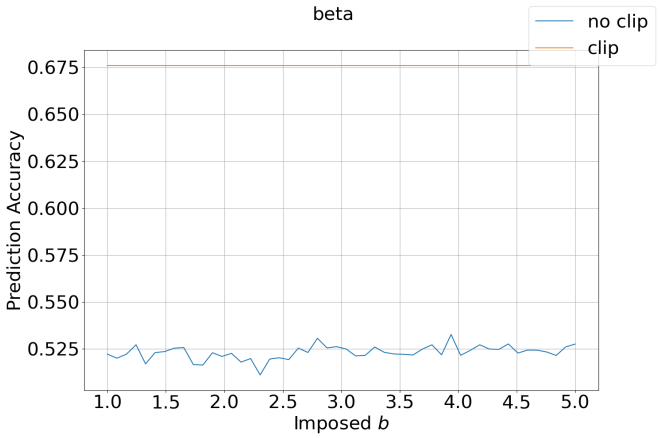



Figure 3.2: `MAP_NNM`. Influence of parameter $b$ of the Beta$(b, b)$ prior distribution. $n = 50$ players, season 2013, $\tau = 5.0$, $\alpha_{\text{te}} = 0.3$.

Figure 3.2 shows that modifying parameter $b$ of the Beta prior distribution assumed on $P_{ij}$ does not affect the prediction accuracy for the `MAP_NNM` method when the $P$ matrix is clipped. This result is extremely similar to the ones in Section 2.7.3 for matrix completion with MAP estimators. As detailed in 3.2.2, it is expected for a solution to problem 3.5 that the choice of the prior distribution (and its parameter) should not influence the clipped accuracy. However, `MAP_NNM` solves the slightly different problem 3.6, which includes an additional nuclear norm term. This term however does not depend on the prior. It can thus be guessed that the clipped accuracy should behave identically as it does without this additional term, i.e. stay constant. This is indeed what is observed.

# Chapter 4

# Low-rank MAP rating formulation

## 4.1 Motivation of rating formulation

The probability matrix formulations, introduced in Chapter 1 and detailed in Chapters 2 and 3, have several fundamental flaws, inevitably preventing them from achieving good results. Here follows a non-exhaustive list of shortcomings:

1. The matrix $P$ is heavily constrained as it belongs to $\mathcal{P}$: $0 \leqslant P_{ij} \leqslant 1$, $P_{ij} + P_{ji} = 1$ and $P_{ii} = \frac{1}{2}$.
2. The diagonal elements $P_{ii}$ have no real meaning, but they need to be taken into account during the optimization process as they are constraints and could have an impact on the assumed low-rank structure.
3. It is unclear why matrix $P$ should have some low-rank structure.
4. Except for simplicity purposes, it is hard to motivate any prior distribution on a probability $P_{ij}$.
5. Too little information is taken into account in order to reach prediction levels, as described in Section 1.1.

This Chapter 4 explores ideas in order to tackle all those issues. First, the widely used Bradley-Terry-Luce (BTL) model for outcome prediction will be detailed. It justifies how to link winning probabilities $P_{ij}$ with rating levels in simple ways. Two different ratings could be used: $S_i$, which is positive, and $E_i$, which is unconstrained. Then, this BTL model is extended by a new dimension such as the tournament in order to account for one more feature from the dataset. The variable becomes a matrix of ratings, where $S_i^c$ and $E_i^c$

represent the rating of player $i$ in tournament $c$. In their paper [XTFF19], Xia, Tan, Filstroff and Févotte tackled the $S$ ratings formulation. In addition to the low-rank structure, they exploit the nonnegativity of $S$ for interpretability purposes. In this work, the focus is set on the second unconstrained ratings $E$. We can easily motivate this rating matrix to be low-rank. Using a MAP framework on it allows us to finally write a new formulation of the prediction problem, called the *$E$ rating formulation*.

## 4.2 Bradley-Terry-Luce model

The Bradley-Terry-Luce (BTL) model is a widely used model in the context of confrontations between two players or two teams [BTL22]. It assigns the probability of winning a match only by knowing the rating difference $\Delta E_{ij}$ between players $i$ and $j$ thanks to a logistic relation, i.e. a sigmoid function [Sig22]. The parameter $\lambda$ is a scaling parameter, which can be fixed arbitrarily, often to $\lambda = 1$:

$$P_{ij} = \frac{1}{1 + e^{-\lambda \Delta E_{ij}}}.$$

The BTL formula can be inverted in order to find the rating difference from winning probabilities. This implies the inverse sigmoid function, also called logit function [Log22b]:

$$\Delta E_{ij} = \frac{1}{\lambda} \ln \left( \frac{P_{ij}}{1 - P_{ij}} \right).$$

In order to fully understand this model, its parameters and variables, we will reconstruct it step by step. The BTL model is derived from an intuitive transitive property on the odds of winning $R_{ij}$. The odds of player $i$ winning a match against player $j$ represent the probability that $i$ wins over the probability that $j$ wins. It is a way to represent relative strength between players:

$$R_{ij} = \frac{P_{ij}}{1 - P_{ij}}$$
$$P_{ij} = \frac{R_{ij}}{1 + R_{ij}}.$$

From the definition of probability, there is an intrinsic positivity constraint: $R_{ij} \geqslant 0$. Furthermore, there is an inverse relationship between $R_{ij}$ and $R_{ji}$:

$$R_{ij} = \frac{P_{ij}}{1 - P_{ij}} = \frac{1 - P_{ji}}{P_{ji}} = \frac{1}{R_{ji}}. \tag{4.1}$$

The idea of the BTL model is to assume that the odds are *multiplicative transitive*:

$$R_{ij} R_{jk} = R_{ik}.$$

To rephrase, if $R_{ij} = l$ ($i$ beats $j$ $l$ times more often than $j$ beats $i$), and $R_{jk} = m$ ($j$ beats $k$ $m$ times more often than $k$ beats $j$), then $R_{ik} = ml$ ($i$ beats $k$ $lm$ times more often than $k$ beats $i$). In terms of probability, this transitive property is not trivial and reads:

$$\frac{P_{ij}}{1-P_{ij}}\frac{P_{jk}}{1-P_{jk}} = \frac{P_{ik}}{1-P_{ik}}.$$

Before going further, let us remark that this transitivity hypothesis is questionable. Indeed, in the real world, a strong player could have trouble against a particular player which is supposed to be weaker, like his pet peeve. Similarly to the signs in rock-paper-scissors, there could be no stronger player within a triplet of players: they each beat one another. In other words, these players, like these signs, are not transitive. However, this assumption allows to define the easy yet effective BTL model, so let us continue.

We can define a vector $S$ for which $S_i \geqslant 0$ corresponds to a rating (or level, or points, or score) of player $i$. The odds $R_{ij}$ of $i$ beating $j$ are simply the ratio between their ratings:

$$R_{ij} = \frac{S_i}{S_j}.$$

This vector $S$ is defined up to a multiplicative constant, i.e. multiplying $S$ by any constant would not impact the odds $R_{ij}$. This extra degree of freedom implied by this definition of $S$ can be fixed by normalization of this vector. We could for example impose $S_1 = 1$, or $\|S\| = 1$. The choice in this thesis is to impose that the product of all components equals to 1 and is explained further: $\prod_{i=1}^{n} S_i = 1$. Moreover, this vector $S$ naturally relates to matrix $P$:

$$P_{ij} = \frac{1}{1+\frac{S_j}{S_i}} = \frac{S_i}{S_i+S_j}.$$

The properties of $S$ are desirable. First, the stronger the player is, the bigger his rating $S_i$ will be and the higher his winning probability will be. Then, the probability $P_{ij}$ is extremely easy to compute. $P_{ij}$ has a nice interpretation as well: the fraction of points that $i$ owns compared to the total points that are involved in the match. Finally, the transitive property is naturally satisfied.

However, there are two downsides to $S$. On the one hand, it is constrained to be positive. Even if it can be natural to define a score to be positive (for example, Netflix films are rated between 0 and 5 [Net22]), it is an extra restriction that needs to be taken into account during the optimization process, and which is preferably avoided. On the other hand, even though it intrinsically satisfies the multiplicative transitivity, we would prefer a rating score which is *additive transitive*. In other words, the probability should depend on the difference in rating instead of the quotient of ratings.

This leads to the definition of vector $E$ which fulfils those last requirements and can be seen as simply a parametrization of $S$, with a free to choose coefficient $\lambda$:

$$E_i = \frac{1}{\lambda}\ln(S_i)$$
$$S_i = e^{\lambda E_i}.$$

$E$ gives an absolute score for each player, corresponding to another rating for each player. The ratings can be either positive or negative, avoiding the positivity constraint. This vector is translation invariant as the winning probability only depends on the difference of ratings, so we impose that the average rating is zero in order to fix this degree of freedom: $\frac{1}{n}\sum_{i=1}^{n} E_i = \sum_{i=1}^{n} E_i = 0$. Note that this condition is equivalent to the normalization of $S$ described above $\prod_{i=1}^{n} S_i = 1$. Other choices are possible for fixing this translation degree of freedom, such as $E_1 = 0$. The vector $E$ is also scaling invariant: the parameter $\lambda$ encodes the standard deviation of $E$.

The matrix $\Delta E$ holds the value of the score differences between two players $i$ and $j$. The element $\Delta E_{ij} = E_i - E_j = \frac{1}{\lambda}\ln(R_{ij}) = \frac{1}{\lambda}\ln(\frac{S_i}{S_j})$ is the rating difference between two players. If $i$ beats $j$, then $E_i > E_j \iff \Delta E_{ij} > 0$, which is an elegant and intuitive condition. Besides, it satisfies naturally the additive transitivity constraint:

$$\Delta E_{ij} + \Delta E_{jk} = \Delta E_{ik}.$$

Finally, we have found the BTL model equation back. This model is rather easy to understand and grasp and is therefore applied in a variety of situations. For example, the chess world uses its ELO rating system built on top of the BTL model. The ELO system uses $\lambda = \frac{1}{400}$ and it can be normalized for a average rating of ${E_i}^c = 1500$, such as on Lichess [ELO]. The ELO system adds a dynamical aspect: it introduces an update formula for the ratings after each new result. It predicts winning probability based only on the current ratings of players, not accounting for any other external factors. This makes it compatible with the assumptions of Section 1.1, but at the same time it inherits all flaws associated with those assumptions.

## 4.3 Increasing by one dimension

In order to complexify the model to improve results, we can add a new dimension to our problem. This allows to take one more feature into the model. For example, we can differentiate the played matches by the surface, the tournament or any other complementary criterion. Considering that the extra dimension is the tournament, let us fix some notation:

- $\mathcal{T}$ is the set of all tournaments in the dataset $\mathcal{D}$.
- $t = |\mathcal{T}|$ is the number of tournaments.
- ${P_{ij}}^c$ is the probability of player $i$ beating player $j$ in tournament $c$. It is still constrained as usual: $0 \leqslant {P_{ij}}^c \leqslant 1$, ${P_{ij}}^c + {P_{ji}}^c = 1$ and ${P_{ii}}^c = \frac{1}{2}$.
- ${E_i}^c$ is a rating for player $i$ in tournament $c$.
- $\Delta {E_{ij}}^c = {E_i}^c - {E_j}^c$ is the rating difference between players $i$ and $j$ in tournament $c$.
- ${S_i}^c = e^{\lambda {E_i}^c}$ is the positive rating for player $i$ in tournament $c$ ($ {S_i}^c \geqslant 0$), and $\lambda > 0$ is a free parameter.
- ${R_{ij}}^c = \frac{{P_{ij}}^c}{1-{P_{ij}}^c} = \frac{{S_i}^c}{{S_j}^c}$ is the odds of player $i$ beating player $j$ in tournament $c$. It is positive: ${R_{ij}}^c \geqslant 0$.
- $\Omega$ is redefined in order to account for the extra dimension:

$$\Omega = \{(i, j, c) \in (\mathcal{N} \times \mathcal{N} \times \mathcal{T}) : {m_{ij}}^c > 0\}.$$

The constitutive relations of the BTL model still hold true:

$${P_{ij}}^c = \frac{1}{1 + e^{-\lambda \Delta {E_{ij}}^c}} = \frac{{S_i}^c}{{S_i}^c + {S_j}^c}. \tag{4.2}$$

## 4.4 Low-rank formulations on ratings

### 4.4.1 ML on $S$ ratings

In paper [XTFF19], Xia, Tan, Filstroff and Févotte use the positive ratings $S \in \mathbb{R}^{n \times t}$ (called $\Lambda^\top$ in the paper) and impose a low-rank nonnegative matrix factorization (NMF, as described in Section 2.4.4) $\Lambda = WH$ such that $W \in \mathbb{R}_+^{t \times r}$ and $H \in \mathbb{R}_+^{r \times n}$. The skill level of player $i$ on surface $j$ is then ${S_i}^c = \Lambda_{ci} = [WH]_{ci} = \sum_{k=1}^{r} W_{ck} H_{ki}$. This decomposition naturally imposes $S$ ratings to be positive and low-rank. Moreover, by adding some column normalization on $W$ ($\sum_{c=1}^{t} W_{ck} = 1, \forall 1 \leqslant k \leqslant r \iff 1_{t\times 1}^\top W = 1$) and global normalization on $H$ ($\sum_{k=1}^{r} \sum_{i=1}^{n} H_{ki} = 1 \iff \langle H, 1_{r \times n} \rangle = 1$), we gain the uniqueness of this decomposition and more interpretability. Matrix $W$ can be seen as the *dictionary matrix* where $W_{ck}$ gives the probability of tournament $c$ to be of type $k$. Matrix $H$ is the *coefficient matrix* and $H_{ki}$ gives the skill level of player $i$ on tournaments of type $k$.

The optimization problem associates an ML framework with the BTL model and the NMF. Let us recall the expression of the likelihood function from 3.2 and the expression of ${P_{ij}}^c = \frac{S_i^c}{S_i^c + S_j^c}$ from 4.2. It gives:

$$\begin{aligned} \min_{W,H} & -\sum_{(i,j,c)\in\Omega} w_{ij}^c \left( \ln\left( \sum_{k=1}^{r} W_{ck} H_{ki} \right) - \ln\left( \sum_{k=1}^{r} W_{ck} H_{ki} + \sum_{k=1}^{r} W_{ck} H_{kj} \right) \right) \\ & 1_{t\times 1}^\top W = 1 \\ & \langle H, 1_{r\times n} \rangle = 1 \\ & W \geqslant 0 \\ & H \geqslant 0. \end{aligned} \tag{4.3}$$

The formulation [XTFF19] has two major flaws that could be tackled. First, even if it has interpretability purposes, working with $S$ ratings instead of $E$ ratings adds positivity constraints, and constraining a problem can only deteriorate the optimal point. Then, the formulation does not use any prior distribution on ratings as it falls into the ML framework and not into the MAP framework.

### 4.4.2 MAP on $E$ ratings

`Eratings`. This leads to a new MAP ratings formulation, which includes the prior distribution of $E$. The assumptions from Section 3.2.2, i.e. that all ${P_{ij}}^c$ are assumed i.i.d. *a priori*, allow to express the prior $f_E(E)$ of $E$ in function of independent and identically distributed ${E_i}^c$: $f_E(E) = \prod_{(i,c)\in(\mathcal{N}\times\mathcal{T})} f_{{E_i}^c}({E_i}^c)$. Indeed, as all ${P_{ij}}^c$ are i.i.d., the inverse BTL formula tells that all rating differences $\Delta {E_{ij}}^c$ should be independent and identically distributed as well. If the ratings are themselves i.i.d., this always holds true. We then get the following MAP problem on the $E$ ratings:

$$\min_E \sum_{(i,j,c)\in\Omega} w_{ij}^c \ln\left(1 + e^{-\lambda({E_i}^c - {E_j}^c)}\right) - \sum_{(i,c)\in(\mathcal{N}\times\mathcal{T})} \ln\left(f_{{E_i}^c}({E_i}^c)\right). \tag{4.4}$$

Note that this problem is convex as the likelihood term is convex (see Appendix C.3) and as $-\ln\left(f_{{E_i}^c}({E_i}^c)\right)$ is chosen convex as well.

Section 4.6 motivates that $E$ should have a low-rank structure. The BMF method is used to impose a low-rank constraint on $E \in \mathbb{R}^{n\times t}_{\leqslant r}$ as we define $E = CT^\top$, with $C \in \mathbb{R}^{n\times r}$ and $T \in \mathbb{R}^{t\times r}$. We have ${E_i}^c = \sum_{k=1}^{r} C_{ik} T_{ck}$. Moreover, we choose $E$ to be centered at 0, and therefore $\sum_{i=1}^{n} {E_i}^c = 0$. An easier sufficient condition asks $1_{n\times 1}^\top C = 0$. Before defining the MAP problem on $E$, we will first define the objective function $O(C, T)$ for clarity and

conciseness. Note that this function is not convex due to BMF as proved in Appendix C.4

$$O(C,T;f_{E_i{}^c}) = \sum_{(i,j,c)\in\Omega} w_{ij}^c \ln\left(1+e^{-\lambda\left(\sum_{k=1}^r (C_{ik}-C_{jk})T_{ck}\right)}\right) - \sum_{(i,c)\in(\mathcal{N}\times\mathcal{T})} \ln\left(f_{E_i{}^c}\left(\sum_{k=1}^r C_{ik}T_{ck}\right)\right). \tag{4.5}$$

The problem then reads:

$$\begin{aligned} &\min_{C,T} O(C,T;f_{E_i{}^c}) \\ &1_{n\times 1}^\top C = 0. \end{aligned} \tag{4.6}$$

Several choices are possible for the prior distribution of $E_i{}^c$ and are listed in Table 4.1. For example, $E_i{}^c$ can be assumed logistic zero-mean $E_i{}^c \sim \mathrm{Log}(0,s)$, like it is done for chess ratings based on the BTL model [Log22a]:

$$\begin{aligned} O(C,T;\mathrm{Log}(0,s)) = &\sum_{(i,j,c)\in\Omega} w_{ij}^c \ln\left(1+e^{-\lambda\left(\sum_{k=1}^r (C_{ik}-C_{jk})T_{ck}\right)}\right) \\ &+ \sum_{(i,c)\in(\mathcal{N}\times\mathcal{T})} \left(\frac{1}{s}\sum_{k=1}^r C_{ik}T_{ck} + 2\ln\left(1+e^{-\frac{1}{s}\left(\sum_{k=1}^r C_{ik}T_{ck}\right)}\right)\right). \end{aligned}$$

## 4.5 Prior knowledge on $E$

This section enumerates possible choices for the prior distribution of $E$. More generally, it also enumerates possible choices for the other variables $\Delta E$, $S$, $R$ and $P$. Indeed, regardless of the variable chosen to impose the prior distribution, the prior distribution on the other variables can be deduced thanks to the BTL model equations. However, assuming simple prior distribution on some variable can lead to really complicated and unusual distributions which sometimes hardly make intuitive sense and lack interpretability.

