

Article

Not peer-reviewed version

A Practical Guide to the Multi-Dimensional Elo Rating System

[Adam Hamilton](#)*, [Anna Kalenkova](#), Matthew Roughan

Posted Date: 2 June 2026

doi: 10.20944/preprints202606.0161.v1

Keywords: rating; ranking; matrix completion; distributed algorithms



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Practical Guide to the Multi-Dimensional Elo Rating System

Adam Hamilton ^{1,*}, Anna Kalenkova ² and Matthew Roughan ¹

¹ School of Mathematical Sciences, Adelaide University

² School of Computer Science and Information Technology, Adelaide University

* Correspondence: adam.h.hamilton@adelaide.edu.au

Abstract

The multi-dimensional Elo rating system was designed to be a method of extending the more popular Elo system to model a wider variety of phenomena based on pairwise comparisons. This is useful in comparing the behaviours of machine learning models that would otherwise be black boxes. Unfortunately, the mElo rating system as it is currently formulated in the academic literature contains feedback loops that prevent the mElo rating system from correctly inferring an expected pairwise comparison matrix. In this paper, we propose an altered version of the mElo rating system that removes the feedback loops and ensures that a certain level of accuracy can be achieved. Furthermore, we demonstrate how the geometry associated with the model behind the multi-dimensional Elo rating system introduces complexities not present in the Elo rating system. We discuss some of these differences and how they affect the use and interpretation of the mElo rating system. We conclude the paper with a list of recommendations for the practitioner to ensure that the mElo rating system is being used correctly to infer a unique pairwise comparison matrix.

Keywords: rating; ranking; matrix completion; distributed algorithms

1. Introduction

The Elo rating system and its variants are very popular for comparing players in games (e.g., Chess), sports (e.g., FIFA), and e-sports (e.g., StarCraft) [1–5]. The results are useful for estimating the outcomes as well as other tasks such as match-making.

Most rating systems work by assuming that players' skills can be quantified by a single number. As a result, the Elo rating system does not work well when the relation "is more likely to win against" is non-transitive. This is particularly important in the field of machine learning. Machine Learning (ML) results are often treated as black boxes, where the most effective way to assess whether one is better than another is to simply observe which one "wins" at a task. Hence, a system such as Elo that uses this type of binary input is invaluable. However, often ML solutions cannot be ranked using a total ordering [6,7].

The *multi-dimensional Elo* (mElo) rating system was first published in 2018 as a way of extending the Elo rating system to estimate probabilities of victory in the case where non-transitivity is present [6], in particular for ML applications. Unlike the Elo rating system, which gives every player a scalar-valued rating, the mElo rating system gives every player a vector-valued rating, and the probability of victory is given by a logistic function of a quadratic form between the two ratings. This rating system allows one to infer probabilities of victory in non-transitive settings.

The mElo rating system was designed to be a method of extending the more popular Elo system to situations where it would not normally work. This would, in theory, allow one to model a richer set of interactions. However, in practice, the mElo rating system routinely underperforms when compared to the Elo rating system [3,8,9]. We show here that the mElo rating system, as it is known in the academic

literature, has not been used/implemented correctly; and we hypothesize that this fact is largely to blame for mElo's underperformance.

The purpose of this paper is to answer the question: How does one know whether they are using the mElo rating system correctly? In this paper, we show that mElo rating system does work, but only if one is careful. In the original paper on the mElo rating system [6], there exists feedback loops which cause the mElo ratings to stop approximating the expected payoff matrix. Our paper bridges a gap in the existing literature by showing how the mElo rating system can avoid these feedback loops and how one can be certain that the mElo ratings are being used to estimate unique maximum likelihood estimators without getting stuck in local optima.

The underlying statistical model behind mElo maps vector-valued ratings to a matrix of probabilities of victory. This model has a greater level of expressive power than Elo. Unfortunately, that comes at a cost. The statistical model underpinning the mElo rating system is more complicated than the Bradley-Terry model used in Elo [10], and consequently, there are more ways that things can go wrong when using the mElo rating system. And the mElo rating system is not deterministic like the Elo rating system. This has the effect of making the results less amenable to interpretation.

In the original formulation of the mElo rating system [6], the update algorithm introduces the potential for feedback loops which can prevent the update algorithm from finding optimal mElo ratings. The intention of the mElo rating system is to find the optimum of a log-likelihood function. We use this understanding to present conditions on the data for this optimisation problem to have a unique solution modulo symmetry, which assures us that the rating system works correctly.

This paper also looks at situations in which it appears that the mElo rating system is not working when in fact it is. We show that there are multiple sets of mElo ratings that correspond to the same set of probabilities of victory, arising from the symmetry group associated with the optima. This can make it difficult to interpret mElo ratings. We argue that this does not mean that the mElo rating system is flawed, even if experiments may not appear reproducible.

The aim of this paper is to be useful to the statistician or data scientist who may be considering using the mElo rating system. Additionally, we hope the paper is of interest to those who are work in the design and analysis of algorithms and would appreciate a specific case study.

In summary, the contributions of this paper are.

1. We refine the mElo algorithm from [6] to eliminate the problem of feedback and to be more robust.
2. We discuss necessary and sufficient conditions to determine whether probabilities of victory can be uniquely determined from a given dataset.
3. We explain why the mElo rating system will not find a maximum likelihood if all players begin with the same rating.
4. We consider the identifiability problem associated with the mElo rating system. We show that there are different sets of ratings that correspond to the same set of probabilities. We discuss the implications this has for interpreting mElo ratings, and explain that while it has its inconveniences, this ambiguity in mElo ratings is actually a good thing.
5. We verify these theoretical results using numerical experiments.
6. Finally, we conclude with a checklist for the practitioner, to allow a data scientist to use the mElo rating system correctly.

2. Background

The multi-dimensional Elo (mElo) rating system [6] is an extension of the better known Elo rating system [1]. The main difference between mElo and Elo is that mElo can be used to estimate probabilities of victory when the binary relation "is more likely to win against" is non-transitive. That is we have situations where there exist three players A, B, and C where Player A is more likely to win against Player B, who is more likely to win against Player C, who in turn, is more likely to win against Player A.

In this section, we introduce the mElo rating system. In Section 7.2, we provide a numerical simulation on a test problem.

Both the Elo and mElo rating systems assume that the probabilities of victory can be given according to an *expected payoff matrix* P , where

$$P_{i,j} = \Pr[\text{Player } i \text{ beats Player } j].$$

Because both Elo and mElo are used for modelling zero-sum games without draws¹, we have the requirement

$$P_{i,j} = 1 - P_{j,i}.$$

The goal of both the Elo and mElo rating systems is to estimate this expected payoff matrix based on a dataset consisting of an order list of the outcomes of multiple games.

Neither the Elo nor mElo rating systems work directly with the expected payoff matrix, instead both systems work with an *advantage matrix* A , where the advantage matrix is related to the expected payoff matrix by

$$P = \sigma(A),$$

where σ is the logistic function². The zero-sum nature of the game implies that A is skew-symmetric, i.e., $A^T = -A$.

The Elo rating system implicitly assumes that the advantage matrix is a *strongly transitive additive comparison matrix* (STACM) [6,11]. A STACM is a matrix M where for all unique triples (i, j, k) , $M_{i,j} + M_{j,k} + M_{k,i} = 0$. One important feature of STACMs is that they can be expressed by finding a rating r_i for every player, such that the advantage matrix's entries are given by $A_{i,j} = r_i - r_j$. In the context of Elo, these numbers r_i are called *Elo ratings* and the Elo rating system works by estimating these Elo ratings (up to addition of a constant) using stochastic gradient descent [8].