Table 4.1 summarises some prior distributions. Starting from one distribution assumed on one of the variables, the prior distributions implied for the other variables are shown. The derivations can be found in Appendix E. It is worth noticing that most prior distributions from the table are negative log-convex, which make them suitable for the optimization problems throughout this thesis. In particular, this is true for the uniform distribution, for the beta distribution and for the logistic distribution, which are the distributions tested in this thesis.

| ${E_i}^c$, ${E_j}^c$ | $\Delta {E_{ij}}^c$ | ${S_i}^c$, ${S_j}^c$ | ${R_{ij}}^c$ | ${P_{ij}}^c$ |
|---|---|---|---|---|
| Gumbel | **Logistic** | Inverse Exponential | Lomax | **Uniform** |
| $\text{Gumbel}\left(-\frac{\gamma}{\lambda}, \frac{1}{\lambda}\right)$ | $\text{Log}\left(0, \frac{1}{\lambda}\right)$ | $\text{InvExp}\left(e^{\gamma}\right)$ | $\text{Lomax}(1,1)$ | $\text{Uni}(0,1)$ |
| **Logistic** | Diff-Logistics | Burr | ? | Beta-like |
| $\text{Log}\left(0, s\right)$ | $\text{DiffLog}\left(s\right)$ | $\text{Burr}\left(\frac{1}{s\lambda}, 1, 1\right)$ | | $\text{BetaLike}\left(\frac{1}{s\lambda}\right)$ |
| ? | Pow-Logistic | ? | ? | **Beta** |
| | $\text{PowLog}\left(0, \frac{l}{\lambda}, b\right)$ | | | $\text{Beta}(b,b)$ |
| **"Uniform"** | / | / | / | / |
| Impossible | | | | |
| / | / | **"Uniform"** | / | / |
| | | Impossible | | |

Table 4.1: Table of prior distributions, initial assumption in bold, ? for still to be computed distributions, / for not properly defined distributions because uniform distribution on ${E_i}^c$ and ${S_i}^c$ are not properly defined (infinite domain).

Let us note that the prior of ${E_i}^c$ cannot be assumed uniform. Indeed, the uniform distribution is only well-defined on a finite domain, but the domain of ${E_i}^c$ is infinite (from $-\infty$ to $+\infty$). However, considering no prior knowledge can still be done by removing the log-prior term in problem 4.4. In this case, the problem transforms into an ML formulation. For the same reason, a uniform prior prior cannot be assumed on $S$ (its domain is infinite from 0 to $\infty$).

## 4.6 Interpretability of low-rank structure on $E$

Could we interpret physically matrices $C$ and $T$ of the $E = CT^\top$ decomposition? Without any complementary constraints, it seems hard to do. Indeed, as seen in Section 2.3, the decomposition is not unique.

The idea behind the low-rank decomposition of $E$ is that there exist *meta-tournaments*, which represent the possible types of tournaments. Intuitively, we guess that the main feature of a tennis tournament is the surface on which it is played. The low-rank structure should naturally identify those types of tournaments and express each tournament as a combination of meta-tournaments. In order to do that, we can ask $T \geqslant 0$ and $T1_{r\times 1} = 1_{t\times 1}$. Then, $C$ and $T$ could be understood similarly as $H$ and $W$ in Section 4.4.1. Indeed, the element ${E_i}^c$ is the

skill level of a player $i$ at tournament $c$ and is equal to $\sum_{k=1}^{r} C_{ik}T_{ck}$. So, $C_{ik}$ would represent the skill level of player $i$ at tournaments of type $k$, and $T_{ck}$ can be seen as the probability of tournament $c$ being of type $k$, or as the similarity of tournament $c$ with meta-tournament $k$.

However, the initial goal of working on ratings $E$ instead of rating $S$ is precisely to avoid constraints in order to get a better solution. So, it looks odd to add new ones, even if it has some interpretability purposes. A compromise that can be done is to impose only one of those constraints. For example, if $T1_{r\times 1} = 1_{t\times 1}$ is added in formulation 4.6, it means that the rating in tournament $c$ is an affine combination of ratings in meta-tournaments.

## 4.7 BCD algorithm for the low-rank MAP rating formulation

In this section, a Block Coordinate Descent (BCD) algorithm is created in order to solve the `Eratings` problem 4.6.

As seen in Section 2.4.2, the factorization destroys the convexity of the problem. The solution is then to optimize alternatively on $C$ and $T$, because the problem stays block-component-wise convex (see Appendix C.3). The global algorithm can be viewed as a Block Coordinate Descent (BCD) algorithm, also called Gauss-Seidel algorithm [Lyu21]. As we have two block coordinates, it is expected to converge to a stationary point, i.e. to a local optimum [Gri00]. In addition to the optimal value to converge, it is intuitive to ask for the variable $E$ to converge as well, hence the definition of $\delta_t$ in the BCD algorithm:

**Algorithm 1** BCD for `Eratings` problem

**Inputs:**

Confrontation tensor $W \in \mathbb{N}^{n \times n \times t}$

Rank bound $r$ such that $0 < r \leqslant \min\{n, t\}$

Initial players matrix $C_0 \in \mathbb{R}^{n \times r}$ such that $1_{n\times 1}^\top C_0 = 0$

Initial tournaments matrix $T_0 \in \mathbb{R}^{t \times r}$ (such that $T_0 \geqslant 0$, $T_0 1_{r\times 1} = 1_{t\times 1}$)

Initial ratings matrix $E_0 = C_0 {T_0}^\top \in \mathbb{R}^{n \times t}$

BTL parameter $\lambda > 0$,

Prior distribution on ${E_i}^c$ with parameter(s) $\theta$ (identical for all $i, c$) $f_{{E_i}^c}(x; \theta)$

Tolerance $\epsilon > 0$

Maximum number of iterations $n_{\max} > 0$

**Outputs:**

Ratings matrix $E \in \mathbb{R}^{n \times t}$

Players matrix $C \in \mathbb{R}^{n \times r}$

Tournaments matrix $T \in \mathbb{R}^{t \times r}$

Probability tensor $P \in \mathbb{R}^{n \times n \times t}$

**Algorithm:**

$t = 0$

$\delta_t = 2\epsilon$

**while** $\delta_t > \epsilon$ **and** $t < n_{\max}$ **do**

$C_{t+1} = \underset{C}{\operatorname{argmin}}\, O(C, T_t; f_{{E_i}^c})$ such that $1_{n\times 1}^\top C = 0_{r\times 1}$

$T_{t+1} = \underset{T}{\operatorname{argmin}}\, O(C_{t+1}, T; f_{{E_i}^c})$ (such that $T \geqslant 0$, $T 1_{r\times 1} = 1_{t\times 1}$)

$E_{t+1} = C_{t+1} {T_{t+1}}^\top$

$\delta_{t+1} = \|E_{t+1} - E_t\|_F^2$

$t = t + 1$

**end while**

$C = C_t$

$T = T_t$

$E = E_t$

$P$ such that ${P_{ij}}^c = \frac{1}{1+e^{-\lambda({E_i}^c - {E_j}^c)}}$

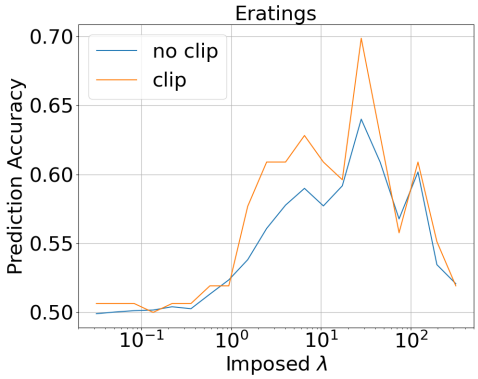
## 4.8 Results

In this last section, we present results for the `Eratings` method introduced in this chapter. In particular, we look at the influence of the method's parameters and of the prior distribution choice. Then, a global comparison is made between all methods presented during

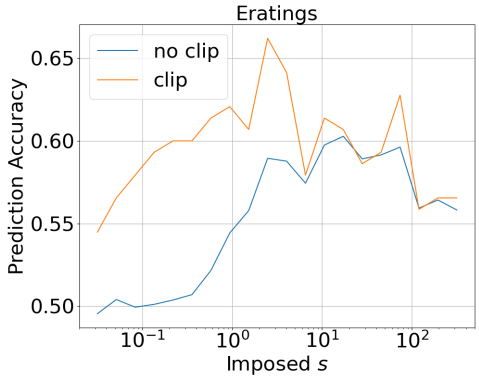
this thesis. Two metrics are examined: the running time and the prediction accuracy over the 2000 to 2016 seasons.

### 4.8.1 Optimal parameters

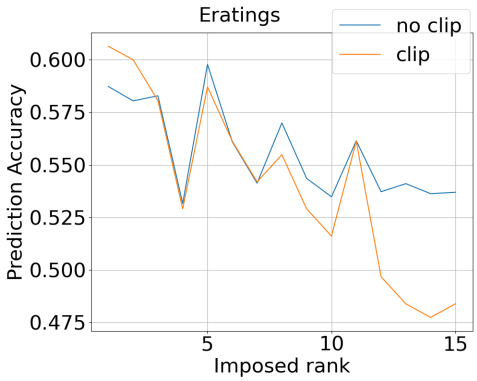
In order to compare the `Eratings` method with the ones from the other chapters, we have to first find the best parameters $\lambda$ (from the BTL formula), $s$ (from the logistic prior distribution) and $r$ (the rank imposed by the BMF).




(a) $\lambda_{\text{opt}} = 10$ for $s = 1$, $r = 2$




(b) $s_{\text{opt}} = 10$ for $\lambda = 1$, $r = 2$




(c) $r_{\text{opt}} = 2$ for $\lambda = 10$, $s = 1$

Figure 4.1: Finding best $\lambda$, $s$ and $r$ for `Eratings`. $n = 50$ players, season 2013, $E_i^c \sim \text{Log}(0, s)$, $n_{\max} = 20$, $\alpha_{\text{te}} = 0.3$.

From Figure 4.1a and 4.1b, it seems that the range of optimal values for $\lambda$ and $s$ is closely related. Indeed, it seems that it is in fact the product $s\lambda$ which is truly important, and not simply the individual values of $\lambda$ and $s$. The optimal range spans $s\lambda \in [3, 100]$, so we will choose $s\lambda = 10$ as the optimal value. A theoretical explanation of the importance of the product is given in Section 4.8.2.

From Figure 4.1c, we can observe that the accuracy grows when $r$ gets small. There

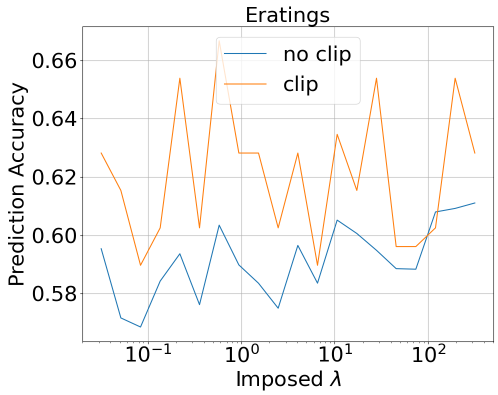
are two factors creating this phenomenon. The first explanation is simply that the low-rank structure seems to be an efficient approach for this problem, which is what this thesis hopes to demonstrate: the lower the rank is, the better the prediction will be. This is true as for a given precision threshold $\epsilon$ (with no maximum number of iterations), the accuracy decreases when $r$ increases. The other explanation involves a slower convergence of the BCD algorithm for larger $r$. Indeed, it turns out that a larger $r$ requires more steps in order to converge, certainly because there are more variables to handle. Then, as the maximum number of iterations $n_{\max} = 20$ stays identical, stopping after $n_{\max}$ in every case means that we get further from the optimal point when $r$ grows. The prediction accuracy then decreases. Experimentally, this is verified as well: in order to reach a given threshold $\epsilon = 10^{-1}$, the BCD algorithm required 6 iterations for $r = 1$, 60 for $r = 2$ and 69 for $r = 3$. However, the importance of each effect in the final result needs to be quantified better and explored deeper.

### 4.8.2 Scaling invariance and product $s\lambda$

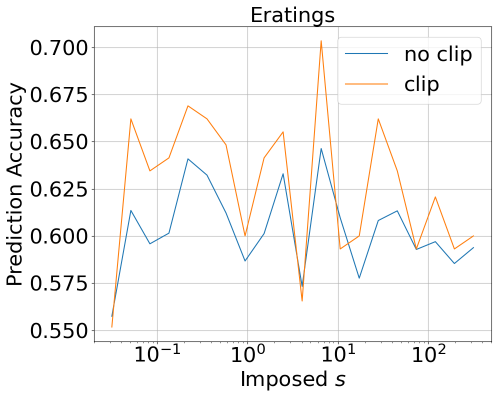
On the one hand, the parameter $\lambda$ encodes the scaling invariance of the distribution of $E$ from the BTL model. $\lambda$ appears only in the term corresponding to the likelihood of the winning probabilities $P$ (the first term in the objective function 4.5). To understand how this term influences the final distribution of $E$, we remind ourselves that the likelihood term of $P$ considers an uniform prior on $P$. It implies on the ratings that $E \sim \text{Gumbel}\left(\frac{-\gamma}{\lambda}, \frac{1}{\lambda}\right)$ (see Table 4.1), which has a standard deviation proportional to $\frac{1}{\lambda}$. This means that the term of maximum likelihood induces a dispersion of the ratings inversely proportional to $\lambda$.

On the other hand, a logistic distribution is assumed as prior: $E \sim \text{Log}\,(0, s)$. This distribution has a standard deviation proportional to $s$. So, the prior term induces a dispersion of the ratings proportional to $s$.

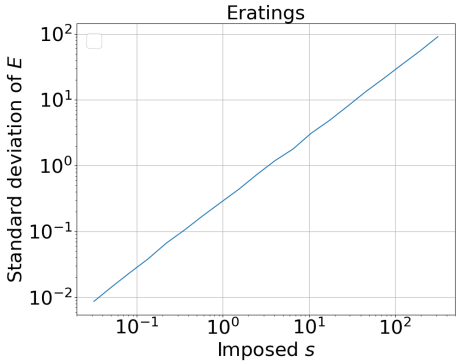
If both terms grow in the same way, then the final distribution should be a scaled version of the previous one. In other words, the only difference is that the standard deviation $\sigma_E$ should be multiplied by the growth factor. In order to make sure that both terms increase in the same manner, a simple condition is to ask for the ratio of their individual deviations $\frac{s}{\frac{1}{\lambda}}$ to stay constant, or equivalently the product $s\lambda$ stays constant.




(a) No optimal $\lambda$




(b) No optimal $s$




(c) Linear relation with $\sigma_E$

Figure 4.2: Keeping the product $\lambda s = 10$ constant. $n = 50$ players, season 2013, $E_i^c \sim \text{Log}(0, s)$, $r = 2$, $n_{\max} = 20$ $\alpha_{\text{te}} = 0.3$.

The experiment which kept product $s\lambda = 10$ constant shows, in Figures 4.2a and 4.2b, a strongly oscillating behaviour of the prediction accuracy when varying $\lambda$ or $s$, without a global increasing or decreasing trend. This indicates that it is only the value of this product which has a deep impact on the accuracy, even if there is a large variance.

However, the product $\lambda s$ has a clear impact on matrix $E$, which is shown in Figure 4.2c. The standard deviation of the ratings $E$ denoted $\sigma_E$ depends linearly on $s$ when $\lambda s$ is fixed ($\sigma_E$ is multiplied by 10 when $s$ is multiplied by 10 and $\lambda$ divided by 10), which was expected from the theoretical analysis.

### 4.8.3 Interpretability

Some experiments have been made in order to verify if the addition of the constraints $T \geqslant 0$ and/or $T1_{r\times 1} = 1_{t\times 1}$ could be beneficial for the interpretability of the low-rank

structure. This turns out not to be the case.

The intuitive idea that *meta-tournaments* should represent different surfaces (clay, grass, hard) is not reflected in matrix $T$. It was expected that, for a rank 3 decomposition, the biggest coefficient between $T_{c1}$, $T_{c2}$ and $T_{c3}$ tells which type the tournament should probably be. However, the biggest coefficients for all tournaments for a given surface do not coincide.

Moreover, the predictions are less accurate than when considering no constraint. This was expected as adding constraints to an optimisation problem can only deteriorate its optimal point. To make things worse, the convergence of the BCD algorithm is empirically much slower with the added constraints.

### 4.8.4 Running time

The running time can heavily vary between methods as showed in Table 4.2. The general rule is that methods calling external solvers (such as cvxpy [cvx]), namely `SDP_NNM`, `LASSO`, `WLASSO`, `MAP_NNM` and `Eratings`, take much longer to run than direct methods. The slowest method is `Eratings`, because the BCD algorithm calls the external solver twice per iteration. In fact, the algorithm is stopped because it reached the maximum number of iterations, not because it converged.

| Method | Run time [s] | Parameters |
|---|---|---|
| `SDP_NNM` | 263.406 | |
| `SoftImpute` | 0.068 | $\tau = 5.0$ |
| iterativeSVD | 0.164 | $r = 2$ |
| `lmafit` | 0.047 | $r = 2$ |
| `kNN` | 0.032 | $k = 10$ |
| `LASSO` | 158.329 | $\tau = 5.0$ |
| `WLASSO` | 191.635 | $\tau = 5.0$ |
| `columns_mean` | 0.000 | |
| `rows_mean` | 0.000 | |
| `MAP_NNM` | 382.213 | $\tau = 5.0$ |
| `Eratings` | 665.599 | $r = 2$, $\lambda = 1.0$, $s = 10.0$, $n_{\max} = 20$ |

Table 4.2: Running time for different algorithms, season 2001

### 4.8.5 Comparison of all methods over the seasons

On Figure 4.3, we compare the accuracy of all methods over the seasons 2000 to 2016. We can observe several elements:

- As shown in Section 2.7.4, depending on the data randomly selected in the testing dataset, the prediction accuracy can vary a lot. Therefore, values in this experiment need to be taken with caution. However, it looks like some seasons were more predictable than others. From the specific runs of this experiment, the worst year for prediction was 2000 (0.51-0.56 % accuracy), and the best was 2012 (0.63-0.70 % accuracy). Surprisingly, the most recent seasons were on average more predictable than the oldest ones.

- The `LASSO` and `WLASSO` methods introduced in Section 2.6 (in green) are among the best methods for almost all seasons. In particular, they seem to outperform classical LRMC algorithms such as `SDP_NNM`, `lmafit` or `SoftImpute`. This is probably due to the fact that the probability constraints are built in these new formulations, while they need to be imposed afterwards for the classical algorithms.

- The weights added to `WLASSO` are conclusive as they allow `WLASSO` to outperform `LASSO` for the majority of the seasons, reaching a peak of 70% for the 2012 season.

- The `MAP_NNM` method introduced in Section 3.3 (in pink) also scores as one of the top methods. However, it seems to be in general a bit less accurate than `LASSO` and `WLASSO` methods.

- The `Eratings` method introduced in Section 4.4.2 (in red) lacks robustness. Indeed, the results vary considerably from year to year: sometimes being the best method (2009) and other times the worst one (2004).

- The naive methods `rows_mean` and `columns_mean` actually do not score as bad as expected, for example by beating more complex methods such as `SDP_NNM`.

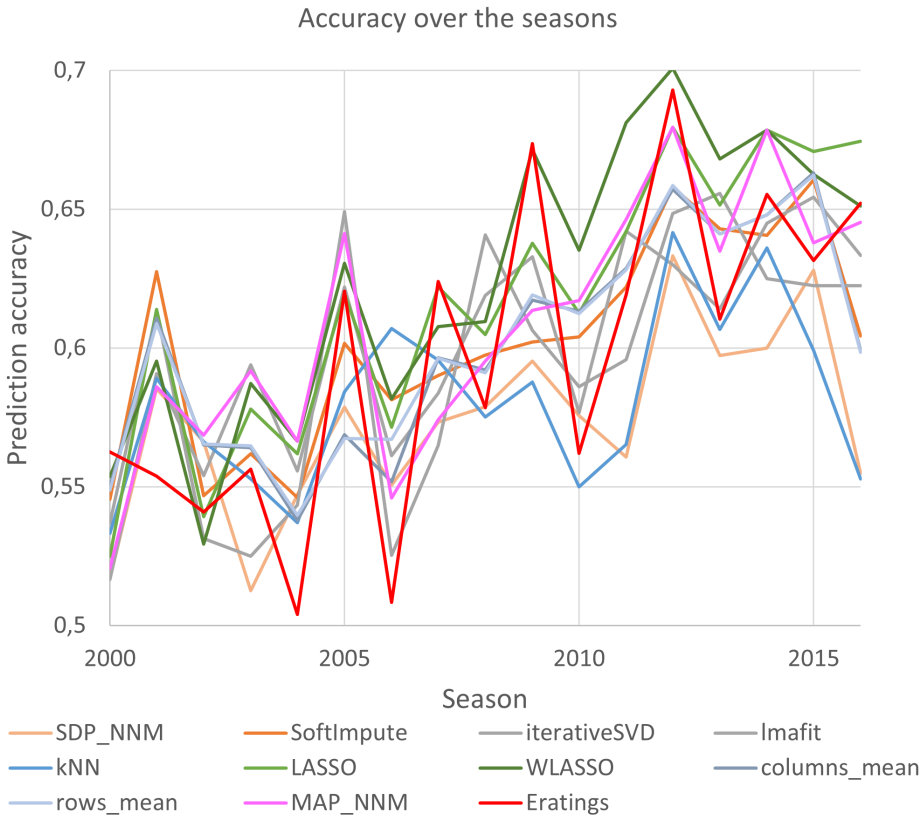



Figure 4.3: Comparison of accuracy over 2000-2016 seasons. $n = 50$ players, $\alpha_{\text{te}} = 0.3$, all hyper-parameters are tuned identically as is Table 4.2. Remark: Despite being independent, points are linked together to improve the readability of the graph.

# Conclusion

## Summary

The hard problem of guessing the outcome of sports events is tackled in this thesis by using low-rank approaches. The problem, which is stated in the Chapter 1, makes two main assumptions. Firstly, there exists a matrix $\widetilde{P}$ which dictates the winning probabilities of each player. Secondly, the sampling of each match of the dataset from the matrix is performed independently. Those assumptions are big limitations of the model. On the one hand, they imply that the only feature influencing the outcome of a confrontation is the pair of players. This excludes plenty of elements which should intuitively matter such as the surface of the match, the current state of fitness of players, the importance of the tournament, the ranking of the players or the results of their previous encounters. On the other hand, winning probabilities are fundamentally constrained in two ways: probabilities are bounded between zero and one and the sum has to be one for each pair of players.

Despite the intrinsic flaws of the probability matrix formulation, two low-rank techniques are tested in order to retrieve an approximation $P$ of the true probability matrix $\widetilde{P}$: matrix completion and MAP estimation. They are compared on the basis of several metrics, but the most important and intuitive one is prediction accuracy. By exploring the properties of this accuracy metric, it turns out that it is maximized when the probabilities are clipped to zero or to one. This possibly counter-intuitive phenomenon has massive consequences on the problem. First, avoiding probabilities to go to the extremes (one or zero) by normalization towards $\frac{1}{2}$ becomes useless. Then, physically, it shows that the only important information is to know which player should most likely win.

Low-rank matrix completion is the first technique used to tackle this problem. The main issue with this approach is the creation of the matrix to complete from the dataset. Three different estimators are tested (ML, MAP and CM) but, in the end, they give all identical results concerning the prediction accuracy. Indeed, adding symmetric prior knowledge in order to normalize the probabilities is useless, as explained before. A second issue is that the low-rank structure of the $P$ matrix is hard to motivate, even if it is rather easy to implement,

using NNM of BMF formulations and even if it seems to give better results. A third concern is that in order to avoid that the matrix becomes too sparse and to keep a manageable size for the solvers, only the most active players are considered. One last problem of classical LRMC methods is that they do not impose any of the probability constraints. In order to integrate them into the problem, two new LRMC formulations are introduced: `LASSO` and `WLASSO`, where the W stands for the weights added to the standard constrained `LASSO`. These new methods, despite running slower, slightly outperform classical matrix completion methods, and the `WLASSO` method ultimately comes on top.

A second attempt consists in formulating directly a MAP low-rank estimation problem, without setting up a matrix to complete in the first place. This leads to two new formulations: `MAP_NNM` and `MAP_BMF`. The second approach is currently not viable because it is too hardly constrained, but the first one works and gets results competing with `LASSO` and `WLASSO` methods. As for the matrix completion problem, the addition of a prior distribution does not improve results as long as it stays symmetric. A solution would be to use an asymmetric prior distribution which is described further in the future work section below.

The last part of this work tries to make a shift of perspective by guessing the ratings of the players instead of their winning probabilities directly. Indeed, the latter can be computed afterwards from the ratings via the famous BTL model. This idea has multiple purposes: it removes the constraints due to probabilities and it allows to include one extra dimension in the problem, which can for example be the tournament. This time, the low-rank assumption is made on the ratings, which gives rise to the hope of gaining more interpretability as the low-rank assumption is this time made on the ratings. However, it is hard to extract any physical sense. Using BMF for the low-rank structure, a new MAP formulation called `Eratings` is introduced, as well as a block-coordinate descent algorithm in order to solve it. Disappointingly, the algorithm is pretty slow and the accuracy of the predictions does not benefit from this new angle of attack. Moreover, assuming zero-mean priors on the ratings leads to the same useless normalization as the previous MAP formulations.

The main contribution of this work is probably the proof that maximizing the prediction accuracy metric requires to clip the winning probabilities to zero or one. It has some rather counter-intuitive but important consequences. First, in order to be right as often as possible, betters should always bet for the supposed best player, never for the weakest one. Then, it implies the uselessness of looking to normalize the probabilities towards $\frac{1}{2}$ in order to make them more realistic. Finally, it has the consequence that symmetric normalizing priors on probability distributions are useless as well.

These low-rank approaches outperforms classical LRMC methods by only a small margin. The main flaw of these techniques is certainly the lack of information taken into account. Considering more features, similarly to what is done in the rest of the literature, could

drastically improve the results. In particular, this could be done by considering asymmetric prior on the probabilities. This clue is the main idea which could lead to future researches and is explained below. Some other unexplored thoughts are detailed as well.

# Future work

1. Use asymmetric (around $\frac{1}{2}$) prior distributions on $P_{ij}$ (for $P_{ij}$ formulations) in order to affect the solution after clipping. This asymmetry reflects what is truly expected from prior knowledge about the players: which one should *a priori* be the best, without considering the results of previous games. This asymmetric prior could be easily extracted from the current ATP ratings of the players or from an ELO rating that they could be attributed. It could even be a function of plenty of other features in order to include them into the problem, as the limited number of features is heavily problematic in the original problem statement. An easy way to model asymmetric prior on probabilities $P_{ij}$ is to use asymmetric Beta distribution $\text{Beta}(a, b), a \neq b$ which is very versatile, easy to manipulate and the derived estimators have short closed expressions. This clue could be the main topic for further research about low-rank approaches for match predictions.

2. The same idea of asymmetric prior could be brought to the rating formulations. This implies that ${E_i}^c$ and ${E_j}^c$ should be independent but not identically distributed. In particular, they could have similar prior distributions, but with different means. Otherwise, the prior distribution on ${\Delta E_{ij}}^c$ is always zero-mean symmetric (see Appendix E.4), which implies that the prior on ${P_{ij}}^c$ is always symmetric around $\frac{1}{2}$, making it useless in the end.

3. In order to maximize the prediction accuracy, it is proven that all probabilities need to be clipped to zero or one (see Section A). This change of perspective could potentially lead to new formulations of the prediction problem, where the objective function directly takes this feature into account. It could possibly have several benefits: more accurate, simpler or faster methods. For example, considering the matrix completion problem, we could constrain the matrix to complete $\overline{P}$ to be clipped: $\overline{P_{ij}} = \mathcal{C}\big(g_{ij}(\mathcal{D})\big)$. Another idea is to convert the MAP formulations to combinatorial problems, by adding the constraint that the variable matrix is binary: $P \in \{0, 1\}^{n \times n}$.

4. Let us consider the BTL model without any dimensional extension. There should exist an ELO-like rating vector $E$. The matrix $\Delta E$ can be rewritten in terms of this vector: $\Delta E = E\mathbf{1}^\top - \mathbf{1}E^\top$. This matrix has rank 2 (this can be deduced from Appendix B.4). Therefore, it could be interesting to apply matrix completion with an imposed rank of

2 on this matrix. In order to compute the known elements, the idea is to first compute $\overline{P_{ij}}$ for $(i,j) \in \Omega$ with estimators described in this work, and then to apply the inverse BTL formula to get $\overline{\Delta E_{ij}}$. Note that this is already a relaxation as a rank two matrix does not need to be in the form $E\mathbf{1}^\top - \mathbf{1}E^\top$. However, it has the advantage of removing all constraints from the probability matrix formulations while having a motivation for the low-rank structure.

# Bibliography


[AAM14] P.-A. Absil, Luca Amodei, and Gilles Meyer. Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. *Computational Statistics*, 29(3-4):569–590, June 2014. URL: http://link.springer.com/10.1007/s00180-013-0441-6, doi:10.1007/s00180-013-0441-6.

[AMS08] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*, volume 78. December 2008. Journal Abbreviation: Mathematics of Computation - Math. Comput. Publication Title: Mathematics of Computation - Math. Comput. doi:10.1515/9781400830244.

[ANdJ21] ANJ Autorité Nationale de Jeux. Rapports 2021-2021 | ANJ, 2021. URL: https://anj.fr/sites/default/files/2021-07/Rapport_ANJ_web.pdf.

[ATP22] ATP Tour, May 2022. Page Version ID: 1088905200. URL: https://en.wikipedia.org/w/index.php?title=ATP_Tour&oldid=1088905200.

[Bay22] Bayes' theorem, May 2022. Page Version ID: 1089020494. URL: https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=1089020494.

[Bet22] Beta distribution, May 2022. Page Version ID: 1086717817. URL: https://en.wikipedia.org/w/index.php?title=Beta_distribution&oldid=1086717817.

[Bev15] Martin Bevc. Predicting the Outcome of Tennis Matches From Point-by-Point Data. *undefined*, 2015. URL: https://www.semanticscholar.org/paper/Predicting-the-Outcome-of-Tennis-Matches-From-Data-Bevc/1cbe1beca36298e4aec813bf967d5aab034c7dcb.

[BTL22] Bradley–Terry model, February 2022. Page Version ID: 1073657694. URL: https://en.wikipedia.org/w/index.php?title=Bradley%E2%80%93Terry_model&oldid=1073657694.

[Buu] Douwe Buursma. Predicting sports events from past results Towards effective betting on football matches | Semantic Scholar. URL: https://www.semanticscholar.org/paper/Predicting-sports-events-from-past-results-Towards-Buursma/5e22c4362df3b0accbe04517c41848a2b229efd1.

[BZA20] Thomas Bendokat, Ralf Zimmermann, and P.-A. Absil. A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects. Technical Report arXiv:2011.13699, arXiv, December 2020. arXiv:2011.13699 [cs, math] type: article. URL: http://arxiv.org/abs/2011.13699, doi:10.48550/arXiv.2011.13699.

[CCS08] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. A Singular Value Thresholding Algorithm for Matrix Completion. Technical Report arXiv:0810.3286, arXiv, October 2008. arXiv:0810.3286 [math] type: article. URL: http://arxiv.org/abs/0810.3286, doi:10.48550/arXiv.0810.3286.

[CDlTCB13] Ricardo Cabral, Fernando De la Torre, João P. Costeira, and Alexandre Bernardino. Unifying Nuclear Norm and Bilinear Factorization Approaches for Low-Rank Matrix Decomposition. In *2013 IEEE International Conference on Computer Vision*, pages 2488–2495, December 2013. ISSN: 2380-7504. doi:10.1109/ICCV.2013.309.

[Cle21] Clerni. Singular Values and Matrix Rank, January 2021. URL: https://math.stackexchange.com/q/3967922.

[CR08] Emmanuel J. Candes and Benjamin Recht. Exact Matrix Completion via Convex Optimization. Technical Report arXiv:0805.4471, arXiv, May 2008. arXiv:0805.4471 [cs, math] type: article. URL: http://arxiv.org/abs/0805.4471, doi:10.48550/arXiv.0805.4471.

[cvx] Welcome to CVXPY 1.2 — CVXPY 1.2 documentation. URL: https://www.cvxpy.org/index.html.

[CW22] Zhaoliang Chen and Shiping Wang. A review on matrix completion for recommender systems. *Knowledge and Information Systems*, 64(1):1–34, January 2022. doi:10.1007/s10115-021-01629-6.

[DM10] Wei Dai and Olgica Milenkovic. SET: an algorithm for consistent matrix completion. Technical Report arXiv:0909.2705, arXiv, February 2010. arXiv:0909.2705 [cs, math] type: article. URL: http://arxiv.org/abs/0909.2705, doi:10.48550/arXiv.0909.2705.

[DS20] Alexander De Seranno. Predicting Tennis Matches Using Machine Learning., 2020.

[DV] Hazan Deglayan and Simon Vary. lmafit.py.

[ELO] Chess rating systems • lichess.org. URL: https://lichess.org/page/rating-systems.

[Faz02] Maryam Fazel. *Matrix rank minimization with applications*. PhD Thesis, PhD thesis, Stanford University, 2002.

[Fel] Alex Rubinsteyn Feldman, Sergey. fancyimpute: Matrix completion and feature imputation algorithms. URL: https://github.com/iskandr/fancyimpute.

[FNP+19] Simon Foucart, Deanna Needell, Reese Pathak, Yaniv Plan, and Mary Wootters. Weighted matrix completion from non-random, non-uniform sampling patterns. Technical Report arXiv:1910.13986, arXiv, October 2019. arXiv:1910.13986 [cs, math, stat] type: article. URL: http://arxiv.org/abs/1910.13986, doi:10.48550/arXiv.1910.13986.

[fra20] franck. The history of Sport Bet in the world, October 2020. URL: https://www.fanstorpedo.com/the-history-of-sport-bet-in-the-world/.

[FRW11] Massimo Fornasier, Holger Rauhut, and Rachel Ward. Low-rank matrix recovery via iteratively reweighted least squares minimization. Technical Report arXiv:1010.2471, arXiv, July 2011. arXiv:1010.2471 [math] type: article. URL: http://arxiv.org/abs/1010.2471, doi:10.48550/arXiv.1010.2471.

[Gri00] L. Grippo. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, April 2000. doi:10.1016/S0167-6377(99)00074-7.

[Gum22] Gumbel distribution, January 2022. Page Version ID: 1066253732. URL: https://en.wikipedia.org/w/index.php?title=Gumbel_distribution&oldid=1066253732.

[GXM+17] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted Nuclear Norm Minimization and Its Applications to Low Level Vision. *International Journal of Computer Vision*, 121(2):183–208, January 2017. doi:10.1007/s11263-016-0930-5.

[Hac12] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Science & Business Media, February 2012. Google-Books-ID: a5P71o6xcNMC.

[Her22] Julien Herman. Mathematics of tennis ranking: dynamical aspects and game outcome prediction by optimization methods. Master's thesis, UCLouvain, June 2022.