The set of STACMs form a strict subset of rank-2 skew-symmetric matrices. The space of skew-symmetric matrices orthogonal to the space of STACMs are called *cyclic matrices*. The mElo rating system is able to model more complicated relationships by assuming that the advantage matrix is skew-symmetric matrix of fixed rank that contains a non-trivial cyclic component. The mElo rating system contains a hyper-parameter k , where it assumes that the advantage matrix is of rank $2k$ (skew-symmetric matrices must be of even rank).

Let M_{2k} be the set of skew-symmetric matrices of rank at most $2k$. The types of advantage matrices inferred by the Elo rating system and the mElo rating system, for different values of k are shown below.

$$\{\text{STACMs}\} \subset M_2 \subset M_4 \subset \dots \subset M_{2k} \subset \dots \subset M_{\text{full rank}}.$$

The mElo rating system calculates probabilities of victory by assigning every player a $2k$ dimensional vector $C_i \in \mathbb{R}^{2k}$ called their *mElo rating*. The advantage matrix is given by

$$A = C^T \Omega C,$$

where $C \in \mathbb{R}^{2k \times m}$ is a matrix whose columns are the mElo ratings of each player and $\Omega \in \mathbb{R}^{2k \times 2k}$, is the block diagonal matrix whose blocks are given by

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (1)$$

¹ There are a number of extensions to ratings systems to allow for draws and other types of non-binary outcomes.

² Note: Other rating systems can be created using other sigmoid functions.

Therefore, given two players i and j , with mElo ratings C_i and C_j , the mElo rating system predicts that Player i will win against Player j with probability

$$P_{i,j} = \sigma(C_i^\top \Omega C_j).$$

The Elo and mElo rating system both include a method of updating the ratings of two players based on the outcome of a game between them. Let us consider a game between players i and j . If Player i wins the game, then we assign $S_i = 1$ and $S_j = 0$, and if j wins, then $S_i = 0$ and $S_j = 1$, and the (un-normalised) update rule for the players' mElo ratings takes the form

$$C_i^{(t+1)} = C_i^{(t)} + \eta \left(S_i - \sigma((C_i^{(t)})^\top \Omega C_j^{(t)}) \right) \Omega C_j^{(t)}, \quad (2)$$

for Player i , and

$$C_j^{(t+1)} = C_j^{(t)} + \eta \left(S_j - \sigma((C_j^{(t)})^\top \Omega C_i^{(t)}) \right) \Omega C_i^{(t)},$$

for Player j , where

- $C_i^{(t)}$ is the mElo rating of Player i at game t .
- $\eta > 0$ is the learning rate.

This update formula can be derived by differentiating the log-likelihood function associated with a single game.

$$\ell = \log \left(S_i \sigma((C^\top \Omega C)_{i,j}) + (1 - S_i) \sigma((C^\top \Omega C)_{j,i}) \right),$$

with respect to the entries of C . That is, the mElo ratings are updated after each game using stochastic gradient descent on the log-likelihood function.

A key advantage of both the Elo system and mElo system (as described above) is that update rules affect only the ratings of the player involved in the game. And likewise predictions about outcomes can be made using only the ratings of the players concerned. We refer to this as the *distributed property* of a ratings system, and it is very useful, particularly when the pool of players is large.

To summarise, the mElo_{2k} rating system consists of

- A pair of hyper-parameters k and η , where η is the learning rate.
- An expected payoff matrix $A \in \mathbb{R}^{m,m}$ of rank $2k$.
- An expected payoff matrix given by $P = \sigma(A)$.
- A vector of parameters $C_i \in \mathbb{R}^{2k}$ for each player i .
- A formula for updating ratings after the completion of each game.

2.1. Original Formulation of the mElo Rating System

Readers of [6] may notice that our rating system differs slightly from the mElo rating system presented there. In this subsection, we explain how our formulation of the mElo rating system differs from theirs and why. In the original mElo rating system, the authors also model the advantage matrix as a rank- $2k$ skew-symmetric matrix, but the authors also assign each player an Elo rating in addition to a vector-valued rating. After each game, both the Elo ratings and the vector-valued ratings are updated using stochastic gradient descent. Expressed mathematically, this version of the mElo rating system assigns every player a rating (r_i, C_i) , where $r_i \in \mathbb{R}$, and $C_i \in \mathbb{R}^{2k}$. The probability of victory is calculated by

$$P_{i,j} = \sigma(r_i - r_j + C_i^\top \Omega C_j)$$

and ratings are updated according to

$$r_i^{(t+1)} = r_i^{(t)} + \eta_1 (S_i - P_{i,j}),$$

and

$$C_i^{(t+1)} = C_i^{(t)} + \eta_2 \left(S_i - \sigma((C_i^{(t)})^\top \Omega C_j^{(t)}) \right) \Omega C_j^{(t)},$$

where η_1 and η_2 are separate learning rates. The idea behind this choice was to separate the inference of the cyclic component from the transitive Elo ratings.

In our paper, we will not focus on the scalar component r_i of the mElo rating system. We do this for two reasons. First, we believe this formulation to be misleading. It is not true that the cyclic and transitive components are truly separated. There is no mechanism in place that ensures that $C^\top \Omega C$ is a cyclic matrix. In practice, the component of the advantage matrix given by C can also contain a large transitive component.

Second, we omit the scalar component r in order to simplify the analysis in subsequent sections of this paper. The results presented are not affected by the presence of a scalar component and will hold in the original formulation of [6], although it would be harder to see what is happening.

3. Feedback Loops

One crucial difference between the Elo and mElo rating systems is that the lengths of the gradient vectors in the likelihood function are bounded in Elo and unbounded in mElo. Both systems attempt to find maximum likelihood estimators by applying stochastic gradient descent to their respective log-likelihood functions. However, since the gradients in the mElo rating system can be of any size, the estimated parameters can change by huge amounts after even a single game. These huge steps can lead to feedback cycles in which our parameter estimates either grow incredibly large and miss the optima of the log-likelihood function. Or the parameter estimates can negatively interfere with each other, shrink towards the origin and also, miss the optima of the log-likelihood function. Fortunately, these feedback problems can be mitigated by normalising the parameter updates with each game.

In this section of the paper, we demonstrate the problems of positive and negative feedback on a toy problem involving two players, $k = 1$ (and hence a rank-2 advantage matrix) and a learning rate $\eta \in (0, 1)$. We show how the learning rate of the mElo update algorithm can lead to the players' mElo ratings shrinking towards the origin at an exponential rate.

Let us say we have two players called y and z . Initially, we assign these players the rating vectors

$$\begin{aligned} C_y^{(0)} &= (x, 0)^\top, \\ C_z^{(0)} &= (0, x)^\top. \end{aligned}$$

After players y and z play a game, their ratings are updated according to

$$\begin{aligned} C_y^{(t+1)} &= C_y^{(t)} + \eta \left(S_y - \sigma(C_y^{(t)\top} \Omega C_z^{(t)}) \right) \Omega C_z^{(t)}, \\ C_z^{(t+1)} &= C_z^{(t)} - \eta \left(S_y - \sigma(C_y^{(t)\top} \Omega C_z^{(t)}) \right) \Omega C_y^{(t)}. \end{aligned}$$

Since, $C_y^{(0)} = (x, 0)^\top$ and $C_z^{(0)} = (0, x)^\top$, we can simplify the above expressions and show that.

$$x(t+1) = x(t) + \eta \left(S_y - \sigma(x(t)^2) \right) x(t).$$

We are going to look at what happens when $x(t)$ is small and $\eta \in (0, 1)$. We will look at when it is possible for x to move away from the origin and when x will be pulled back to the origin.