[HH09] Justin P. Haldar and Diego Hernando. Rank-Constrained Solutions to Linear Matrix Equations Using PowerFactorization. *IEEE Signal Processing Letters*, 16(7):584–587, July 2009. Conference Name: IEEE Signal Processing Letters. doi:10.1109/LSP.2009.2018223.

[HLB20] Wooseok Ha, Haoyang Liu, and Rina Foygel Barber. An equivalence between critical points for rank constraints versus low-rank factorizations. Technical Report arXiv:1812.00404, arXiv, December 2020. arXiv:1812.00404 [math] type: article. URL: http://arxiv.org/abs/1812.00404, doi:10.48550/arXiv.1812.00404.

[HZY+13] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, September 2013. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2012.271.

[Inv] InverseExponential: The Inverse Exponential Distribution in actuar: Actuarial Functions and Heavy Tailed Distributions. URL: https://rdrr.io/cran/actuar/man/InverseExponential.html.

[JNS12] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank Matrix Completion using Alternating Minimization. Technical Report arXiv:1212.0467, arXiv, December 2012. arXiv:1212.0467 [cs, math, stat] type: article. URL: http://arxiv.org/abs/1212.0467, doi:10.48550/arXiv.1212.0467.

[LAS22] Lasso (statistiques), April 2022. Page Version ID: 193130650. URL: https://fr.wikipedia.org/w/index.php?title=Lasso_(statistiques)&oldid=193130650.

[LB09] Kiryung Lee and Yoram Bresler. ADMiRA: Atomic Decomposition for Minimum Rank Approximation. Technical Report arXiv:0905.0044, arXiv, June 2009. arXiv:0905.0044 [cs, math] type: article. URL: http://arxiv.org/abs/0905.0044, doi:10.48550/arXiv.0905.0044.

[LCM13] Zhouchen Lin, Minming Chen, and Yi Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *Journal of Struc-*

*tural Biology*, 181(2):116–127, February 2013. arXiv:1009.5055 [cs, math]. URL: http://arxiv.org/abs/1009.5055, doi:10.1016/j.jsb.2012.10.010.

[LHSZ19] Xiao Peng Li, Lei Huang, Hing Cheung So, and Bo Zhao. A Survey on Matrix Completion: Perspective of Signal Processing. Technical Report arXiv:1901.10885, arXiv, May 2019. arXiv:1901.10885 [eess] type: article. URL: http://arxiv.org/abs/1901.10885, doi:10.48550/arXiv.1901.10885.

[Log22a] Logistic distribution, May 2022. Page Version ID: 1088866751. URL: https://en.wikipedia.org/w/index.php?title=Logistic_distribution&oldid=1088866751.

[Log22b] Logit, April 2022. Page Version ID: 1083765870. URL: https://en.wikipedia.org/w/index.php?title=Logit&oldid=1083765870.

[Lom21] Lomax distribution, August 2021. Page Version ID: 1041195491. URL: https://en.wikipedia.org/w/index.php?title=Lomax_distribution&oldid=1041195491.

[Lyu21] Hanbaek Lyu. Convergence and complexity of block coordinate descent with diminishing radius for nonconvex optimization. Technical Report arXiv:2012.03503, arXiv, October 2021. arXiv:2012.03503 [math, stat] type: article. URL: http://arxiv.org/abs/2012.03503, doi:10.48550/arXiv.2012.03503.

[mas22] massionb. Master_thesis_bastien_massion, June 2022. original-date: 2022-06-06T17:48:46Z. URL: https://github.com/massionb/Master_thesis_Bastien_Massion.

[Mat] Matrix product and rank. URL: https://www.statlect.com/matrix-algebra/matrix-product-and-rank.

[MF12] Karthik Mohan and Maryam Fazel. Iterative reweighted algorithms for matrix rank minimization. *The Journal of Machine Learning Research*, 13(1):3441–3473, November 2012.

[MGC11] Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1):321–353, June 2011. doi:10.1007/s10107-009-0306-5.

[MHT10] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine*

*Learning Research*, 11(80):2287–2322, 2010. URL: http://jmlr.org/papers/v11/mazumder10a.html.

[Net09] Netflix Prize: Forum / Grand Prize awarded to team BellKor's Pragmatic Chaos, September 2009. URL: https://web.archive.org/web/20090924184639/http://www.netflixprize.com/community/viewtopic.php?id=1537.

[Net22] Netflix Prize, April 2022. Page Version ID: 1083006521. URL: https://en.wikipedia.org/w/index.php?title=Netflix_Prize&oldid=1083006521.

[NKS19] Luong Trung Nguyen, Junhan Kim, and Byonghyo Shim. Low-Rank Matrix Completion: A Contemporary Survey. *IEEE Access*, 7:94215–94237, 2019. Conference Name: IEEE Access. doi:10.1109/ACCESS.2019.2928130.

[OGA22] Guillaume Olikier, Kyle A. Gallivan, and P.-A. Absil. An apocalypse-free first-order low-rank optimization algorithm. Technical Report arXiv:2201.03962, arXiv, January 2022. arXiv:2201.03962 [cs, math] type: article. URL: http://arxiv.org/abs/2201.03962, doi:10.48550/arXiv.2201.03962.

[Pra] Robin Praet. Predicting Sport Results. page 73.

[RYL+18] Andy Ramlatchan, Mengyun Yang, Quan Liu, Min Li, Jianxin Wang, and Yaohang Li. A survey of matrix completion methods for recommendation systems. *Big Data Mining and Analytics*, 1(4):308–323, December 2018. Conference Name: Big Data Mining and Analytics. doi:10.26599/BDMA.2018.9020008.

[Sch22] Matrix norm, May 2022. Page Version ID: 1087093764. URL: https://en.wikipedia.org/w/index.php?title=Matrix_norm&oldid=1087093764.

[Sig22] Sigmoid function, June 2022. Page Version ID: 1091009275. URL: https://en.wikipedia.org/w/index.php?title=Sigmoid_function&oldid=1091009275.

[TCS+01] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics (Oxford, England)*, 17(6):520–525, June 2001. doi:10.1093/bioinformatics/17.6.520.

[TW16] Jared Tanner and Ke Wei. Low rank matrix completion by alternating steepest descent methods. *Applied and Computational Harmonic Analysis*, 40(2):417–429, March 2016. URL: https://www.sciencedirect.com/

science/article/pii/S1063520315001062, doi:10.1016/j.acha.2015.08.003.

[TY10] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Scopus*, 2010. Accepted: 2014-10-28T02:29:59Z. URL: https://scholarbank.nus.edu.sg/handle/10635/102811.

[Van12] Bart Vandereycken. Low-rank matrix completion by Riemannian optimization—extended version. Technical Report arXiv:1209.3834, arXiv, September 2012. arXiv:1209.3834 [math] type: article. URL: http://arxiv.org/abs/1209.3834, doi:10.48550/arXiv.1209.3834.

[WLG19] Leighton Vaughan Williams, Chunping Liu, and Hannah Gerrard. How well do Elo-based ratings predict professional tennis matches? Technical Report 2019/03, Economics, Nottingham Business School, Nottingham Trent University, June 2019. Publication Title: NBS Discussion Papers in Economics. URL: https://ideas.repec.org/p/nbs/wpaper/2019-03.html.

[WYZ12] Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, December 2012. doi:10.1007/s12532-012-0044-1.

[XTFF19] Rui Xia, Vincent Y. F. Tan, Louis Filstroff, and Cédric Févotte. A Ranking Model Motivated by Nonnegative Matrix Factorization with Applications to Tennis Tournaments. Technical Report arXiv:1903.06500, arXiv, June 2019. arXiv:1903.06500 [cs, eess, stat] type: article. URL: http://arxiv.org/abs/1903.06500, doi:10.48550/arXiv.1903.06500.

[XYWZ12] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin Zhang. An Alternating Direction Algorithm for Matrix Completion with Nonnegative Factors. *Frontiers of Mathematics in China*, 7(2):365–384, April 2012. arXiv:1103.1168 [cs, math]. URL: http://arxiv.org/abs/1103.1168, doi:10.1007/s11464-012-0194-5.

[ZCW18] Xiaoxia Zhang, Degang Chen, and Kesheng Wu. Incremental nonnegative matrix factorization based on correlation and graph regularization for matrix completion. *International Journal of Machine Learning and Cybernetics*, 10(6), March 2018. Institution: Lawrence Berkeley National Lab. (LBNL),

Berkeley, CA (United States). National Energy Research Scientific Computing Center (NERSC). URL: https://www.osti.gov/biblio/1526987, doi:10.1007/s13042-018-0808-7.

[ZS18] Wen-Jun Zeng and Hing Cheung So. Outlier-Robust Matrix Completion via $\ell _p$ -Minimization. *IEEE Transactions on Signal Processing*, 66(5):1125–1140, March 2018. Conference Name: IEEE Transactions on Signal Processing. doi:10.1109/TSP.2017.2784361.

[ZZH+21] Qiyun Zhang, Xuyun Zhang, Hongsheng Hu, Caizhong Li, Yinping Lin, and Rui Ma. Sports match prediction model for training and exercise using attention-based LSTM network. *Digital Communications and Networks*, August 2021. URL: https://www.sciencedirect.com/science/article/pii/S2352864821000602, doi:10.1016/j.dcan.2021.08.008.

# Appendix A

# Prediction accuracy metric $\mathcal{A}$

## A.1 Definition

A way of defining how well a method performs at predicting outcomes of matches consists in making predictions for the test set and comparing it with the real data.

Intuitively, one could argue that if we find $P_{ij}$, which we hope is close to the true $\widetilde{P_{ij}}$, then we should predict the victory of player $i$ with a probability $P_{ij}$ and its defeat with a probability $1 - P_{ij}$.

It turns out that this predictive pattern is suboptimal, even in the case of perfect reconstruction ($P_{ij} = \widetilde{P_{ij}}$). Let us demonstrate where this claim comes from and let us find out what the optimal strategy actually is.

First, the forecasting $F_{ij,k}$ of the result of the $k^{\text{th}}$ match between players $i$ and $j$ is defined as a Bernouilli random variable with parameter $P_{ij}$. $F_{ij,k} = 1$ when we predict that player $i$ will win the match, $F_{ij,k} = 0$ when we predict that player $i$ will loose the match:

$$F_{ij,k} \sim \text{Ber}(P_{ij}) \iff \begin{cases} \mathbb{P}[F_{ij,k} = 1] = P_{ij}, \\ \mathbb{P}[F_{ij,k} = 0] = 1 - P_{ij}. \end{cases}$$

According to the definitions, a correct prediction happens when $f_{ij,k} = d_{ij,k}$, for a match $d_{ij,k} \in \mathcal{D}_{\text{te}}$. On the contrary, a wrong prediction means that $f_{ij,k} \neq d_{ij,k}$. We can define a new binary random variable $C_{ij,k}$ which is equal to 1 if the prediction is correct and 0 otherwise. Its distribution is not trivial and is discussed further. We can now define the global prediction accuracy $\mathcal{A}$ on a dataset $\mathcal{D}$ as the ratio of the number of correctly guessed

games over the total number of matches:

$$\mathcal{A}(P) = \frac{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} \sum\limits_{k=1}^{m_{ij}} c_{ij,k}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}}.$$

## A.2 Explicit expression

In order to be able to derive an expression for the global accuracy, we need to find the distribution of $C_{ij,k}$. In other words, we need to find what is the probability for a good prediction ($\mathbb{P}[C_{ij,k} = 1]$) and for a wrong prediction ($\mathbb{P}[C_{ij,k} = 0]$). To simplify the analysis, we will assume that $F_{ij,k}$ and $D_{ij,k}$ are independent random variables: $\mathbb{P}[F_{ij,k} = f_{ij,k} \cap D_{ij,k} = d_{ij,k}] = \mathbb{P}[F_{ij,k} = f_{ij,k}]\mathbb{P}[D_{ij,k} = d_{ij,k}]$. Four cases can occur.

1. Win correctly predicted: $f_{ij,k} = 1, d_{ij,k} = 1$, with probability $P_{ij}\widetilde{P_{ij}}$.
2. Win wrongly predicted: $f_{ij,k} = 0, d_{ij,k} = 1$, with probability $(1 - P_{ij})\widetilde{P_{ij}}$.
3. Loss wrongly predicted: $f_{ij,k} = 1, d_{ij,k} = 0$, with probability $P_{ij}(1 - \widetilde{P_{ij}})$.
4. Loss correctly predicted: $f_{ij,k} = 0, d_{ij,k} = 0$, with probability $(1 - P_{ij})(1 - \widetilde{P_{ij}})$.

Therefore, we can express the probability of a correct prediction for the $k^{\text{th}}$ match between $i$ and $j$, that we will consider as our objective function:

$$\begin{aligned}
\mathbb{P}[C_{ij,k} = 1] &= \mathbb{P}[F_{ij,k} = D_{ij,k}] \\
&= \mathbb{P}[F_{ij,k} = 1, D_{ij,k} = 1] + \mathbb{P}[F_{ij,k} = 0, D_{ij,k} = 0] \\
&= P_{ij}\widetilde{P_{ij}} + (1 - P_{ij})(1 - \widetilde{P_{ij}}) \\
&= 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}.
\end{aligned}$$

From this, we can conclude that $C_{ij,k}$ is a Bernouilli random variable with parameter $1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}$. As $\widetilde{P_{ij}}$ is fixed, $C_{ij,k}$ depends only on the variable $P_{ij}$:

$$C_{ij,k} \sim \text{Ber}(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}) \iff \begin{cases} \mathbb{P}[C_{ij,k} = 1] = 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}, \\ \mathbb{P}[C_{ij,k} = 0] = P_{ij} + \widetilde{P_{ij}} - 2P_{ij}\widetilde{P_{ij}}. \end{cases}$$

Moreover, $C_{ij,k} = C_{ji,k}$ by symmetry. Here is a little proof:

$$\mathbb{P}[C_{ji,k} = 1] = \mathbb{P}[F_{ji,k} = D_{ji,k}]$$

$$
\begin{aligned}
&= \mathbb{P}[F_{ji,k} = 1, D_{ji,k} = 1] + \mathbb{P}[F_{ji,k} = 0, D_{ji,k} = 0] \\
&= P_{ji}\widetilde{P_{ji}} + (1 - P_{ji})(1 - \widetilde{P_{ji}}) \\
&= (1 - P_{ij})(1 - \widetilde{P_{ij}}) + P_{ij}\widetilde{P_{ij}} \\
&= 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}} \\
&= \mathbb{P}[C_{ij,k} = 1].
\end{aligned}
$$

Let us now develop the expression of the expected prediction accuracy $\mathcal{A}$. As a remainder, the expectation of a Bernouilli random variable $B$ is equal to its paramater $p$: $\mathbb{E}[B] = 1.p + 0.(1-p) = p$:

$$
\begin{aligned}
\mathbb{E}[\mathcal{A}(P)] &= \mathbb{E}\left[\frac{\sum_{\substack{(i,j)\in\Omega \\ i>j}} \sum_{k=1}^{m_{ij}} c_{ij,k}}{\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}}\right] \\
&= \mathbb{E}\left[\sum_{\substack{(i,j)\in\Omega \\ i>j}} \left(\frac{1}{\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \sum_{k=1}^{m_{ij}} c_{ij,k}\right)\right] \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \left(\frac{1}{\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \sum_{k=1}^{m_{ij}} \mathbb{E}[c_{ij,k}]\right) \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \left(\frac{1}{\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \sum_{k=1}^{m_{ij}} 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right) \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right).
\end{aligned}
$$

## A.3 Optimal strategy on average I

The goal is to find the whole matrix $P$ that maximizes the expected accuracy of prediction. This can be formulated as the following optimization problem:

$$\begin{aligned}\underset{P\in\mathcal{P}}{\operatorname{argmax}}\,\mathbb{E}\left[\mathcal{A}(P)\right] &= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \sum_{\substack{(i,j)\in\Omega\\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}} \left(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right) \\ &= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \sum_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij} \left(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right).\end{aligned}$$

The last optimization problem is separable: it is equivalent to several smaller independent optimization problems which are easier to solve. In our case, each subproblem only involves one scalar variable $P_{ij}$:

$$\begin{aligned}P^* &= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \sum_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij} \left(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right) \\ {P_{ij}}^* &= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}}\, m_{ij} \left(1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}}\right) && \forall (i,j)\in\Omega, i>j \\ &= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}}\, 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}} && \forall (i,j)\in\Omega, i>j.\end{aligned}$$

In the end, $P$ needs to be found in order to maximize the probability of a good prediction for every pair $(i,j)$ of players, which is a natural equivalence for maximizing the total accuracy over all pairs $(i,j)$. Moreover, we can notice that the values $P_{ij}$ for $(i,j)\in\Omega^c$ do not influence the accuracy of the predictions, which is logical since there is no match to predict for those pairs. Finally, we only need the values of $P_{ij}$ with $(i,j)\in\Omega$ but such that $i>j$ because the other ones are simply computed as $P_{ji} = 1 - P_{ij}$.

To finally find the optimal $P_{ij}$, we can solve analytically one of the subproblems. The objective value of these subproblems now corresponds to the probability of good prediction for a game between the two interested players:

$$\begin{aligned}\underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}}\, 1 - P_{ij} - \widetilde{P_{ij}} + 2P_{ij}\widetilde{P_{ij}} &= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}} -P_{ij} + 2P_{ij}\widetilde{P_{ij}} \\ &= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}}\, P_{ij}\left(2\widetilde{P_{ij}} - 1\right).\end{aligned}$$

Therefore, we find the optimal value of $P_{ij}$ with respect to $\widetilde{P_{ij}}$:

$${P_{ij}}^* = \begin{cases} 0 & \text{if} \quad 0 \leqslant \widetilde{P_{ij}} < \frac{1}{2} \\ \frac{1}{2} & \text{if} \quad \widetilde{P_{ij}} = \frac{1}{2} \\ 1 & \text{if} \quad \frac{1}{2} < \widetilde{P_{ij}} \leqslant 1. \end{cases}$$

It is noticeable that we always satisfy the equality ${P_{ij}}^* = 1 - {P_{ji}}^*$ for all $(i,j) \in \Omega$. Besides, when $\widetilde{P_{ij}} = \frac{1}{2}$, the optimal value of $P_{ij}$ is not constrained : ${P_{ij}}^* \in [0,1]$. However, we choose ${P_{ij}}^* = \frac{1}{2}$ to preserve symmetry around $\frac{1}{2}$ and make sure that the constraints ${P_{ii}}^* = \frac{1}{2}$ are satisfied. Another choice is ${P_{ij}}^* = 0$ or ${P_{ij}}^* = 1$, as it allows ${P_{ij}}^*$ to be a simple binary variable.

## A.4 Clipping operator $\mathcal{C}$

In order to simplify the solution, we define the *clipping operator* $\mathcal{C}$ on a probability $P_{ij}$ or on the whole probability matrix $P$ (by element-wise application):

$$\big[\mathcal{C}(P)\big]_{ij} = \mathcal{C}(P_{ij}) = \begin{cases} 0 & \text{if} \quad 0 \leqslant P_{ij} < \frac{1}{2} \\ \frac{1}{2} & \text{if} \quad P_{ij} = \frac{1}{2} \\ 1 & \text{if} \quad \frac{1}{2} < P_{ij} \leqslant 1. \end{cases}$$

The solution is therefore:

$$P^* = \mathcal{C}(\widetilde{P}).$$

We can nicely interpret physically this result. The goal of our problem is to find which player is stronger in each pair $(i,j)$, and then always bet that he will win the coming confrontations.

This seems to contradict the basic idea of the existence of the matrix $\widetilde{P}$, when our objective was to reconstruct $P$ such $P \approx \widetilde{P}$. Instead, according to this metric $\mathcal{A}$, the correct objective would be to find $P \approx \mathcal{C}(\widetilde{P})$.

Fortunately, there is a way to conciliate those two paradoxical claims and continue to try methods such that $P \approx \widetilde{P}$. We have to realize that, on the one hand, the usual objective is stronger than the new one, and, on the other hand, that we can find back a good solution for the weaker objective if we already own one for the strongest one. Indeed, if we have found a matrix $P \approx \widetilde{P}$, then we simply apply the clipping operator in order to get a matrix $P^{\text{cl}}$ approaching the optimal when maximizing $\mathcal{A}$:

$$P^{\text{cl}} = \mathcal{C}(P) \approx \mathcal{C}(\widetilde{P}) = P^*.$$

Mathematically, the following implication is true, but its inverse is false in general, proving that the usual objective is stronger:

$$P \approx \widetilde{P} \qquad \Longrightarrow \qquad \mathcal{C}(P) \approx \mathcal{C}(\widetilde{P}).$$

During the clipping process, we lose information: matrix $P^{\text{cl}} = \mathcal{C}(P)$ is less complex than the original $P$. Indeed, each $P_{ij}$ can have a uncountable number of values on the interval $[0, 1]$, while $P'_{ij}$ can basically only take 3 values in $\{0, \frac{1}{2}, 1\}$. Physically, we go from knowing what is the percentage of player $i$ winning against $j$, to simply knowing whether he is more likely to win or to lose: we do not care about how much stronger or weaker the player is with respect to his opponent.

## A.5 Optimal strategy on average II

What is the best accuracy theoretically reachable on average? As we have seen, the prediction accuracy seems to be fundamentally linked with $\widetilde{P_{ij}}$. In some sense, the outcome of a game is fundamentally probabilistic and hereby unpredictable. This whole development still follows from the assumptions that the whole dataset depends only on the mystical probability $\widetilde{P_{ij}}$, determined only by the identities of players $i$ and $j$. It seems rather logical that the expected accuracy could be improved by using more information and getting rid of this assumption:

$$
\begin{aligned}
\mathbb{E}\left[\mathcal{A}(P)\right] &\leqslant \max_{P\in\mathcal{P}} \mathbb{E}\left[\mathcal{A}(P)\right] \\
&= \mathbb{E}\left[\mathcal{A}\big(\mathcal{C}(\widetilde{P})\big)\right] \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - \big[\mathcal{C}(\widetilde{P})\big]_{ij} - \widetilde{P_{ij}} + 2\big[\mathcal{C}(\widetilde{P})\big]_{ij}\widetilde{P_{ij}}\right) \\
&= \sum_{\substack{(i,j)\in\Omega, i>j \\ 0\leqslant\widetilde{P_{ij}}\leqslant\frac{1}{2}}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - \big[\mathcal{C}(\widetilde{P})\big]_{ij} - \widetilde{P_{ij}} + 2\big[\mathcal{C}(\widetilde{P})\big]_{ij}\widetilde{P_{ij}}\right) \\
&\qquad + \sum_{\substack{(i,j)\in\Omega, i>j \\ \frac{1}{2}<\widetilde{P_{ij}}\leqslant 1}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - \big[\mathcal{C}(\widetilde{P})\big]_{ij} - \widetilde{P_{ij}} + 2\big[\mathcal{C}(\widetilde{P})\big]_{ij}\widetilde{P_{ij}}\right) \\
&= \sum_{\substack{(i,j)\in\Omega, i>j \\ 0\leqslant\widetilde{P_{ij}}\leqslant\frac{1}{2}}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - \widetilde{P_{ij}}\right) + \sum_{\substack{(i,j)\in\Omega, i>j \\ \frac{1}{2}<\widetilde{P_{ij}}\leqslant 1}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(\widetilde{P_{ij}}\right) \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(\frac{1}{2} + \left|\frac{1}{2} - \widetilde{P_{ij}}\right|\right).
\end{aligned}
$$

We can compare the accuracy to what the intuitive method gives, i.e. finding the perfect

reconstruction $P_{ij} = \widetilde{P_{ij}}$:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{A}\left(\widetilde{P}\right)\right] &= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - \widetilde{P_{ij}} - \widetilde{P_{ij}} + 2\widetilde{P_{ij}}\widetilde{P_{ij}}\right) \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(1 - 2\widetilde{P_{ij}} + 2\widetilde{P_{ij}}^2\right) \\
&= \sum_{\substack{(i,j)\in\Omega \\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \left(\frac{1}{2} + 2\left(\frac{1}{2} - \widetilde{P_{ij}}\right)^2\right).
\end{aligned}$$

## A.6 Experimental optimal strategy

During experiments, $\widetilde{P}$ is unknown. Hence, the average prediction accuracy $\mathbb{E}\left[\mathcal{A}\left(P\right)\right]$ seems uncomputable. However, once our data test set is revealed, we can compute the experimental optimal prediction accuracy. When $d_{ij,k} = 0$, then $c_{ij,k} = 1$ when $f_{ij,k} = 0$, which happens with probability $1 - P_{ij}$. When $d_{ij,k} = 1$, then $c_{ij,k} = 1$ when $f_{ij,k} = 1$, which happens with probability $P_{ij}$. What is the maximum experimental prediction accuracy?

$$\begin{aligned}
P^* = \underset{P\in\mathcal{P}}{\operatorname{argmax}}\, \mathcal{A}(P) &= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \frac{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} \sum\limits_{k=1}^{m_{ij}} c_{ij,k}}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \\
&= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \frac{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} \left(\sum\limits_{\substack{1\leqslant k\leqslant m_{ij} \\ d_{ij,k}=0}} c_{ij,k} + \sum\limits_{\substack{1\leqslant k\leqslant m_{ij} \\ d_{ij,k}=1}} c_{ij,k}\right)}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}} \\
&= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \frac{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} \left(\sum\limits_{\substack{1\leqslant k\leqslant m_{ij} \\ d_{ij,k}=0}} (1 - P_{ij}) + \sum\limits_{\substack{1\leqslant k\leqslant m_{ij} \\ d_{ij,k}=1}} P_{ij}\right)}{\sum\limits_{\substack{(i,j)\in\Omega \\ i>j}} m_{ij}}
\end{aligned}$$

$$
\begin{aligned}
&= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \frac{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} \Big((m_{ij}-w_{ij})(1-P_{ij})+w_{ij}P_{ij}\Big)}{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}} \\
&= \underset{P\in\mathcal{P}}{\operatorname{argmax}} \frac{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}\left(\left(1-\frac{w_{ij}}{m_{ij}}\right)(1-P_{ij})+\frac{w_{ij}}{m_{ij}}P_{ij}\right)}{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}} \\
{P_{ij}}^* &= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}} \left(1-\frac{w_{ij}}{m_{ij}}\right)(1-P_{ij})+\frac{w_{ij}}{m_{ij}}P_{ij} \qquad \forall (i,j)\in\Omega, i>j \\
&= \underset{0\leqslant P_{ij}\leqslant 1}{\operatorname{argmax}} P_{ij}\left(2\frac{w_{ij}}{m_{ij}}-1\right) \qquad \forall (i,j)\in\Omega, i>j.
\end{aligned}
$$

The optimal $P$ in order to maximize the experimental total accuracy is given by: ${P_{ij}}^* = \mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right)$. By the way, the result indicates that the likelihood estimator $\frac{w_{ij}}{m_{ij}} \approx \widetilde{P_{ij}}$ is natural and even desirable. The upper bound for the experimental accuracy is now explicitly computable:

$$
\begin{aligned}
\mathcal{A}(P) &\leqslant \max_{P\in\mathcal{P}}\left[\mathcal{A}(P)\right] \\
&= \mathcal{A}\left(\mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right)\right) \\
&= \frac{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}\left(\left(1-\frac{w_{ij}}{m_{ij}}\right)\left(1-\mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right)\right)+\frac{w_{ij}}{m_{ij}}\mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right)\right)}{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}} \\
&= \sum_{\substack{(i,j)\in\Omega\\ i>j}} \frac{m_{ij}}{\sum\limits_{\substack{(i,j)\in\Omega\\ i>j}} m_{ij}}\left(\frac{1}{2}+\left|\frac{1}{2}-\frac{w_{ij}}{m_{ij}}\right|\right).
\end{aligned}
$$

A major difference in this result compared with the previous expected upper bound is that it is often true that $w_{ij}$ and $m_{ij}$ for $(i,j)\in\Omega$ are small ($\leqslant 2$) in a real dataset $\mathcal{D}$. Therefore, the ratio $\frac{w_{ij}}{m_{ij}}$ is subject to a big variance and is biased towards 0 or 1. This allows the experimental accuracy to be better than the expected one.

# Appendix B

# Rank and singular values distribution of skew-symmetric matrices

## B.1 Singular value of $\frac{1}{2}1_{n\times n}$

We can easily rewrite the matrix $\frac{1}{2}1_{n\times n}$ as a product of vectors, which means that it is of rank 1. Moreover, the vector can easily be normalized:

$$\frac{1}{2}1_{n\times n} = \frac{1}{2}1_{n\times 1}{1_{n\times 1}}^\top = \frac{n}{2}\left(\frac{1_{n\times 1}}{\sqrt{n}}\right)\left(\frac{1_{n\times 1}}{\sqrt{n}}\right)^\top.$$

This last expression represents the SVD decomposition of the matrix. It is clear that $\sigma_1 = \frac{n}{2}$ is the only non-zero singular value.

## B.2 Proof that $P'$ is skew-symmetric

Recall that by definition $P = \frac{1}{2}1_{n\times n} + P'$. It is trivial that $\frac{1}{2}1_{n\times n}$ is symmetric: $\frac{1}{2}1_{n\times n} = \frac{1}{2}{1_{n\times n}}^\top$. Then, $P' = P - \frac{1}{2}1_{n\times n}$ is skew-symmetric:

$$\begin{aligned}
P' &= P - \frac{1}{2}1_{n\times n} \\
&= (1_{n\times n} - P^\top) - \frac{1}{2}1_{n\times n} \\
&= \frac{1}{2}1_{n\times n} - P^\top \\
&= \frac{1}{2}1_{n\times n}^\top - P^\top \\
&= -\left(P - \frac{1}{2}1_{n\times n}\right)^\top
\end{aligned}$$

$$= -P'^{\top}.$$

## B.3 Eigenvalues of skew-symmetric matrix

Let matrix $A \in \mathbb{R}^{n\times n}$ be skew-symmetric: $A = -A^{\top}$. Note that this matrix is real, thus $\overline{A} = A$, where $\overline{A}$ is the complex conjugate matrix of $A$. Let $\lambda$ an eigenvalue of $A$: $Ax = \lambda x$. This implies that $x^*Ax = x^*\lambda x = \lambda x^*x = \lambda\|x\|^2$.

Moreover, we have:

$$\begin{aligned}
&(\overline{Ax}) = (\overline{\lambda x})\\
&\bar{A}\bar{x} = \bar{\lambda}\bar{x} = A\bar{x} && \text{as } \bar{A} = A \text{ A is real}\\
&\quad Ax = \lambda x \Longrightarrow x^*Ax = x^*\lambda x = \lambda x^*x = \lambda\|x\|^2\\
&\quad A\bar{x} = \bar{\lambda}\bar{x} \Longleftrightarrow (A\bar{x})^T = (\bar{\lambda}\bar{x})^T\\
&\quad \bar{x}^T A^T = \bar{}^T\bar{\lambda}^T\\
&\quad x^*A^T = x^*\bar{\lambda}\\
&\quad \Longrightarrow x^*A^T = x*\bar{\lambda}x = \bar{\lambda}x^*x = \bar{\lambda}\|x\|^2\\
&\quad x^*(-A)x = \bar{\lambda}\|x\|^2\\
&\quad -x^*Ax = \bar{\lambda}\|x\|^2\\
&\quad x^*Ax = -\bar{\lambda}\|x\|^2
\end{aligned}$$

$$\begin{aligned}
\Longrightarrow x^*Ax = -\bar{\lambda}\|x\|^2 = \lambda\|x\|^2 \forall x \in \mathbb{C}\\
\Longleftrightarrow \lambda = \bar{\lambda}\\
\lambda = a + bi\\
-\bar{\lambda} = -(a - bi) = -a + bi \; a, b \in \mathbb{R}\\
\Longleftrightarrow a + bi = -a + bi\\
\lambda = bi
\end{aligned}$$

All eigenvalues are complex $\lambda_j = bi$ or $\lambda_j = 0$. If $\lambda$ is an eigenvalue and $\bar{\lambda} = -\lambda$ (since $A = -A^T$) :

$$\begin{aligned}
&Ax = \lambda x\\
&(\overline{Ax}) = (\overline{\lambda x}) = \bar{A}\bar{x} = \bar{\lambda}\bar{x} = A\bar{x} = -\lambda\bar{x}\\
&y = \bar{x} \forall y \in \mathbb{C}\\
&Ay = -\lambda y
\end{aligned}$$

$$\implies -\lambda = \bar{\lambda} = bi \text{ is also an eigenvalue}$$

If $A = A^T = \bar{A}$ ( A real skew-symmetric), then all $\lambda$ are either $\lambda = bi$ either $\lambda = 0$. If $\lambda_j = bi$ , $b \neq 0 \implies \lambda_{j+1} = -bi = -\lambda_j = \bar{\lambda_j}$
All nonzero eigenvalues are strictly imaginary and come in conjugated pairs.
Rank of A is pair (r = 2m)

$$\lambda \in \{b_1 i, -b_1 i, ..., b_m i, -b_m i, 0, ..., 0\}$$

## B.4 Singular values of skew-symmetric matrix

$u_i v_i^T$ and $v_i u_i^T$ are linearly independent if and only if $v_i \neq k, u_i$.
$v_i \neq k, u_i$ because otherwise $A_i = u_i u_i^T = A_i^T \implies$ symmetric and not anti-symmetric.
$u_i v_i^T$ and $v_i u_i^T$ are linearly independent.
Eigenvalues of $A^T A : Ax = \lambda_j^A x \ \forall x \in \mathbb{C}$

$$\lambda_j^{A^T A} = A^T A x = A^T \lambda_j^A x = \lambda_j^A A^T x = -\lambda_j^A A x = -\lambda_j^A \lambda_j^A x$$
$$\lambda_j^{A^T A} x = -(\lambda_j^A)^2 x$$
$$\lambda_j^{A^T A} = -(\lambda_j^A)^2$$

$$\text{If } \lambda_j^A = \lambda_{2k}^A = b_k i \implies -(\lambda_j^A)^2 = -(b_k i)^2 = b_k^2 = \lambda_j^{A^T A} = \lambda_{2k}^{A^T A}$$
$$\text{If } \lambda_j^A = \lambda_{2k+1}^A = -b_k i \implies -(\lambda_j^A)^2 = -(b_k i)^2 = b_k^2 = \lambda_j^{A^T A} = \lambda_{2k+1}^{A^T A}$$

$$\sigma_j^A := \sqrt{\lambda_j^{A^T A}} = \sqrt{b_k^2} = b_k \geqslant 0$$
$$\sigma_{2k} = \sigma_{2k+1} = b_k \longrightarrow \text{Always 2 identical singular values}$$
$$\sigma \in \{b_1, b_1, ..., b_m, b_m, 0, ...., 0\}$$

$$\begin{aligned} A + A^T = 0 &= \sum_{i=1}^{n} \sigma_i u_i v_i^T + \sum_{i=1}^{n} \sigma_i v_i u_i^T = \sum_{i=1}^{n} \sigma_i (u_i v_i^T + v_i u_i^T) = 0 \\ &= \sum_{i=1}^{r} \sigma_i (u_i v_i^T + v_i u_i^T) \\ &= \sum_{k=1}^{m} \sigma_{2k-1} (u_{2k-1} v_{2k-1}^T + v_{2k-1} u_{2k-1}^T) = \sum_{k=1}^{m} \sigma_{2k} (u_{2k} v_{2k}^T + v_{2k} u_{2k}^T) \\ &= \sum_{k=1}^{m} \sigma_{2k} (u_{2k-1} v_{2k-1}^T + v_{2k-1} u_{2k-1}^T + u_{2k} v_{2k}^T + v_{2k} u_{2k}^T) \end{aligned}$$

As $u_i v_i^T$ is linearly independent of $v_i u_i^T$ and independent of $u_{i+1} v_{i+1}^T$.