When $0 < \eta < 1$ the term $\eta(S_i - \sigma(x(t)^2))$ will always lie in the interval $(-1, 1)$. Now, $x(t+1) = x(t) \left(1 + \eta(S_y - \sigma(x(t)^2)) \right)$, so regardless of the outcome of the game, we always multiply $x(t)$ by a positive number. This means that regardless of the outcomes of the games, the sign of $x(t)$ will never change. Hence, the odds $P_{y,z}$ given by the mElo rating system will be in the following intervals

- $(0, 0.5)$, if x starts with a negative value.

- (0.5, 1), if x starts with a positive value.
- 0.5, if x starts at zero, i.e., $x(t) = 0, \forall t$.

Without loss of generality, we presume that x begins with a positive value. If Player z wins, we multiply x by $(1 - \eta\sigma(x^2))$, and if Player y , then we multiply x by $(1 + \eta\sigma(x^2))$. Assuming that x is close to zero, $\sigma(x^2) \approx 0.5$ and after N games, we would expect Player y to win Np of these games and lose $N(1 - p)$ games, where p is the true probability that Player y wins against Player z . Over the course of N games, for small values of x , the value of x after these games can be approximated

$$\begin{aligned} x^{t+N} &\cong x(1 + \eta/2)^{Np}(1 - \eta/2)^{N(1-p)} \\ &\cong x \left((1 + \eta/2)^p (1 - \eta/2)^{1-p} \right)^N. \end{aligned}$$

As the number of games N approaches infinity, x will increase exponentially, decrease exponentially or do neither if $(1 + \eta/2)^p (1 - \eta/2)^{1-p}$ is greater than one, less than one, and equal to one, respectively.

This shows that if the probability p that i beats j satisfies the inequality

$$p < \frac{\log\left(\frac{2}{2-\eta}\right)}{\log\left(\frac{2+\eta}{2-\eta}\right)}. \quad (3)$$

Then x will tend to shrink towards the origin at an exponential rate. As was mentioned before, if $p < 0.5$, then, regardless of the outcomes of the games, the mElo rating vectors will never predict that i will win the game with probability $p < 0.5$.

From Equation (3) we see that for values of $\eta \in (0, 1)$, there exist values of p such that if i beats j with probability p , then C_i and C_j will still shrink towards the origin at an exponential rate. This means that the mElo ratings will not approximate the expected payoff matrix, which is a key purpose of the mElo rating system. For a more in-depth discussion and examples where the mElo ratings grow without bound instead of shrinking, the reader is invited to consult [12].

3.1. Normalised Updates

These feedback loops occur because in the mElo rating system, the difference between $C^{(t)}$ and $C^{(t+1)}$ can grow without bound. We recommend avoiding these feedback loops by normalising our updates, i.e., after a game between Players i and j , the mElo_{2k} ratings vectors are updated according to the formulae:

$$C_i^{(t+1)} = C_i^{(t)} + \frac{\eta \left(S_i - \sigma(\hat{P}_{i,j}^{(t)}) \right) \Omega C_j^{(t)}}{\sqrt{|C_i^{(t)}|^2 + |C_j^{(t)}|^2}}, \quad (4)$$

and

$$C_j^{(t+1)} = C_j^{(t)} - \frac{\eta \left(S_i - \sigma(\hat{P}_{i,j}^{(t)}) \right) \Omega C_i^{(t)}}{\sqrt{|C_i^{(t)}|^2 + |C_j^{(t)}|^2}}, \quad (5)$$

where

$$\hat{P}_{i,j}^{(t)} = C_i^{(t)\top} \Omega C_j^{(t)}.$$

This means that the mElo update algorithm still explores the space of $2k \times m$ -sized matrices C , but does so in only in steps whose size has upper and lower bounds.

3.2. Numerical Simulations

To illustrate the results, we run an experiment in which we have two players: y and z . Player y begins with a mElo rating of (0.1, 0) and Player z begins with a mElo rating of (0, 0.1). In this experiment, we have two players with perpendicular mElo ratings of equal length. When we set the learning rate to be $\eta = 0.9$, and the probability of y beating z is equal to 0.6, then Inequality Equation (3) is satisfied, and we would expect to see exponential shrinkage. We can see that the lengths of both

players' mElo ratings do, in fact, shrink at an exponential rate. A diagram of this experiment is shown in Figure 1.

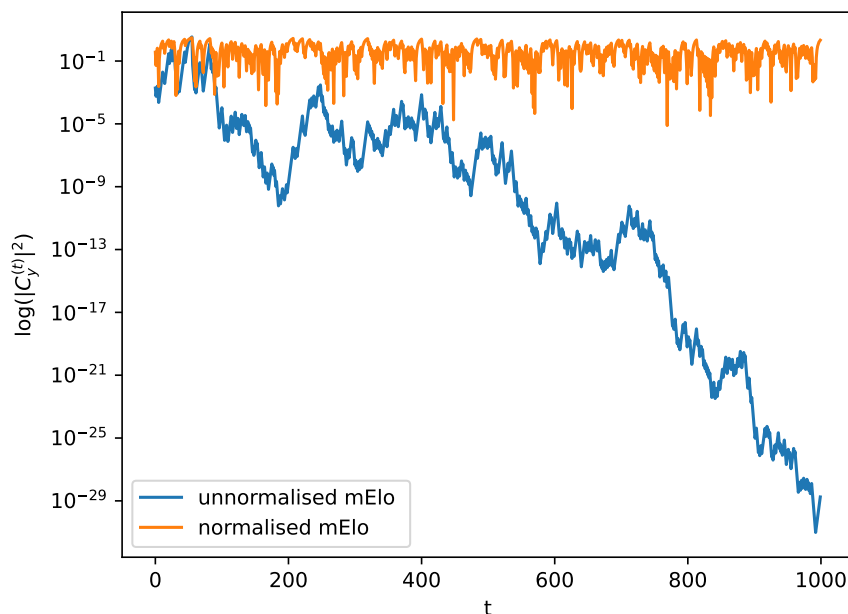


Figure 1. A semi-log plot of the lengths of mElo ratings of Players y and z with a learning rate of $\eta = 0.9$ and $P_{y,z} = 0.6$. Since both players have perpendicular mElo ratings of the same length, the two lines in this plot coincide. We can see that the lengths of the player's mElo ratings decrease at an exponential rate as predicted.

This is a single specific case of feedback loops in the mElo rating system. If the learning rate is too large, then the mElo rating vectors will increase exponentially rather than decrease. In situations where there are more than two players, an un-normalised mElo rating system can avoid these feedback loops when the learning rate is low. The reader is encouraged to read [12] for more information. The fact that we can show that it does break down in certain edge cases is representative of a larger problem with the mElo rating system and motivates the need for normalised updates.