$$u_i v_i^T = v_{i+1} u_{i+1} \iff v_i u_i^T = -u_{i+1} v_{i+1}^T$$
$$u_{2k-1} v_{2k-1}^T + v_{2k-1} u_{2k-1}^T + u_{2k} v_{2k}^T + v_{2k} u_{2k}^T = -v_{2k} u_{2k}^T - u_{2k} v_{2k}^T + u_{2k} v_{2k}^T + v_{2k} u_{2k}^T = 0$$

Conditions :

- $\sigma_{2k-1} = \sigma_{2k}$
- $u_{2k-1} v_{2k-1}^T = -v_{2k} u_{2k}^T$

$$\begin{aligned}
&u_{2k-1} v_{2k-1}^T = -v_{2k} u_{2k}^T \\
&u_{2k-1} v_{2k-1}^T v_{2K-1} = -v_{2k} u_{2k}^T v_{2k-1} \\
&u_{2k-1} = -v_{2k} u_{2k}^T v_{2k-1} \qquad\qquad \alpha = u_{2k}^T v_{2k-1} \\
&u_{2k-1} = -v_{2k} \alpha \\
&\|u_{2k-1}\| = 1 = \| - \alpha v_{2k}\| = |\alpha| \|v_{2k}\| = |\alpha| \\
&\implies \alpha = \pm 1 = u_{2k}^T v_{2k-1} \\
&\implies u_{2K-1} = \pm v_{2k}
\end{aligned}$$

By the same reasoning, we can deduce : $v_{2k-1} = \pm u_{2k}$.
We choose $u_{2k-1} = -v_{2k}$,$v_{2k_1} = u_{2k}$.

$$\begin{aligned}
A &= \sum_{i=1}^{r} \sigma_i u_i v_i^T = \sum_{k=1}^{m} \sigma_{2k} (u_{2k-1} v_{2k-1}^T - u_{2k} v_{2k}^T) \\
&= \sum_{k=1}^{m} \sigma_{2k} (-v_{2k} u_{2k}^T + u_{2k} v_{2k}^T) \\
&= \sum_{k=1}^{m} \sigma_{2k} (u_{2k} v_{2k}^T - v_{2k} u_{2k}^T)
\end{aligned}$$

## B.5 BMF decomposition of rank $r$ matrix

Suppose that there exist BMF of the matrix $A \in \mathbb{R}^{m \times n}$ of rank $r$, with $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$: $A = UV^\top$. It is easy to prove that $U$ and $V$ are full-rank $r$. Obviously, we have $\text{rank}(U) \leqslant r$ and $\text{rank}(V) \leqslant r$ by the size of the matrices. Then, it is well-known that a classical matrix multiplication can only reduce the rank of the matrices, i.e. we have the following inequality [Mat]:

$$r = \text{rank}(A) = \text{rank}(UV^\top) \leqslant \min\{\text{rank}(U), \text{rank}(V^\top)\} = \min\{\text{rank}(U), \text{rank}(V)\}$$

$$\iff \begin{cases} r & \leqslant \operatorname{rank}(U) \\ r & \leqslant \operatorname{rank}(V). \end{cases}$$

This implies that $r \leqslant \operatorname{rank}(U) \leqslant r$ and $r \leqslant \operatorname{rank}(V) \leqslant r$, or equivalently $\operatorname{rank}(U) = \operatorname{rank}(V) = r$.

## B.6 Non-uniqueness of BMF

Let $A = UV^\top$ be a BMF of $A$ and $\operatorname{rank}(A)$ its rank. Let now $T$ be any matrix in the set of invertible matrices of size $r$ (which is $\mathbb{R}_r^{r\times r}$), we can then find new matrices $U_T = UT$ and $V_T = VT^{-\top}$ which have the same rank $r$ and such that $U_T {V_T}^\top$ is another BMF of $A$:

$$A = UV^\top = UIV^\top = U\left(TT^{-1}\right)V^\top = UTT^{-1}V^\top = UT\left(VT^{-\top}\right)^\top = U_T {V_T}^\top.$$

## B.7 Non-uniqueness of the orthogonal BMF

Suppose that we have the following orthogonal BMF of $A \in \mathbb{R}_r^{m\times n}$, with $Q \in \mathbb{R}^{m\times r}$ and $R \in \mathbb{R}^{m\times r}$ such that $Q^\top Q = I_r$:

$$A = QR^\top.$$

Even this more constrained factorization is not unique. By choosing $T \in \mathbb{R}^{r\times r}$ orthogonal ($T^\top T = TT^\top = I_r$), we get $A = QR^\top = QTT^\top R^\top = (QT)(RT)^\top = Q_T R_T^\top$, where $Q_T$ is orthogonal again. Indeed, ${Q_T}^\top Q_T = (QT)^\top QT = T^\top Q^\top QT = T^\top I_r T = T^\top T = I_r$. A special case of this factorization is to ask $R^\top$ to be upper triangular. Then this even more specific factorization is known as the *rank reduced QR decomposition* [Hac12].

# Appendix C

# Convexity of objective functions

## C.1 Definition of convexity

Let us recall that a function $f : \mathbb{R}^n \to \mathbb{R} : x \to f(x)$ is convex if $\forall x, y \in \mathbb{R}^n, \forall t \in [0, 1]$:

$$f(tx + (1-t)y) \leqslant tf(x) + (1-t)f(y).$$

If the function is twice differentiable, then an equivalent condition is that its hessian is positive semi-definite:

$$\nabla^2 f(x) \succcurlyeq 0.$$

If the function is univariate ($n = 1$), then it is enough to ask that its second derivative is positive: $\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(f(x)\right) \geqslant 0$.

## C.2 Convexity of negative log-likelihood

Recall that $\mathcal{L}(P_{ij}|\mathcal{D}_{ij}) = P_{ij}^{w_{ij}}\left(1 - P_{ij}\right)^{m_{ij}-w_{ij}}$, with $0 \leqslant P_{ij} \leqslant 1$ and $0 \leqslant w_{ij} \leqslant m_{ij}$. It is easy to prove the convexity of $-\ln\left(\mathcal{L}(P_{ij}|\mathcal{D}_{ij})\right) = -w_{ij}\ln\left(P_{ij}\right) - \left(m_{ij} - w_{ij}\right)\ln\left(1 - P_{ij}\right)$ by proving that its second derivative is always positive:

$$\begin{aligned}
\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(-\ln\left(\mathcal{L}(x|\mathcal{D}_{ij})\right)\right) &= \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(-w_{ij}\ln(x) - \left(m_{ij} - w_{ij}\right)\ln(1-x)\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left(-w_{ij}\frac{1}{x} - \left(m_{ij} - w_{ij}\right)\frac{-1}{1-x}\right) \\
&= \frac{w_{ij}}{x^2} + \frac{m_{ij} - w_{ij}}{(1-x)^2} \\
&\geqslant 0. \qquad \text{(C.1)}
\end{aligned}$$

## C.3 Convexity of $\ln\left(1+e^{-c^\top x}\right)$

Let's prove that $\ln\left(1+e^{-c^\top x}\right)$ is a convex function where $x \in \mathbb{R}^n$ and $c \in \mathbb{R}^n$, by showing its hessian is positive semi-definite. The hessian is given by:

$$\begin{aligned}
\frac{\partial}{\partial x_i}\left(\ln\left(1+e^{-c^\top x}\right)\right) &= -c_i \frac{e^{-c^\top x}}{1+e^{-c^\top x}} \\
&= -c_i \frac{1}{1+e^{c^\top x}} \\
\frac{\partial}{\partial x_l}\left(\frac{\partial}{\partial x_i}\left(\ln\left(1+e^{-c^\top x}\right)\right)\right) &= \frac{\partial}{\partial x_l}\left(-c_i \frac{1}{1+e^{c^\top x}}\right) \\
&= \frac{e^{c^\top x}}{\left(1+e^{c^\top x}\right)^2} c_i c_l \\
\nabla^2\left(\ln\left(1+e^{-c^\top x}\right)\right) &= \frac{e^{c^\top x}}{\left(1+e^{c^\top x}\right)^2} cc^\top
\end{aligned}$$

Its hessian is indeed positive semi-definite:

$$\begin{aligned}
x^\top \nabla^2\left(\ln\left(1+e^{-c^\top x}\right)\right) x &= x^\top \left(\frac{e^{c^\top x}}{(1+e^{c^\top x})^2} cc^\top\right) x \\
&= \frac{e^{c^\top x}}{(1+e^{c^\top x})^2} x^\top cc^\top x \\
&= \frac{e^{c^\top x}}{(1+e^{c^\top x})^2} (c^\top x)^2 \\
&\geqslant 0 \\
\nabla^2\left(\ln(1+e^{-c^\top x})\right) &\succcurlyeq 0.
\end{aligned}$$

Moreover, for some $c$ this function is strictly convex.

## C.4 Non-convexity of $\ln\left(1+e^{-xy}\right)$

However this function $\ln\left(1+e^{-xy}\right)$ is not convex as it contains the product of two variables $x$ and $y$. Here is a counterexample proving that this function is not convex:

$$\nabla^2\left(\ln(1+e^{-xy})\right) = \frac{e^{xy}}{(1+e^{xy})^2} \begin{pmatrix} y^2 & xy - 1 - \frac{1}{e^{xy}} \\ xy - 1 - \frac{1}{e^{xy}} & x^2 \end{pmatrix}.$$

When $x = y = \frac{1}{2}$, the hessian is not semi-positive definite : the smallest eigenvalue is negative ($\lambda_2 \approx -0.314$).

## C.5 Maximum of symmetric prior distribution with convex negative log-prior

Let $f_{\widetilde{P_{ij}}}(x)$ for $0 \leqslant x \leqslant 1$ be the prior distribution of $\widetilde{P_{ij}}$. $f_{\widetilde{P_{ij}}}(x)$ is symmetric around $\frac{1}{2}$ ($f_{\widetilde{P_{ij}}}(x) = f_{\widetilde{P_{ij}}}(1-x)$) and $-\ln(f_{\widetilde{P_{ij}}}(x))$ is convex.

As $f_{\widetilde{P_{ij}}}(x)$ is symmetric around $\frac{1}{2}$, $-\ln(f_{\widetilde{P_{ij}}}(x))$ is also symmetric around $\frac{1}{2}$: $-\ln(f_{\widetilde{P_{ij}}}(x)) = -\ln(f_{\widetilde{P_{ij}}}(1-x))$. Let us use the notation $l(x) = -\ln(f_{\widetilde{P_{ij}}}(x))$ to simplify the rest of the argument. Associated with its convexity, the symmetry of $l(x)$ implies that it has its minimum at $x^* = \frac{1}{2}$. Indeed, let us define a point $x \in [0, \frac{1}{2}]$. Then, we can prove that for any point $z \in [x, 1-x]$ (where $z$ can be parameterized as follows: $z = tx + (1-t)(1-x)$, for any $t \in [0,1]$) we have $l(z) \leqslant l(x)$:

$$\begin{aligned} l(z) = l(tx + (1-t)(1-x)) &\leqslant tl(x) + (1-t)l(1-x) \\ &= tl(x) + (1-t)l(x) \\ &= l(x). \end{aligned}$$

This inequality is true for all $x \in [0, \frac{1}{2}]$. Notice that $z = \frac{1}{2}$ can always be chosen, for any value of $x$ (when $t = \frac{1}{2}$). Therefore, $l\left(\frac{1}{2}\right) \leqslant l(x), \forall x \in [0, \frac{1}{2}]$. Moreover, by symmetry, $l\left(\frac{1}{2}\right) \leqslant l(x) = l(1-x) = l_2(y), \forall (1-x) = y \in [\frac{1}{2}, 1]$. Summing up, we get $l\left(\frac{1}{2}\right) \leqslant l(x), \forall x \in [0,1]$. In other words, $x^* = \frac{1}{2}$ is the minimum of the function on its domain.

# Appendix D

# Estimators for $\widetilde{P_{ij}}$

This appendix contains boring proofs and tedious mathematical developments of Section 2.5.

## D.1 Equivalence between maximizing $\mathbb{P}\left[\mathcal{D}_{ij}|\widetilde{P_{ij}}\right]$ and maximizing $\mathbb{P}\left[w_{ij}|\widetilde{P_{ij}}\right]$

Let us first show that $W_{ij}$ follows a binomial distribution. Similar to what was done for $D_{ij,k}$ and $d_{ij,k}$ in Section 1.3, we can consider the random variable $W_{ij}$ and $w_{ij}$ as its realization. As all $D_{ij,k}$ are identically distributed Bernouilli and independent random variables from assumption 2: $D_{ij,1}, \cdots, D_{ij,m_{ij}} \overset{\text{i.i.d.}}{\sim} \text{Ber}(\widetilde{P_{ij}})$. Therefore, $W_{ij}$ follows a binomial distribution:

$$W_{ij} = \sum_{k=1}^{m_{ij}} D_{ij,k} \sim \text{Bin}(m_{ij}, \widetilde{P_{ij}}).$$

Here is a reminder of the probability mass function of a binomial random variable:

$$\mathbb{P}[W_{ij} = w_{ij}] = \binom{m_{ij}}{w_{ij}} \widetilde{P_{ij}}^{w_{ij}} \left(1 - \widetilde{P_{ij}}\right)^{m_{ij} - w_{ij}} \qquad 0 \leqslant w_{ij} \leqslant m_{ij}. \tag{D.1}$$

Let us now prove the equivalence we are interested in. We could want to maximize the probability of the total number of wins of $i$ over $j$ $\mathbb{P}\left[w_{ij}|\widetilde{P_{ij}}\right]$, instead of each game independently $\mathbb{P}\left[\mathcal{D}_{ij}|\widetilde{P_{ij}}\right]$ (which corresponds to the likelihood function 2.22). However, this would lead to the exact same formulation. Indeed, the consideration that the order of the won games does not matter is encoded into the combinatorial coefficient $-\ln\left(\binom{m_{ij}}{w_{ij}}\right)$

disappearing in the final objective function as it is constant:

$$\begin{aligned}
-\ln\left(\mathbb{P}\left[w_{ij}|\widetilde{P_{ij}}\right]\right) &= -\ln\left(\binom{m_{ij}}{w_{ij}}\widetilde{P_{ij}}^{w_{ij}}\left(1-\widetilde{P_{ij}}\right)^{m_{ij}-w_{ij}}\right)\\
&= -\ln\left(\binom{m_{ij}}{w_{ij}}\mathbb{P}\left[\mathcal{D}_{ij}|\widetilde{P_{ij}}\right]\right)\\
&= -\ln\left(\binom{m_{ij}}{w_{ij}}\right) - \ln\left(\mathbb{P}\left[\mathcal{D}_{ij}|\widetilde{P_{ij}}\right]\right).
\end{aligned}$$

## D.2 ML estimator

The ML estimator $\overline{P_{ij}}^{\mathrm{ML}} = \frac{w_{ij}}{m_{ij}}$ is really easy to compute thanks to the negative log-tick:

$$\begin{aligned}
\overline{P_{ij}}^{\mathrm{ML}} &= \operatorname*{argmax}_{0\leqslant\widetilde{P_{ij}}\leqslant 1} \mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij})\\
&= \operatorname*{argmax}_{0\leqslant\widetilde{P_{ij}}\leqslant 1} \ln\left(\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij})\right)\\
&= \operatorname*{argmin}_{0\leqslant\widetilde{P_{ij}}\leqslant 1} -\ln\left(\mathcal{L}(\widetilde{P_{ij}}|\mathcal{D}_{ij})\right)\\
&= \operatorname*{argmin}_{0\leqslant\widetilde{P_{ij}}\leqslant 1} -\ln\left(\widetilde{P_{ij}}^{w_{ij}}\left(1-\widetilde{P_{ij}}\right)^{m_{ij}-w_{ij}}\right)\\
&= \operatorname*{argmin}_{0\leqslant\widetilde{P_{ij}}\leqslant 1} -w_{ij}\ln\left(\widetilde{P_{ij}}\right) - (m_{ij}-w_{ij})\ln\left(1-\widetilde{P_{ij}}\right)\\
0 &= \nabla\left(-w_{ij}\ln\left(\widetilde{P_{ij}}\right) - (m_{ij}-w_{ij})\ln\left(1-\widetilde{P_{ij}}\right)\right)\\
0 &= -w_{ij}\frac{1}{\widetilde{P_{ij}}} - (m_{ij}-w_{ij})(-1)\frac{1}{1-\widetilde{P_{ij}}}\\
w_{ij}(1-\widetilde{P_{ij}}) &= (m_{ij}-w_{ij})\widetilde{P_{ij}}\\
\overline{P_{ij}}^{\mathrm{ML}} &= \frac{w_{ij}}{m_{ij}}.
\end{aligned}$$

## D.3 CM estimator for uniform prior

It is known from Appendix E.1 that $\widetilde{P_{ij}}$ follows a Beta posterior distribution when the prior is uniform: $\widetilde{P_{ij}}|\mathcal{D}_{ij} \sim \mathrm{B}(w_{ij}+1, m_{ij}-w_{ij}+1)$. It is then easy to show that $\overline{P_{ij}}^{\mathrm{CM}} = \frac{w_{ij}+1}{m_{ij}+2}$:

$$\begin{aligned}
\overline{P_{ij}}^{\mathrm{CM}} &= \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right]\\
&= \int_0^1 \widetilde{p_{ij}}.(m_{ij}+1)\binom{m_{ij}}{w_{ij}}\widetilde{p_{ij}}^{w_{ij}}\left(1-\widetilde{p_{ij}}\right)^{m_{ij}-w_{ij}} d\widetilde{p_{ij}}
\end{aligned}$$

$$
\begin{aligned}
&= (m_{ij}+1)\binom{m_{ij}}{w_{ij}} \int_0^1 \widetilde{p_{ij}}^{w_{ij}+1} \left(1-\widetilde{p_{ij}}\right)^{m_{ij}-w_{ij}} d\widetilde{p_{ij}} \\
&= (m_{ij}+1)\binom{m_{ij}}{w_{ij}} .\mathrm{B}(w_{ij}+2, m_{ij}-w_{ij}+1) \\
&= (m_{ij}+1)\frac{m_{ij}!}{w_{ij}!(m_{ij}-w_{ij})!}\frac{(w_{ij}+1)!(m_{ij}-w_{ij})!}{(m_{ij}+2)!} \\
&= \frac{w_{ij}+1}{m_{ij}+2}.
\end{aligned}
$$