4. Initialising mElo Ratings

The standard Elo rating system is an iterative algorithm and thus requires initialisation, but it may be initialised by giving every player the same rating. This is appealing as it treats the players equally and can be simply implemented. It also allows for easy, incremental addition of players (for instance, as new players join a game like Chess).

It seems to have been assumed that the mElo rating could be similarly initialised³, but it cannot. In this section, we show that if we assign every player the same mElo rating, then the set of players' mElo ratings will be stuck in a local, stationary point, and will not be able to find a maximum likelihood estimator for C . We then give necessary and sufficient conditions to initialise the mElo rating vectors such that the mElo ratings will approach a maximum likelihood estimator with probability one.

The mElo rating system assumes that the advantage matrix is of rank $2k$. The ratings are updated using Equation (2), *i.e.*,

$$C_i^{(t+1)} = C_i^{(t)} + \eta \left(S_i - \sigma \left((C_i^{(t)})^\top \Omega C_j^{(t)} \right) \right) \Omega C_j^{(t)},$$

³ It is hard to be conclusive as initialisation processes are rarely reported in detail, but subsequent results support the result.

where the matrix Ω is a block diagonal matrix made up of copies of

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

If we consider the $(2n - 1)$ th and $(2n)$ th entries of player i 's mElo rating, $c_i^{(t)}[2n - 1, 2n]$, then the direction in which we update this vector is

$$\delta c_i[2n - 1, 2n] \propto \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} c_j^{(t)}[2n - 1, 2n].$$

Each 2-dimensional component of a player i 's mElo rating is updated in the direction of player j 's corresponding 2-dimensional component multiplied by Ω . The only effect the other components of c_j have on c_i is through the term $(S_i - P_{i,j})$. This leads to the following theorem.

Theorem 1. Consider a $2k \times m$ matrix C of mElo ratings. If the column space of $C^{(0)}$ is spanned by the vectors v_1, v_2, \dots, v_d , then the column space of $C^{(t)}$ will be spanned by the vectors $v_1, \Omega v_1, v_2, \Omega v_2, \dots, v_d, \Omega v_d$ with probability one for all sufficiently large values of time step t (presuming all players can play one another).

Proof. Let C be a $2k \times m$ matrix of mElo ratings with columns spanned by v_1, v_2, \dots, v_d , with each iteration of the update algorithm. We take two players i and j and add a scalar multiple of Ωc_j to c_i , and some scalar multiple of Ωc_i to c_j . This means that after a large number of iterations, each player's mElo rating will be given by some linear combination of $\Omega^k c_i$, for $i \in \{1, 2, \dots, m\}$. Since $\Omega^2 = -I_{2k \times 2k}$, the columns of C will all be made up of linear combinations of c_i and Ωc_i , for all values of $i \in \{1, 2, \dots, m\}$. Thus, the space is spanned by the vectors $v_1, \Omega v_1, v_2, \Omega v_2, \dots, v_d, \Omega v_d$. \square

Notice how in Theorem 1, the rank of C almost surely changes from $\dim(\text{span}(v_1, v_2, \dots, v_d))$ to $\dim(\text{span}(v_1, \Omega v_1, v_2, \Omega v_2, \dots, v_d, \Omega v_d))$, and does not increase past that value.

A corollary of this fact is that if we initialise every player with the same mElo rating c , then the matrix $C^{(t)}$ will have a column space that will always be spanned by c and Ωc and thus never be larger than rank 2. That is, the matrix cannot express any of the mElo ratings outside of this very limited subset.

This means that the only way we can approximate a maximum likelihood estimator for a rank $2k$ advantage matrix is if we initialise C with full rank or with a column space X such that $X \cup \Omega X$ spans \mathbb{R}^{2k} .

If we assign players' initial mElo ratings with random values drawn from a continuous distribution, the initial matrix C will be of full rank with probability one.

This by itself is a rather loose set of conditions on randomly initialising the mElo ratings. The condition is just that we sample our matrix C from a continuous distribution over a subset of $\mathbb{R}^{2k \times m}$ with positive measure. In the interests of making the mElo rating system fair and simple to implement, we suggest initialising the mElo ratings independently. This treats each player equally, and eliminates the need for any extra coordination between the population of players.

The result is a very simple initialisation, but it has one inconvenient consequence. Since we are initialising ratings randomly, we no longer have a deterministic rating system. However, pseudo-random numbers (PRNs) provide a means to specify a reproducible rating system if specify details such as PRN seeds and the exact method of generation.

5. Symmetries and Non-Unique Ratings

One of the main sources of complication in the mElo rating system is that it has a significantly more complex symmetry group than the Elo rating system.

In the Elo rating system, if we add a constant to every player's Elo rating, then the underlying advantage matrix does not change, *i.e.*, it is invariant under shifts. Usually, there is an additional criterion added to an Elo-like system to create a unique solution: for instance, it is common to require that the mean rating is a fixed value (such as zero). This removes the translational symmetry, resulting in unique solutions.

The symmetry of the mElo rating system is the *group*⁴ of transformations on the matrix C that leave the advantage matrix unchanged. That is, the group of transformations g on the matrix of ratings C such that $(gC)^\top \Omega (gC) = C^\top \Omega C$. This symmetry group is well-known in mathematics [13]. It is the *symplectic group*, defined by exactly this transformational invariance.

The fact that mElo has a more complicated symmetry group means that one cannot alter the mElo rating system to obtain an algorithm that estimates a unique, maximum likelihood estimator for C with the *distributed* property.

The distributed property of the Elo or mElo rating system means that the results of a game between Players i and j only affect the ratings of Players i and j . The distributed property implies that Player w 's rating won't change as a result of a game between players x and y . It is this distributed property that makes it possible to scale an algorithm like mElo to a massive population size. Every player is given their own ratings along with identical instructions on how to update it. There is no need for a single entity to keep track of each player. The system can therefore be *distributed* across a set of administrators, each keeping track of their own subset of players.

Uniqueness (of ratings) isn't just a convenience or a means towards reproducible experiments. If there were a unique matrix of ratings C corresponding to every advantage matrix A , then this would allow one to assign meaning to each of the ratings, which would in turn allow us to interpret the mElo ratings. For instance, one's Elo rating in Chess is used to assign Grand Master status (at a rating above 2500).

Unfortunately, in mElo, unique ratings don't exist without sacrificing the distributed property. The proof can be found in the final chapter of [12]. It is quite technical, so we omit it here. However, we can understand the gist of it if we consider that the rotational subgroup of the symplectic group commutes with the update process of the mElo rating system. This means, that the mElo rating system will behave exactly the same if we rotated all the mElo ratings by a given amount. However, that would change all of the ratings, not just those of the primary players. Hence, any transformation designed to force a unique orientation onto the set of ratings (*e.g.*, a transformation to ensure the mean direction was fixed in the same manner that Elo ensures the mean rating value is fixed) would necessarily change all of the ratings after each update.

It is a major difficulty to lose the distributed property, so in all implementations of mElo of which we are aware, uniqueness is sacrificed. This makes the mElo rating system harder to use when it comes to interpreting results, as the values of a single mElo rating vector will tell us very little. Only the relative orientations of ratings vectors has meaning.

One important takeaway from this section is that if two statisticians were to independently run the mElo rating system (with random initialisations) and fit mElo ratings to the same sets of players and game outcomes, then, with high probability, the resulting set of mElo rating vectors would differ. Likewise, if the same sets of players were to play in two tournaments, then ratings derived separately for the two tournaments would differ (because there would likely be different outcomes in some matches). But this is OK. It is not an indication that either statistician had made a mistake or that the system is misbehaving.

If one wishes to assess the accuracy of mElo, then one must assess the accuracy of the advantage matrices, not the ratings themselves.

⁴ We mean here the mathematical notion of a group, *i.e.*, a set of objects and a binary operation that satisfy group axioms: closure, associativity, identity, and the existence of inverses.

6. Symmetries and Non-Unique Ratings

One of the main sources of complication in the mElo rating system is that it has a significantly more complex symmetry group than the Elo rating system.

In the Elo rating system, if we add a constant to every player's Elo rating, then the underlying advantage matrix does not change, *i.e.*, it is invariant under shifts. Usually, there is an additional criterion added to an Elo-like system to create a unique solution: for instance, it is common to require that the mean rating is a fixed value (such as zero). This removes the translational symmetry, resulting in unique solutions.

The symmetry of the mElo rating system is the *group*⁵ of transformations on the matrix C that leave the advantage matrix unchanged. That is, the group of transformations g on the matrix of ratings C such that $(gC)^T \Omega (gC) = C^T \Omega C$. This symmetry group is well-known in mathematics [13]. It is the *symplectic group*, defined by exactly this transformational invariance.

The fact that mElo has a more complicated symmetry group means that one cannot alter the mElo rating system to obtain an algorithm that estimates a unique, maximum likelihood estimator for C with the *distributed* property.

The distributed property of the Elo or mElo rating system means that the results of a game between Players i and j only affect the ratings of Players i and j . The distributed property implies that Player w 's rating won't change as a result of a game between players x and y . It is this distributed property that makes it possible to scale an algorithm like mElo to a massive population size. Every player is given their own ratings along with identical instructions on how to update it. There is no need for a single entity to keep track of each player. The system can therefore be *distributed* across a set of administrators, each keeping track of their own subset of players.

Uniqueness (of ratings) isn't just a convenience or a means towards reproducible experiments. If there were a unique matrix of ratings C corresponding to every advantage matrix A , then this would allow one to assign meaning to each of the ratings, which would in turn allow us to interpret the mElo ratings. For instance, one's Elo rating in Chess is used to assign Grand Master status (at a rating above 2500).

Unfortunately, in mElo, unique ratings don't exist without sacrificing the distributed property. The proof can be found in the final chapter of [12]. It is quite technical, so we omit it here. However, we can understand the gist of it if we consider that the rotational subgroup of the symplectic group commutes with the update process of the mElo rating system. This means, that the mElo rating system will behave exactly the same if we rotated all the mElo ratings by a given amount. However, that would change all of the ratings, not just those of the primary players. Hence, any transformation designed to force a unique orientation onto the set of ratings (*e.g.*, a transformation to ensure the mean direction was fixed in the same manner that Elo ensures the mean rating value is fixed) would necessarily change all of the ratings after each update.

It is a major difficulty to lose the distributed property, so in all implementations of mElo of which we are aware, uniqueness is sacrificed. This makes the mElo rating system harder to use when it comes to interpreting results, as the values of a single mElo rating vector will tell us very little. Only the relative orientations of ratings vectors has meaning.

One important takeaway from this section is that if two statisticians were to independently run the mElo rating system (with random initialisations) and fit mElo ratings to the same sets of players and game outcomes, then, with high probability, the resulting set of mElo rating vectors would differ. Likewise, if the same sets of players were to play in two tournaments, then ratings derived separately for the two tournaments would differ (because there would likely be different outcomes in some matches). But this is OK. It is not an indication that either statistician had made a mistake or that the system is misbehaving.

⁵ We mean here the mathematical notion of a group, *i.e.*, a set of objects and a binary operation that satisfy group axioms: closure, associativity, identity, and the existence of inverses.

If one wishes to assess the accuracy of mElo, then one must assess the accuracy of the advantage matrices, not the ratings themselves.

7. Uniquely Recovering the Advantage Matrix

Another way to view the mElo rating system is through the lens of low-rank matrix completion. The purpose of the mElo ratings is to be able to infer the entries of a low-rank matrix (the advantage matrix) using a smaller number of observed entries (the outcomes of games between players) [14]. In this section, we review the matrix completion problem, show that mElo is a particular example of low-rank matrix completion, and use this fact to derive necessary and sufficient conditions for an advantage matrix to be uniquely recovered.

Matrix completion problems typically involve finding the entries of an unknown matrix $M \in \mathbb{R}^{n_1 \times n_2}$, given some observed entries $m_{i,j}, \{i, j\} \subset [n_1] \times [n_2]$. Without additional assumptions, this problem is impossible, so we assume that the matrix has some underlying structure. In this case, we presume that M has a fixed rank $r \leq \min(n_1, n_2)$. This assumption makes it possible to calculate the entries of M based on other entries under some conditions.

It is possible that the observed entries of the low-rank matrix M have also been corrupted by noise. In that case, the goal then becomes to find some rank- r matrix \hat{M} , that minimises a distance function between the observed partial matrix, and the completed low-rank matrix [14].

In the mElo rating system, we are attempting to infer the advantage matrix A , which is both rank- $2k$ and skew-symmetric (because the observed entries of the matrix are the outcomes of games with a winner and loser). The mElo rating system completes the matrix under noise by finding an advantage matrix \hat{A} , that maximises the log-likelihood function

$$\ell(C) = \sum_{\{i,j\} \in D} \log \left(S_i \sigma((C^\top \Omega C)_{i,j}) + (1 - S_i) \sigma((C^\top \Omega C)_{j,i}) \right),$$

where D is the ordered list of pairs of players who play each other and S_i is the indicator variable that equals one when Player i beats Player j , i.e., $S_j = 1 - S_i$. In most settings, only a subset of players will compete directly, and so the mElo rating system can be naturally viewed as an example of a matrix completion problem.

In deriving necessary and sufficient conditions for the advantage matrix to be uniquely completable, we take the necessary and sufficient conditions for generic matrix completion derived in [15,16] and adapt them to our purposes, which differ somewhat from typical matrix completion problems because of the addition of properties such as skew symmetry.

Definition 1. We say that Player i is comparable to a set of players Γ if there is a bounded maximum likelihood estimator for Player i 's mElo rating given fixed mElo ratings for every player in Γ .

Theorem 2. Take a set Γ of players and Player $i \notin \Gamma$. Let C_Γ be the set of mElo ratings of the players in Γ , and let players $sw = \{sw_1, sw_2, \dots, sw_\alpha\} \subseteq \Gamma$ be the set of players in Γ that play against Player i and win, and let $C_{sl} = \{C_{sw_1}, C_{sw_2}, \dots, C_{sw_\alpha}\} \in \mathbb{R}^{2k}$, be their respective mElo ratings. Let $sl = \{sl_1, sl_2, \dots, sl_\beta\} \subseteq \Gamma$ be the set of players in Γ that play against Player i and lose, and let $C_{sl} = \{C_{sl_1}, C_{sl_2}, \dots, C_{sl_\beta}\} \in \mathbb{R}^{2k}$, be their respective mElo ratings. Then Player i is comparable with the set Γ if and only if \mathbb{R}^{2k} is contained in the convex cone of the mElo vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$.

Remark: Note that the players in Γ can play against Player i multiple times, and therefore the sets sw and sl need not be disjoint.