## D.4 CM estimator is MMSE estimator

It is easy to prove mathematically that $\overline{P_{ij}}^{\mathrm{MMSE}} = \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] = \overline{P_{ij}}^{\mathrm{CM}}$ from the theoretical definition $\mathrm{MSE} = \mathbb{E}\left[\left(\widetilde{P_{ij}} - \overline{P_{ij}}\right)^2 \middle| \mathcal{D}_{ij}\right]$, where the expectation is made over $\widetilde{P_{ij}}$ and the optimization variable is $\overline{P_{ij}}$:

$$
\begin{aligned}
\overline{P_{ij}}^{\mathrm{MMSE}} &= \underset{0\leqslant\overline{P_{ij}}\leqslant 1}{\operatorname{argmax}} \mathbb{E}\left[\left(\widetilde{P_{ij}} - \overline{P_{ij}}\right)^2 \middle| \mathcal{D}_{ij}\right] \\
&= \underset{0\leqslant\overline{P_{ij}}\leqslant 1}{\operatorname{argmax}} \mathbb{E}\left[\widetilde{P_{ij}}^2|\mathcal{D}_{ij}\right] + \mathbb{E}\left[\overline{P_{ij}}^2|\mathcal{D}_{ij}\right] - 2\mathbb{E}\left[\widetilde{P_{ij}}\overline{P_{ij}}|\mathcal{D}_{ij}\right] \\
&= \underset{0\leqslant\overline{P_{ij}}\leqslant 1}{\operatorname{argmax}} \mathbb{E}\left[\widetilde{P_{ij}}^2|\mathcal{D}_{ij}\right] + \overline{P_{ij}}^2 - 2\overline{P_{ij}}\mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] \\
0 &= \nabla\left(\mathbb{E}\left[\widetilde{P_{ij}}^2|\mathcal{D}_{ij}\right] + \overline{P_{ij}}^2 - 2\overline{P_{ij}}\mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right]\right) \\
0 &= 2\overline{P_{ij}} - 2\mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] \\
\overline{P_{ij}}^{\mathrm{MMSE}} &= \mathbb{E}\left[\widetilde{P_{ij}}|\mathcal{D}_{ij}\right] = \overline{P_{ij}}^{\mathrm{CM}}.
\end{aligned}
$$

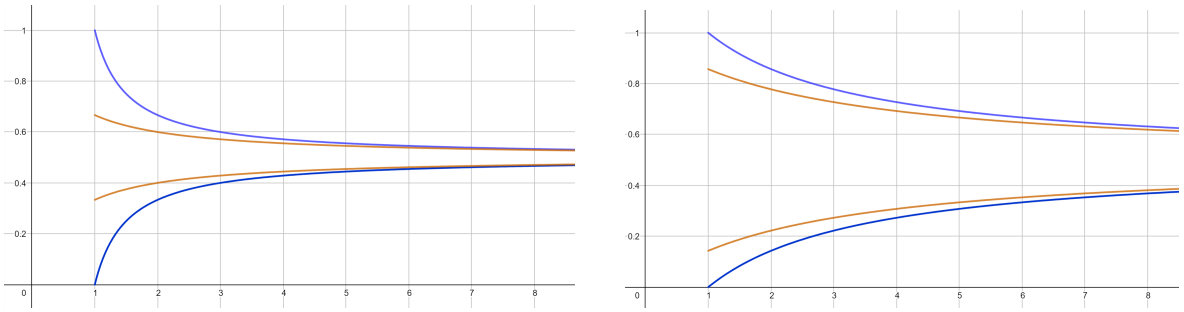
## D.5 MAP and CM estimators of beta prior

Assuming a Beta prior distribution of parameter $b$ for $\widetilde{P_{ij}}$, then the posterior distribution of $\widetilde{P_{ij}}$ follows a Beta distribution $\mathrm{Beta}(w_{ij}+b, m_{ij}-w_{ij}+b)$, as shown in Appendix E.2. Therefore, both MAP and CM estimators are easy to compute. They simply correspond to the mode and the mean of this distribution, which can be found in [Bet22]:

$$
\begin{aligned}
\overline{P_{ij}}^{\mathrm{MAP}} &= \frac{w_{ij}+b-1}{m_{ij}+2b-2} \\
\overline{P_{ij}}^{\mathrm{CM}} &= \frac{w_{ij}+b}{m_{ij}+2b}.
\end{aligned}
$$

A symmetric Beta prior distribution has the effect of pushing $\widetilde{P_{ij}}$ towards $\frac{1}{2}$. To quantify this phenomenon, it is easy to compute the bounds that each estimation method sets on the estimators and plot them:

$$\frac{b-1}{m_{ij}+2b-2} \leqslant \overline{P_{ij}}^{\text{MAP}} \leqslant \frac{m_{ij}+b-1}{m_{ij}+2b-2}$$
$$\frac{b}{m_{ij}+2b} \leqslant \overline{P_{ij}}^{\text{CM}} \leqslant \frac{m_{ij}+b}{m_{ij}+2b}.$$



(a) $m_{ij} = 1$ (b) $m_{ij} = 5$

Figure D.1: Upper and lower bounds for the estimation of $P_{ij}$ in function of $b$. The blue line corresponds to the MAP estimator and the orange line to the CM estimator.

## D.6 Proof of the irrelevance of symmetric prior for MAP estimation

In this section, we will proof the following statement, implying that using symmetric negative log-convex prior distribution does not improve the prediction accuracy:

$$\mathcal{C}\left(\overline{P_{ij}}^{\text{MAP}}\right) = \mathcal{C}\left(\overline{P_{ij}}^{\text{ML}}\right) = \begin{cases} 0 & \text{if} \quad 0 \leqslant \frac{w_{ij}}{m_{ij}} < \frac{1}{2} \\ \frac{1}{2} & \text{if} \quad \frac{w_{ij}}{m_{ij}} = \frac{1}{2} \\ 1 & \text{if} \quad \frac{1}{2} < \frac{w_{ij}}{m_{ij}} \leqslant 1. \end{cases}$$

Let $f_{\widetilde{P_{ij}}}(x)$ for $0 \leqslant x \leqslant 1$ be the prior distribution of $\widetilde{P_{ij}}$. $f_{\widetilde{P_{ij}}}(x)$ is symmetric around $\frac{1}{2}$ ($f_{\widetilde{P_{ij}}}(x) = f_{\widetilde{P_{ij}}}(1-x)$) and $-\ln(f_{\widetilde{P_{ij}}}(x))$ is convex.

The MAP estimation problem consists in solving problem 2.24. The objective function is made of two convex terms. First, the negative log-likelihood term: $l_1(x) = -\ln\left(\mathcal{L}(x|\mathcal{D}_{ij})\right) = -w_{ij}\ln(x) - (m_{ij}-w_{ij})\ln(1-x)$. The proof of convexity is given in Appendix C.2. The minimum of $l_1(x)$ is at $x_1 = \frac{w_{ij}}{m_{ij}} = \overline{P_{ij}}^{\text{ML}}$ (see Appendix D.2). The second term is the

negative log-prior $-\ln(f_{\widetilde{P_{ij}}}(x)) = l_2(x)$, convex by hypothesis. It can be proven that it has a minimum at $x_2 = \frac{1}{2}$ (see Appendix C.5).

The objective function of problem 2.24, denoted $l_3(x) = l_1(x) + l_2(x)$, is the sum of two convex functions and is therefore convex. We define $x_m = \min\{x_1, x_2\}$ and $x_M = \max\{x_1, x_2\}$. We will now prove that its minimum $x_3 = \overline{P_{ij}}^{\text{MAP}}$ lies on the interval $[x_m, x_M]$. For this purpose, we will show that for any $x_l \in [0, x_m]$, we have $l_3(x_m) \leqslant l_3(x_l)$ and for any $x_r \in [x_M, 1]$, we have $l_3(x_M) \leqslant l_3(x_r)$.

First, let us prove that $l_1(x_m) \leqslant l_1(x_l)$. There are two cases. If $x_m = x_1 \leqslant x_2$, then the inequality $l_1(x_1) \leqslant l_1(x_l)$ is trivial as $x_1$ is the minimizer of $l_1(x)$ and is thus always true. If $x_m = x_2 \leqslant x_1$, then we have to use the convexity of $l_1(x)$. Since $x_l \leqslant x_2 \leqslant x_1$, there exists a $t_2 \in [0, 1]$ such that we can parameterize $x_2 = t_2 x_l + (1 - t_2)x_1$. Therefore, we can use the definition of convexity:

$$\begin{aligned} l_1(x_2) = l_1(t_2 x_l + (1 - t_2)x_1) &\leqslant t_2 l_1(x_l) + (1 - t_2) l_1(x_1) \\ &\leqslant t_2 l_1(x_l) + (1 - t_2) l_1(x_l) \\ &= l_1(x_l). \end{aligned}$$

So, we have proved that $l_1(x_m) \leqslant l_1(x_l)$. The exact same reasoning can be applied to show that $l_1(x_M) \leqslant l_1(x_r)$. Moreover, the developments can be adapted for $l_2$, and thus: $l_2(x_m) \leqslant l_2(x_l)$ and $l_2(x_M) \leqslant l_2(x_r)$. Associating those four inequations two by two, we get the desired inequalities:

$$\begin{aligned} l_3(x_m) = l_1(x_m) + l_2(x_m) &\leqslant l_1(x_l) + l_2(x_l) = l_3(x_l) \\ l_3(x_M) = l_1(x_M) + l_2(x_M) &\leqslant l_1(x_r) + l_2(x_r) = l_3(x_r). \end{aligned}$$

It is now a fact that $x_3 \in [\min\{x_1, x_2\}, \max\{x_1, x_2\}]$. Applying this result to the MAP problem gives $x_3 = \overline{P_{ij}}^{\text{MAP}} \in \left[\min\left\{\frac{w_{ij}}{m_{ij}}, \frac{1}{2}\right\}, \max\left\{\frac{w_{ij}}{m_{ij}}, \frac{1}{2}\right\}\right]$. Finally, after applying the clipping operator, we get the exact same expression for the MAP estimator as for the ML one, which concludes the proof.

## D.7 Proof of the irrelevance of symmetric Beta prior for CM estimation

If $\widetilde{P_{ij}} \sim \text{Beta}(b, b), b \geqslant 1$, the statement is almost trivial:

$$\mathcal{C}\left(\overline{P_{ij}}^{\text{CM}}\right) = \mathcal{C}\left(\frac{w_{ij} + b}{m_{ij} + 2b}\right)$$

$$= \mathcal{C}\left(\frac{w_{ij}}{m_{ij}}\right)$$
$$= \mathcal{C}\left(\overline{P_{ij}}^{\mathrm{ML}}\right).$$

The proof in the general case is still lacking.

# Appendix E

# Derivation of prior distributions

The relations between prior distributions seen in Table 4.1 are derived in this section.

## E.1 Uniform prior on ${P_{ij}}^c$

If the prior of ${P_{ij}}^c$ is considered uniform $f_{{P_{ij}}^c}(x) = 1, 0 \leqslant x \leqslant 1$, then its posterior distribution can be computed explicitly. The random variable ${P_{ij}}^c$ with a uniform prior follows a Beta distribution, i.e. ${P_{ij}}^c|{\mathcal{D}_{ij}}^c \sim \mathrm{B}({w_{ij}}^c + 1, {m_{ij}}^c - {w_{ij}}^c + 1)$:

$$
\begin{aligned}
f_{{P_{ij}}^c|{\mathcal{D}_{ij}}^c}(x) &= \frac{\mathbb{P}[{\mathcal{D}_{ij}}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)}{\mathbb{P}[{\mathcal{D}_{ij}}^c]} \\
&= \frac{\mathbb{P}[{\mathcal{D}_{ij}}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)}{\displaystyle\int_0^1 \mathbb{P}[{\mathcal{D}_{ij}}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)dx} \\
&= \frac{\mathbb{P}[{\mathcal{D}_{ij}}^c|{P_{ij}}^c].1}{\displaystyle\int_0^1 \mathbb{P}[{\mathcal{D}_{ij}}^c|{P_{ij}}^c].1dx} \\
&= \frac{x^{{w_{ij}}^c}\,(1-x)^{{m_{ij}}^c - {w_{ij}}^c}}{\displaystyle\int_0^1 x^{{w_{ij}}^c}\,(1-x)^{{m_{ij}}^c - {w_{ij}}^c}\,dx} \\
&= \frac{x^{{w_{ij}}^c}\,(1-x)^{{m_{ij}}^c - {w_{ij}}^c}}{\mathrm{B}({w_{ij}}^c + 1, {m_{ij}}^c - {w_{ij}}^c + 1)} \\
&= ({m_{ij}}^c + 1)\binom{{m_{ij}}^c}{{w_{ij}}^c} x^{{w_{ij}}^c}\,(1-x)^{{m_{ij}}^c - {w_{ij}}^c}\,.
\end{aligned}
$$

Uniform prior probability density function (PDF) on ${P_{ij}}^c$ means that its cumulative density function (CDF) is as follows for $0 \leqslant x \leqslant 1$: $F_{{P_{ij}}^c}(x) = \mathbb{P}[{P_{ij}}^c \leqslant x] = x$. In order

to understand how $R_{ij}^c$ and ${\Delta E_{ij}}^c$ are distributed, we need to compute their CDF and then their PDF by deriving it:

$$\begin{aligned}
F_{{R_{ij}}^c}(x) &= \mathbb{P}[{R_{ij}}^c \leqslant x] && x \geqslant 0 \\
&= \mathbb{P}\left[\frac{{P_{ij}}^c}{1-{P_{ij}}^c} \leqslant x\right] \\
&= \mathbb{P}\left[{P_{ij}}^c \leqslant \frac{x}{1+x}\right] \\
&= \frac{x}{1+x} && x \geqslant 0 \\
f_{{R_{ij}}^c}(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(F_{{R_{ij}}^c}(x)\right) \\
&= \frac{1}{(1+x)^2} && x \geqslant 0.
\end{aligned}$$

${R_{ij}}^c$ can be described as following a Lomax distribution of shape parameter $\alpha = 1$ and scale parameter $\lambda = 1$: ${R_{ij}}^c \sim \mathrm{Lomax}(1,1)$ [Lom21]. Moreover, the tensor of difference of ratings ${\Delta E_{ij}}^c$ follows a logistic distribution: ${\Delta E_{ij}}^c \sim \mathrm{Log}\left(0, \frac{1}{\lambda}\right)$:

$$\begin{aligned}
F_{{\Delta E_{ij}}^c}(x) &= \mathbb{P}[{\Delta E_{ij}}^c \leqslant x] \\
&= \mathbb{P}\left[\frac{1}{\lambda}\ln({R_{ij}}^c) \leqslant x\right] \\
&= \mathbb{P}\left[{R_{ij}}^c \leqslant e^{\lambda x}\right] \\
&= \frac{e^{\lambda x}}{1+e^{\lambda x}} \\
&= \frac{1}{1+e^{-\lambda x}} \\
f_{{\Delta E_{ij}}^c}(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(F_{{\Delta E_{ij}}^c}(x)\right) \\
&= \frac{\lambda e^{-\lambda x}}{\left(1+e^{-\lambda x}\right)^2}.
\end{aligned}$$

Can we know the prior distribution of ${E_i}^c$ and ${E_j}^c$ knowing the distribution of ${\Delta E_{ij}}^c$? Even if we consider them to be independent and identically distributed, this problem is underdetermined up to a translation. An equivalent condition to ${\Delta E_{ij}}^c = {E_i}^c - {E_j}^c$ tells that the product of the moment generative functions (MGF) of those random variables evaluated at $t$ and $-t$ respectively is equal to the MGF of ${\Delta E_{ij}}^c$:

$$M_{{\Delta E_{ij}}^c}(t) = M_{{E_i}^c}(t) M_{{E_j}^c}(-t).$$

The latter is well-known [Log22a] and implies the beta function $\mathrm{B}(x,y)$, which can be rewritten in terms of the famous gamma function $\Gamma$ by the identity $\mathrm{B}(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$:

$$M_{{\Delta E_{ij}}^c}(t) = \mathrm{B}\left(1 - \frac{t}{\lambda}, 1 + \frac{t}{\lambda}\right)$$

$$
\begin{aligned}
&= \frac{\Gamma\left(1-\frac{t}{\lambda}\right)\Gamma\left(1+\frac{t}{\lambda}\right)}{\Gamma\left(1-\frac{t}{\lambda}+1+\frac{t}{\lambda}\right)} \\
&= \frac{\Gamma\left(1-\frac{t}{\lambda}\right)\Gamma\left(1+\frac{t}{\lambda}\right)}{\Gamma(2)} \\
&= \frac{\Gamma\left(1-\frac{t}{\lambda}\right)\Gamma\left(1+\frac{t}{\lambda}\right)}{1} \\
&= \Gamma\left(1-\frac{t}{\lambda}\right)\Gamma\left(1+\frac{t}{\lambda}\right) \\
&= e^{\mu t}\Gamma\left(1-\frac{t}{\lambda}\right)e^{-\mu t}\Gamma\left(1+\frac{t}{\lambda}\right) \\
&= e^{\mu t}\Gamma\left(1-\frac{1}{\lambda}t\right)e^{\mu(-t)}\Gamma\left(1-\frac{1}{\lambda}(-t)\right).
\end{aligned}
$$

Gumbel distributions with parameters $\mu$ and $\frac{1}{\lambda}$: ${E_i}^c, {E_j}^c \sim \text{Gumbel}\left(\mu, \frac{1}{\lambda}\right)$ fulfill the condition [Gum22]. Indeed,

$$
\begin{aligned}
M_{{E_i}^c}(t) &= e^{\mu t}\Gamma\left(1-\frac{1}{\lambda}t\right) \\
f_{{E_i}^c}(x) &= \lambda e^{-\left(\lambda(x-\mu)+e^{-\lambda(x-\mu)}\right)} \\
F_{{E_i}^c}(x) &= e^{-e^{-\lambda(x-\mu)}}.
\end{aligned}
$$

We notice that $\mu$ can be chosen freely, confirming that the problem is underdetermined to a translation ! Even if these distributions can be made zero-mean by choosing $\mu = -\frac{\gamma}{\lambda}$, where $\gamma \approx 0.57721$ is the Euler-Mascheroni constant: ${E_i}^c, {E_j}^c \sim \text{Gumbel}\left(-\frac{\gamma}{\lambda}, \frac{1}{\lambda}\right)$. Disappointingly, the Gumbel distribution is never symmetric.