Proof. A mathematically precise definition of Player i being comparable to the players in Γ is that the following log-likelihood function given below has a unique local maximum.

$$\ell(x) = \log(P\{c_i = x | C_\Gamma\}) = \sum_t \log \left((1 - S_{j(t)}) \sigma(x^\top \Omega C_{j(t)}) + S_{j(t)} \sigma(C_{j(t)}^\top \Omega x) \right),$$

where $j(t)$ is the player in Γ that plays player i at time step t and $S_{j(t)}$ is the indicator variable that is equal to 1 if Player $j(t)$ defeats Player i at time step t and is equal to zero otherwise.

Since the $S_{j(t)}$ is a binary variable, each term in the summand in $\ell(x)$ will either be $\log(\sigma(x^\top \Omega C_{j(t)}))$ if Player i wins or $\log(\sigma(C_{j(t)}^\top \Omega x))$ if Player j wins. These functions are all concave down and bounded from above but not bounded below. If Player i beats Player $j(t)$ and we move x in the direction $\Omega C_{j(t)}$, then the corresponding term in $\ell(x)$ will increase and if we move x in any direction perpendicular to $\Omega C_{j(t)}$, the corresponding term in $\ell(x)$ will not change value. Similarly if Player i loses against Player $j(t)$, then moving x in the direction $-\Omega C_{j(t)}$ will increase the corresponding term in $\ell(x)$, but would not change if we moved x in any direction parallel to $-\Omega C_{j(t)}$.

Since each summand in $\ell(x)$ is a concave down function, the function $\ell(x)$ is concave down as well. Since each summand is bounded from above but not from below, and there are a finite number of summands, $\ell(x)$ is also bounded from above.

If there exists a line v such that we can take all of the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$, project them onto v , and have the projection be positive then that means that if we move x along this line v , then $\ell(x)$ will increase indefinitely. If no such line v exists, then no matter which direction we move x in, if we go far enough, one of the summands, will decrease indefinitely, and $\ell(x)$ will approach $-\infty$.

Since $\ell(x)$ is concave down, and if $\lim_{t \rightarrow \infty} \ell(x + tv) = -\infty, \forall v \in \mathbb{R}^{2k}$, then $\ell(x)$ must have a unique local maximum and Player i is therefore comparable to the players in Γ . This is true if and only we cannot construct a line v .

The statement, there does not exist a v such that $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$ all have positive projections onto v is equivalent to the statement that \mathbb{R}^{2k} is contained in the convex cone of the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$. Therefore, Player i is comparable to Γ if and only if \mathbb{R}^{2k} is contained in the convex cone of the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$.

So to summarise, if \mathbb{R}^{2k} is contained in the convex cone of the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$, then that means if we move x in any direction sufficiently far away from the origin, then $\ell(x)$ will approach $-\infty$. Thus $\ell(x)$ has a unique local maximum and i is comparable to Γ . This proves one direction of Theorem 2.

If \mathbb{R}^{2k} is not contained in the convex cone of the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$, then there exists some direction v such that all the vectors $\Omega C_{sw_1}, \Omega C_{sw_2}, \dots, \Omega C_{sw_\alpha}, -\Omega C_{sl_1}, -\Omega C_{sl_2}, \dots, -\Omega C_{sl_\beta}$ have positive projections onto v and if we move x along the line v , then $\ell(x + tv)$ is an increasing function of $t \in \mathbb{R}$. This proves Theorem 2 in the other direction. \square

The intuition behind this theorem is that with every game between Player i and a member of Γ , Player i receives some information about the direction in which to move their mElo rating in order to increase their log-likelihood value. Once Player i plays enough games with enough members of Γ , they get enough directions such that these directions begin to cancel one another out. And when all these directions have cancelled one another out we have arrived at a maximum likelihood estimator for C_i , given C_Γ .

If we have a population of m players and we introduce a new player. This new player's mElo rating introduces $2k$ new variables into the optimisation problem and thus we are optimising the log-likelihood function over a space with $2km + 2k$ dimensions. Due to the symplectic symmetry of

the log-likelihood function, we can remove $(2k^2 + k)$ dimensions from the domain of the objective function (noting that usually $k \ll m$). Whenever a player is comparable to the existing network, we have information about how far we should move in each of the $2k$ new dimensions. This means that the log-likelihood function will have local maxima in the directions corresponding to the new $2k$ dimensions.

Another way we can think about the problem is that we could resample the data and bootstrap the mElo updates updating our mElo ratings using stochastic gradient descent. In the case that a player was not comparable, their mElo ratings would diverge to positive infinity because there would be at least one direction in which we could not update our mElo rating.

This brings us to the main difference between matrix completion in [15], and the mElo rating system. In the typical matrix completion problem, we just need to know the combinatorial properties of the mask (*i.e.*, who plays who). In the mElo rating system, we have a directed interaction network, where people often only play other players once and never again. If it were the case that for every pair of players, we observed enough games between so that we observe both players win at least one game each, then our problem would reduce to that of [15].

Let M be the total population of all the players. A necessary condition for the advantage matrix to have a unique maximum likelihood estimator, is for every player i in the population to be comparable with the set $M \setminus i$. This follows from Theorem 1. If a player in the population was not comparable to the other players in the population, then there would not be a unique maximum likelihood estimator for C_i , given mElo ratings C_j for all players $j \neq i$, and hence there would not be a unique advantage matrix A .

We now prove that this condition is necessary but not sufficient. Use the cycle graph and the mElo rating system with 2-dimensional mElo rating vectors ($k = 2$) as an example. Imagine, we have a population of four players indexed by A,B,C and D, where Player A cannot directly play D and Player B cannot directly play C. We can demonstrate that the probability that A beats D is under-determined. We provide a visual proof in the diagram in Figure 2. From our proof of Theorem 2, if we know the probability that Player i beats Player j , then we know C_j 's component in the ΩC_j direction but not in any other direction.

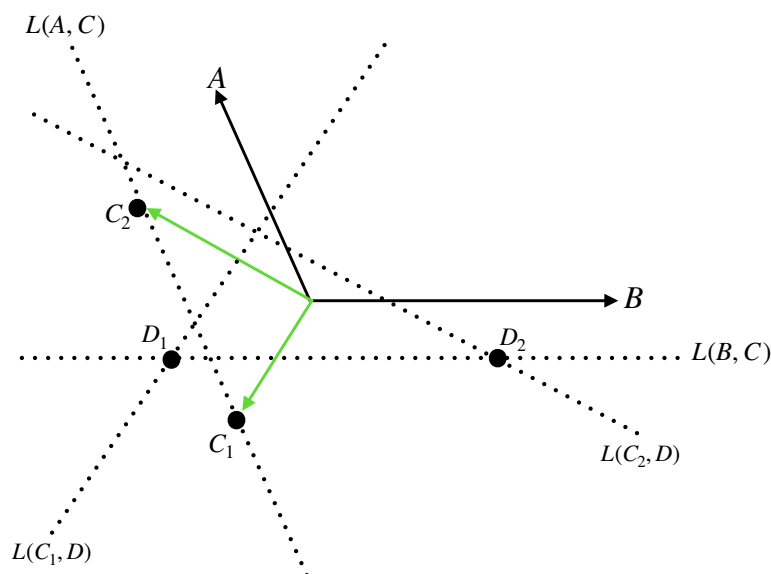


Figure 2. A visual proof that the problem of fitting mElo vectors is under-determined. We fix the values of the mElo vectors A and B , and then we show that the vectors C and D cannot be uniquely determined. Because A can play C , we can observe the probability $P_{A,C}$, hence we know the component of C perpendicular to A . This means that we know that C must lie on the line $L(A,C)$. Likewise, since we also observe the probability $P_{B,D}$, we know that D must lie of the line $L(B,D)$. We can select any point on $L(A,C)$ to be our vector C ; we demonstrate this with two selections C_1 and C_2 . Because we can observe $P_{D,C}$, we can draw the line $L(D,C)$ parallel to our choice of C and where this line intersects $L(B,D)$ is the location of D . Our choices of C_1 and C_2 lead to two different values of D , D_1 and D_2 . Both choices of C and D satisfy all given constraints.

7.1. Sufficient Condition

Now that we have presented a necessary condition for the advantage matrix A to have a unique maximum likelihood estimator, we present a separate condition that is sufficient. From this sufficient condition, we are able to create a polynomial-time algorithm that can check whether the data set corresponds to a unique maximum likelihood estimator for A .

Here, we adapt the matrix completability algorithm from [15] to suit our purposes. The algorithm takes as input, the interaction network for the players in the population. This is a directed graph, where the nodes are the players and there is a directed edge from Player i to Player j if Player i has beaten Player j at least once.

The algorithm works by taking a minimal sub-population of the players such that the advantage matrix associated with this sub-population is uniquely defined. If it is the case that we can observe a clique of $2k + 1$ players, and we have estimates of the probabilities of victory between all pairs of these players, then the advantage matrix (*i.e.*, the maximum of the log-likelihood function) associated with these players is uniquely defined. We assign mElo ratings to these $2k + 1$ players so as to maximise the log-likelihood function.

The algorithm works by iterating through all the players that do not yet have unique mElo ratings. If the algorithm finds a player that is comparable to the players who have been assigned mElo ratings, we then add the player to the population, and recalculate the mElo ratings to maximise this new log-likelihood function. The algorithm then searches for another player that is comparable to this new group, and repeats this process of assigning mElo ratings. If the algorithm cannot, at any stage, find a player that is comparable to its current set of comparable players, then it returns a value of null.

Our algorithm is based on a sufficient condition that may or may not be a necessary condition. It is still an open problem to determine whether there are any datasets that have a unique maximum likelihood estimator but would be assigned a null value by our algorithm. The algorithm is described in pseudocode in Algorithm 1.

Algorithm 1 An to determine whether there exists a unique maximum likelihood estimator for P .

Input: A directed network over m players.
Output: A Boolean that equals True when there exists a unique estimator for the expected payoff matrix.
 Find a clique Γ of $2k + 1$ players
if No clique Γ exists **then**
 return False
else
 stopping search = False
 while stopping search == False **do**
 found comparable player = False
 for $i \in M \setminus \Gamma$ **do**
 if Player i is comparable to Γ **then**
 found comparable player = True
 $\Gamma \leftarrow \Gamma \cup \{i\}$
 end if
 end for
 if found comparable player == false **then**
 stopping search = True
 end if
 end while
 if $\Gamma == M$ **then**
 return True
 else
 return False
 end if
end if

7.2. An Example

In the previous section, we established conditions for a player to be comparable to the other players in a population. Here, we demonstrate this idea using numerical simulation on a toy problem where we have three players P_1, P_2 , and P_3 . They are given the following initial mElo ratings.

$$C_1^{(0)} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, C_2^{(0)} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, C_3^{(0)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

We introduce a new player called P_4 and consider four scenarios in which P_4 wins and loses against the other players. In each of the four scenarios, we plot the relevant vectors used to determine whether P_4 is comparable to the set of other players. The scenarios are

- Scenario 1: P_4 wins against P_1 and P_3 . The log-likelihood function increases in the direction ΩC_1 and ΩC_3 . The log-likelihood will increase in any direction given by a convex combination of these two vectors. Therefore, in this scenario, there is no unique local maximum for C_4 , given C_1 and C_3 .
- Scenario 2: P_4 wins and loses against P_1 . Because P_4 only plays P_1 , the log-likelihood function only changes in the direction ΩC_1 , but this time but the function does not keep increasing as one moves further along ΩC_1 . The information in this game is enough for one to determine the component of C_4 perpendicular to C_1 , but not any of the other components.
- Scenario 3: P_4 wins against P_1, P_2 , and P_3 . In this case we have a log-likelihood function similar to that of Scenario 1. Because ΩC_2 is contained in the convex cone of ΩC_1 and ΩC_3 , the fact that we have observed P_4 winning against P_2 , does not give us very much new information and is not enough to make P_4 comparable to P_1, P_2 , and P_3 .
- Scenario 4: P_4 wins against P_1, P_2 , and P_3 but also loses against P_2 . In this scenario the convex cone of the vectors $\Omega C_1, \Omega C_2, \Omega C_3$, and $-\Omega C_2$ contains \mathbb{R}^2 . We can see that the log-likelihood

function corresponding to this scenario has a single, unique local maximum. In this case, a maximum likelihood estimator for C_4 is uniquely determined from the available data.

The four scenarios are illustrated in Figure 3, which shows the log-likelihood functions. We can see that only Scenario 4 has a well-defined minima showing that in only that case is P_4 comparable to the set $\{P_1, P_2, P_3\}$.