The distribution of the positive rating ${S_i}^c$ can be obtained from the one of ${E_i}^c$ and is known as the Inverse Exponential distribution: ${S_i}^c \sim \text{InvExp}(e^{-\lambda\mu})$ [Inv]. If $\mu = -\frac{\gamma}{\lambda}$, then the dependance on $\lambda$ disappears: ${S_i}^c \sim \text{InvExp}(e^{\gamma} \approx 1.78107)$:

$$
\begin{aligned}
F_{{S_i}^c}(x) &= \mathbb{P}[{S_i}^c \leqslant x] && x \geqslant 0 \\
&= \mathbb{P}\left[e^{\lambda {E_i}^c} \leqslant x\right] \\
&= \mathbb{P}\left[{E_i}^c \leqslant \frac{1}{\lambda}\ln(x)\right] \\
&= e^{-e^{-\lambda\left(\left(\frac{1}{\lambda}\ln(x)\right)-\mu\right)}} \\
&= e^{-e^{-(\ln(x)-\lambda\mu)}} \\
&= e^{-e^{\ln\left(\frac{1}{x}\right)}e^{\lambda\mu}} \\
&= e^{-\frac{e^{-\lambda\mu}}{x}} && x \geqslant 0 \\
f_{{S_i}^c}(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(F_{{S_i}^c}(x)\right)
\end{aligned}
$$

$$= \frac{e^{-\lambda\mu} e^{-\frac{e^{-\lambda\mu}}{x}}}{x^2} \qquad\qquad x \geqslant 0.$$

## E.2 Beta prior on ${P_{ij}}^c$

The prior distribution of ${P_{ij}}^c$ is assumed to be a symmetric Beta distribution of parameter $b \geqslant 1$: ${P_{ij}}^c \sim \text{Beta}(b, b)$. So for $0 \leqslant x \leqslant 1$:

$$f_{{P_{ij}}^c}(x) = \frac{x^{b-1}(1-x)^{b-1}}{\text{B}(b,b)}.$$

Then the posterior distribution of ${P_{ij}}^c$ is then also a Beta distribution, but with different parameters: ${P_{ij}}^c|\mathcal{D}_{ij}{}^c \sim \text{Beta}({w_{ij}}^c + b, {m_{ij}}^c - {w_{ij}}^c + b)$:

$$\begin{aligned}
f_{{P_{ij}}^c|\mathcal{D}_{ij}{}^c}(x) &= \frac{\mathbb{P}[\mathcal{D}_{ij}{}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)}{\mathbb{P}[\mathcal{D}_{ij}{}^c]} \\
&= \frac{\mathbb{P}[\mathcal{D}_{ij}{}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)}{\displaystyle\int_0^1 \mathbb{P}[\mathcal{D}_{ij}{}^c|{P_{ij}}^c].f_{{P_{ij}}^c}(x)dx} \\
&= \frac{x^{{w_{ij}}^c}(1-x)^{{m_{ij}}^c - {w_{ij}}^c}.\frac{x^{b-1}(1-x)^{b-1}}{\text{B}(b,b)}}{\displaystyle\int_0^1 x^{{w_{ij}}^c}(1-x)^{{m_{ij}}^c - {w_{ij}}^c}.\frac{x^{b-1}(1-x)^{b-1}}{\text{B}(b,b)}dx} \\
&= \frac{x^{{w_{ij}}^c + b - 1}(1-x)^{{m_{ij}}^c - {w_{ij}}^c + b - 1}}{\displaystyle\int_0^1 x^{{w_{ij}}^c + b - 1}(1-x)^{{m_{ij}}^c - {w_{ij}}^c + b - 1}\,dx} \\
&= \frac{x^{{w_{ij}}^c + b - 1}(1-x)^{{m_{ij}}^c - {w_{ij}}^c + b - 1}}{\text{B}({w_{ij}}^c + b, {m_{ij}}^c - {w_{ij}}^c + b)}.
\end{aligned}$$

## E.3 Logistic prior on ${E_i}^c$

When assuming ${E_i}^c \sim \text{Log}(0, \text{s})$, then we can prove that $P_{ij}$ follows a really complicated distribution which looks like a Beta distribution, therefore we will call it a Beta-like distribution depending on $\alpha = \frac{1}{s\lambda}$: $P_{ij} \sim \text{BetaLike}(\alpha)$,

$$f_{P_{ij}}(x) = \frac{\alpha\left(\frac{x}{1-x}\right)^\alpha\left(-2\left(\frac{x}{1-x}\right)^\alpha + \alpha\left(\frac{x}{1-x}\right)^\alpha \ln\left(\frac{x}{1-x}\right) + \alpha\ln\left(\frac{x}{1-x}\right) + 2\right)}{(1-x)x\left(\left(\frac{x}{1-x}\right)^\alpha - 1\right)^3}.$$

Visually, we can find that this distribution is negative log-convex (and therefore acceptable) only for approximately $\alpha \geqslant 1.55$ (the exact value of $\alpha$ has not been calculated).

Empirically, we can find that BetaLike($\alpha$) $\approx$ Beta($b, b$) when we have the relation $b \approx 0.35\alpha^2 + 0.25$. This holds for the whole acceptable range of values of $\alpha \geqslant 1.55$ (or $b \geqslant 1.1$, really close to the true convexity condition on $b$: $b \geqslant 1$).

Let $E_i, E_j \sim \text{Log}(\mu, s)$:

$$f_{E_i}(x) = \frac{e^{-\frac{x-\mu}{s}}}{s(1+e^{-\frac{x-\mu}{s}})^2}$$
$$F_{E_i}(x) = \frac{1}{1+e^{-\frac{x-\mu}{s}}}.$$

What distribution does $\Delta E_{i,j} = E_i - E_j$ follow? Let us compute its CDF:

$$\begin{aligned}\mathbb{P}\left[\Delta E_{i,j} \leqslant z\right] &= \mathbb{P}\left[E_i - E_j \leqslant z\right] \\ &= \mathbb{P}\left[E_j \geqslant E_i - z\right].\end{aligned}$$

As $E_i$ and $E_j$ are i.i.d., we have:

$$\begin{aligned}f_{E_i,E_j}(e_i, e_j) &= f_{E_i}(e_i).f_{E_j}(e_j) \\ &= \frac{e^{-\frac{e_i+\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2}.\frac{e^{-\frac{e_j+\mu}{s}}}{s(1+e^{-\frac{e_j-\mu}{s}})^2}.\end{aligned}$$

Let $R$ represent the half-plane represented by the inequality $e_j \geqslant e_i - z$:

$$\begin{aligned}F_{\Delta E_{i,j}}(z) &= \mathbb{P}\left[\Delta E_{i,j} \leqslant z\right] \\ &= \mathbb{P}\left[E_j \geqslant E_i - z\right] \\ &= \iint_R f_{E_i,E_j}(e_i, e_j) de_i de_j \\ &= \int_{e_i=-\infty}^{e_i=\infty} \int_{e_j=e_i-z}^{e_j=\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2} \frac{e^{-\frac{e_j-\mu}{s}}}{s(1+e^{-\frac{e_j-\mu}{s}})^2} de_i de_j \\ &= \int_{e_i=-\infty}^{e_i=+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2}.\left(-\int_{e_j=e_i-z}^{e_j=\infty} \frac{-e^{-\frac{e_j-\mu}{s}}}{s(1+e^{-\frac{e_j-\mu}{s}})^2} de_j\right) de_i.\end{aligned}$$

Let us make the change of variable: $u = 1 + e^{-\frac{e_j-\mu}{s}}$, $du = -\frac{1}{s}e^{\frac{e_j-\mu}{s}} de_j$, $e_j = \infty \to u = 1$, $e_j = e_i - z \to u = 1 + e^{-\frac{e_i-z-\mu}{s}}$:

$$\begin{aligned}F_{\Delta E_{i,j}}(z) &= \int_{-\infty}^{+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2}.\left(-\int_{u=1+e^{-\frac{e_j-z-\mu}{s}}}^{u=1} \frac{1}{u^2} du\right) de_i \\ &= \int_{-\infty}^{+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2} \left[\frac{1}{u}\right]_{1+e^{-\frac{e_i-z-\mu}{s}}}^{1} de_i\end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2} \left( \frac{1}{1} - \frac{1}{1+e^{-\frac{e_i-z-\mu}{s}}} \right) de_i \\
&= \int_{-\infty}^{+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2} de_i - \int_{-\infty}^{+\infty} \frac{e^{-\frac{e_i-\mu}{s}}}{s(1+e^{-\frac{e_i-\mu}{s}})^2} \frac{1}{(1+e^{-\frac{e_i-z-\mu}{s}})} de_i.
\end{aligned}
$$

The first term is the integral over its whole domain of the PDF of the logistic distribution and is therefore 1 by definition. For the second term, we make a change of variable: $u = 1 + e^{-\frac{e_i-\mu}{s}}$, $du = -\frac{1}{s} e^{\frac{e_i-\mu}{s}} de_j$, $e_i = \infty \to u = 0$, $e_j = -\infty \to u = \infty$. Moreover, we define $\beta = e^{\frac{z}{s}}$:

$$
\begin{aligned}
F_{\Delta E_{i,j}}(z) &= 1 + \int_{\infty}^{0} \frac{1}{(1+u)^2} \frac{1}{1+\beta u} du \\
&= 1 + \int_{\infty}^{0} \frac{1}{(1-\beta)^2} \left( \frac{-\beta u + (1-2\beta)}{(1+u)^2} + \frac{\beta^2}{1+\beta u} \right) du \\
&= 1 + \frac{1}{(1-\beta)^2} \int_{\infty}^{0} \frac{-\beta u + 1 - 2\beta}{(1+u)^2} du + \frac{\beta^2}{(1+\beta)^2} \int_{\infty}^{0} \frac{1}{1+\beta u} du \\
&= 1 + \frac{1}{(1-\beta)^2} \int_{\infty}^{0} \frac{-\beta(1+u) + (1-\beta)}{(1+u)^2} du + \frac{\beta^2}{(1+\beta)^2} \int_{\infty}^{0} \frac{1}{1+\beta u} du \\
&= 1 + \frac{-\beta}{(1-\beta)^2} \int_{\infty}^{0} \frac{(1+u)}{(1+u)^2} du + \frac{(1-\beta)}{(1-\beta)^2} \int_{\infty}^{0} \frac{1}{(1+u)^2} du + \frac{\beta^2}{(1-\beta)^2} \int_{\infty}^{0} \frac{1}{1+\beta u} du \\
&= 1 + \frac{-\beta}{(1-\beta)^2} \int_{\infty}^{0} \frac{1}{1+u} du + \frac{1}{1-\beta} \int_{\infty}^{0} \frac{1}{(1+u)^2} du + \frac{\beta}{(1-\beta)^2} \int_{\infty}^{0} \frac{1}{\frac{1}{\beta}+u} du \\
&= 1 + \frac{\beta}{(1-\beta)^2} \int_{\infty}^{0} \frac{-1}{1+u} du + \frac{\beta}{(1-\beta)^2} \int_{\infty}^{0} \frac{1}{\frac{1}{\beta}+u} du + \frac{1}{1-\beta} \int_{\infty}^{0} \frac{1}{(1+u)^2} du \\
&= 1 + \frac{\beta}{(1-\beta)^2} \int_{\infty}^{0} \left( \frac{-1}{1+u} + \frac{1}{\frac{1}{\beta}+u} \right) du + \frac{1}{1-\beta} \int_{\infty}^{0} \frac{1}{(1+u)^2} du \\
&= 1 + \frac{\beta}{(1-\beta)^2} \left[ -\ln(1+u) + \ln\left(\frac{1}{\beta}+u\right) \right]_{\infty}^{0} + \frac{1}{1-\beta} \left[ -\frac{1}{1+u} \right]_{\infty}^{0} \\
&= 1 + \frac{\beta}{(1-\beta)^2} \left[ \ln\left(\frac{\frac{1}{\beta}+u}{1+u}\right) \right]_{\infty}^{0} + \frac{1}{1-\beta} \left[ -\frac{1}{1+u} \right]_{\infty}^{0} \\
&= 1 + \frac{\beta}{(1-\beta)^2} \left( \ln\left(\frac{\frac{1}{\beta}}{1}\right) - \ln(1) \right) + \frac{1}{1-\beta} \left( -\frac{1}{1} - (-0) \right) \\
&= 1 + \frac{\beta}{(1-\beta)^2} \ln\left(\frac{1}{\beta}\right) - \frac{1}{1-\beta} \\
&= 1 - \frac{\beta \ln(\beta)}{(1-\beta)^2} - \frac{1}{1-\beta} \\
&= 1 - \frac{\frac{z}{s}.e^{\frac{z}{s}}}{(1+e^{\frac{z}{s}})^2} - \frac{1}{1-e^{\frac{z}{s}}}
\end{aligned}
$$

$$
\begin{aligned}
&= 1 - \frac{z}{s}\frac{e^{\frac{z}{s}}}{(1-e^{\frac{z}{s}})^2} - \frac{1}{1-e^{\frac{z}{s}}} \\
&= 1 - \frac{1 + (1-\frac{z}{s})e^{\frac{z}{s}}}{(1-e^{\frac{z}{s}})^2} \\
&= \frac{1 - e^{-\frac{z}{s}}(1+\frac{z}{s})}{(1-e^{-\frac{z}{s}})^2}
\end{aligned}
$$

$$
\begin{aligned}
f_{\Delta E_{ij}}(z) &= \frac{\partial F_{\Delta E_{ij}}(z)}{\partial z} \\
&= \frac{-e^{\frac{z}{s}}(e^{\frac{z}{s}}(\frac{z}{s}-2) + \frac{z}{s} + 2)\left(\frac{1}{e^{\frac{z}{s}}}\right)^3}{(1-e^{\frac{z}{s}})^3\left(\frac{1}{e^{\frac{z}{s}}}\right)^3} \\
&= \frac{e^{-\frac{z}{s}}\left((\frac{z}{s}-2) + (\frac{z}{s}+2)e^{-\frac{z}{s}}\right)}{(1-e^{-\frac{z}{s}})^3}.
\end{aligned}
$$

Let us call this the *diff-log* distribution. From there, we can compute the distribution of $P_{ij}$:

$$
\Delta E_{ij} = \frac{1}{\lambda}\ln\left(\frac{P_{ij}}{1-P_{ij}}\right)
$$

$$
\begin{aligned}
F_{P_{ij}}(p) &= \mathbb{P}[P_{ij} \leqslant p] \\
&= \mathbb{P}\left[\frac{1}{1+e^{-\lambda\Delta E_{ij}}} \leqslant p\right] \\
&= \mathbb{P}\left[1 + e^{-\lambda\Delta E_{ij}} \geqslant \frac{1}{p}\right] \\
&= \mathbb{P}\left[e^{-\lambda\Delta E_{ij}} \geqslant \frac{1-p}{p}\right] \\
&= \mathbb{P}\left[e^{\lambda\Delta E_{ij}} \leqslant \frac{p}{1-p}\right] \\
&= \mathbb{P}\left[\lambda\Delta E_{ij} \leqslant \ln\left(\frac{p}{1-p}\right)\right] \\
&= \mathbb{P}\left[\Delta E_{ij} \leqslant \frac{1}{\lambda}\ln\left(\frac{p}{1-p}\right)\right] \\
&= F_{\Delta E_{ij}}\left(z \leqslant \frac{1}{\lambda}\ln\left(\frac{p}{1-p}\right)\right) \\
&= \frac{1 - e^{-\frac{1}{s\lambda}\ln\left(\frac{p}{1-p}\right)}\left(1 + \frac{1}{s\lambda}\ln\left(\frac{p}{1-p}\right)\right)}{\left(1 - e^{-\frac{1}{s\lambda}\ln\left(\frac{p}{1-p}\right)}\right)^2}.
\end{aligned}
$$

Finally, we derive the expression of the Beta-like distribution of $P_{ij}$:

$$F_{P_{ij}}(p) = \frac{1 - \left(\frac{p}{1-p}\right)^{-\frac{1}{s\lambda}} \left(1 + \frac{1}{s\lambda} \ln\left(\frac{p}{1-p}\right)\right)}{\left(1 - \left(\frac{p}{1-p}\right)^{-\frac{1}{s\lambda}}\right)^2}$$

$$f_{P_{ij}}(p) = \frac{d}{dp}\left(F_{P_{ij}}(p)\right) = \frac{\frac{1}{s\lambda}\left(\frac{p}{1-p}\right)^{\frac{1}{s\lambda}} \left(-2\left(\frac{p}{1-p}\right)^{\frac{1}{s\lambda}} + \frac{1}{s\lambda}\left(\frac{p}{1-p}\right)^{\frac{1}{s\lambda}} \ln\left(\frac{p}{1-p}\right) + \frac{1}{s\lambda}\ln\left(\frac{p}{1-p}\right) + 2\right)}{(1-p)p\left(\left(\frac{p}{1-p}\right)^{\frac{1}{s\lambda}} - 1\right)^3}.$$

## E.4 Symmetry of difference of i.i.d. distributions

Let $X$ and $Y$ be two independent identically distributed random variables: $f_X(z) = f_Y(z)$. This means that their joint distribution is symmetric in $x$ and $y$: $f_{X,Y}(x, y) = f_{X,Y}(y, x)$. Therefore, their difference $X - Y$ and $Y - X$ should be identically distributed as well: $f_{X-Y}(z) = f_{Y-X}(z)$. We can notice that by construction $f_{Y-X}(z) = f_{X-Y}(-z)$, which implies $f_{X-Y}(z) = f_{X-Y}(-z)$. This concludes the proof that the distribution of the difference of two i.i.d. random variables is necessarily symmetric around 0.

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**École polytechnique de Louvain**
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**