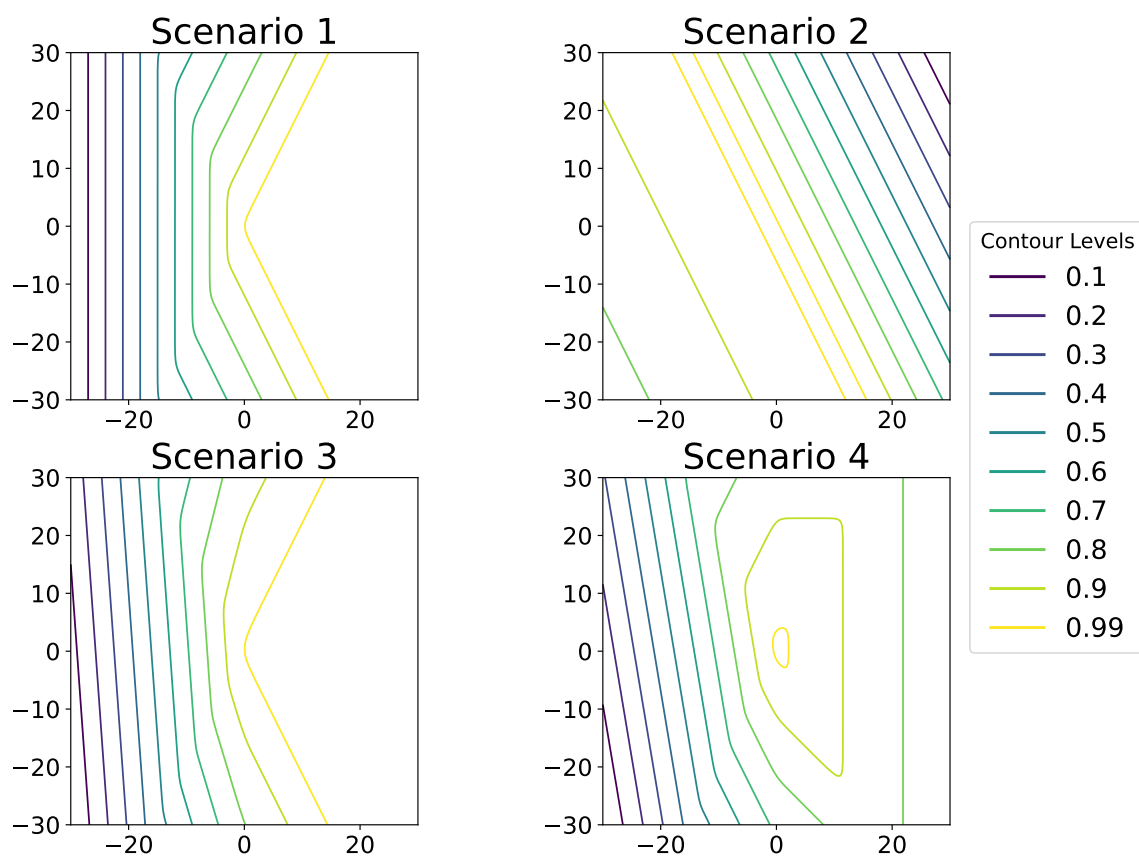


Figure 3. The log-likelihood functions associated with the four scenarios described in Section 7.2. Each figure describes the log-likelihood of the value of C_4 , given fixed values of C_1 , C_2 , and C_3 . We can see that only in Scenario 4 is P_4 comparable to the set $\{P_1, P_2, P_3\}$.

8. Simulation Example

In this section, we will consider a toy problem of three players with three players, called Players P_1 , P_2 , and P_3 , with mElo ratings $C_1^{(t)}$, $C_2^{(t)}$, $C_3^{(t)}$, respectively. It differs from the previous case in that we start with a known expected win/loss matrix between these players:

$$P = \begin{bmatrix} 0.5, 0.2, 0.8 \\ 0.8, 0.5, 0.2 \\ 0.2, 0.8, 0.5 \end{bmatrix}.$$

These three players are given the following (notionally randomised) initial mElo ratings.

$$C_1^{(0)} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, C_2^{(0)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, C_3^{(0)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

These ratings correspond to an expected payoff matrix of

$$\begin{aligned} \hat{P}^{(0)} &= \sigma \left(\begin{bmatrix} C_1^{(0)\top} \\ C_2^{(0)\top} \\ C_3^{(0)\top} \end{bmatrix} \Omega [C_1^{(0)}, C_2^{(0)}, C_3^{(0)}] \right) \\ &= \sigma \left(\begin{bmatrix} 3, 0 \\ 1, 2 \\ -1, -1 \end{bmatrix} \Omega \begin{bmatrix} 3, 1, -1 \\ 0, 2, -1 \end{bmatrix} \right) \\ &= \sigma \left(\begin{bmatrix} 0, & 6, & -3 \\ -6, & 0, & 1 \\ 3, & -1, & 0 \end{bmatrix} \right) \\ &\simeq \begin{bmatrix} 0.5, & 0.99752, & 0.04742 \\ 0.00247, & 0.5, & 0.73105, \\ 0.95257, & 0.26894, & 0.5 \end{bmatrix}. \end{aligned}$$

Note that this quite a long way from the true matrix.

We now show how one can update the mElo ratings after game outcomes to better approximate P .

At each iteration of our simulation, we pick two players randomly from our population. For example, we start by picking Players 2 and 3. The true probability that Player 2 beats Player 3 is equal to $P_{2,3} = 0.2$. In our simulation, we have Player 2 win against Player 3 (even though it was unlikely).

We update the mElo ratings of players 2 and 3 according to equations Equation (4), and Equation (5) respectively, as follows (taking our learning rate $\eta = 0.1$)

- $i = 2, j = 3$;
- $S_i = 1$, (Player i beat Player j);
- $t = 0$, (we are updating our initial ratings);
- $C_i^{(t)} = C_2^{(0)} = [1, 2]^\top$;
- $C_j^{(t)} = C_3^{(0)} = [-1, -1]^\top$ (from our initial ratings); and
- $\hat{P}_{i,j}^{(t)} = \hat{P}_{2,3}^{(0)} \approx 0.73105$ (from our previous calculation of $\hat{P}^{(0)}$).

Substituting our values into Equation (4), we obtain the update

$$C_2^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \frac{0.1(1 - 0.73105)\Omega \begin{bmatrix} -1 \\ -1 \end{bmatrix}}{\sqrt{|[1, 2]|^2 + |[-1, -1]|^2}}$$

$$\simeq \begin{bmatrix} 0.98983 \\ 2.01016 \end{bmatrix}.$$

Similarly, we can use Equation (5) to update Player 3's mElo rating

$$C_3^{(1)} \simeq \begin{bmatrix} -1.02033 \\ -0.98983 \end{bmatrix}.$$

Since Player 1 did not play against anybody in the first timestep, we have $C_1^{(1)} = C_1^{(0)}$.

We continue our simulation for a total of 4000 games. At each timestep, we select two players i and j uniformly at random, have Player i beat Player j with probability given by $P_{i,j}$, and update the player's mElo ratings using equations Equation (4) and Equation (5). The results of our simulation are shown in Figure 4.

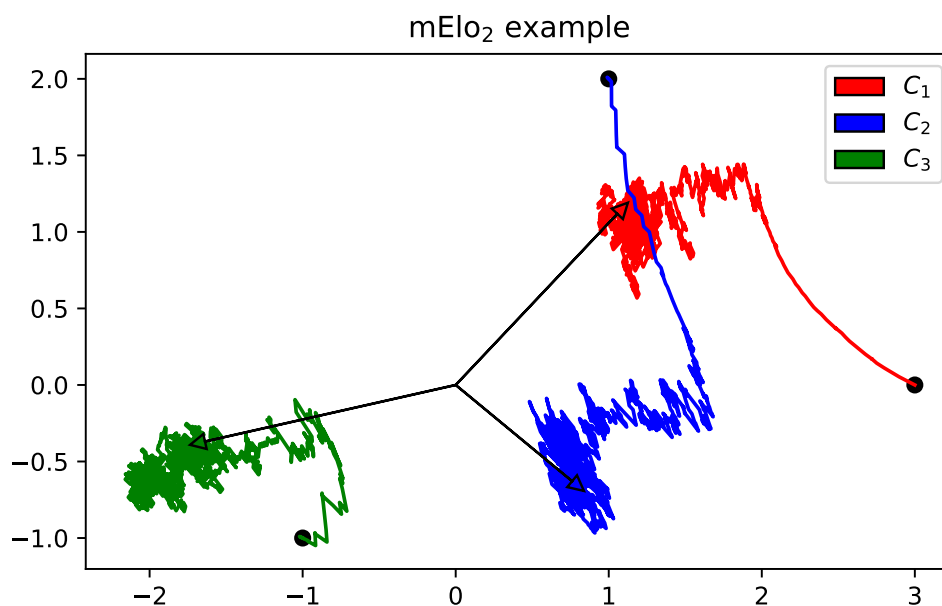


Figure 4. Plot of the trajectories of the players' mElo ratings over the course of the simulation The players initial mElo ratings are.

In Figure 4, we show the changes in mElo ratings for all 3 players. The first and second components of each player's mElo rating vector is shown on the horizontal and vertical axis respectively. The change of each player's mElo ratings are shown as three continuous lines on the plane. We can see the one visible endpoint of each line corresponds to each of the players' initial mElo ratings, the arrows represent the values of the players' mElo ratings at the end of the simulation.

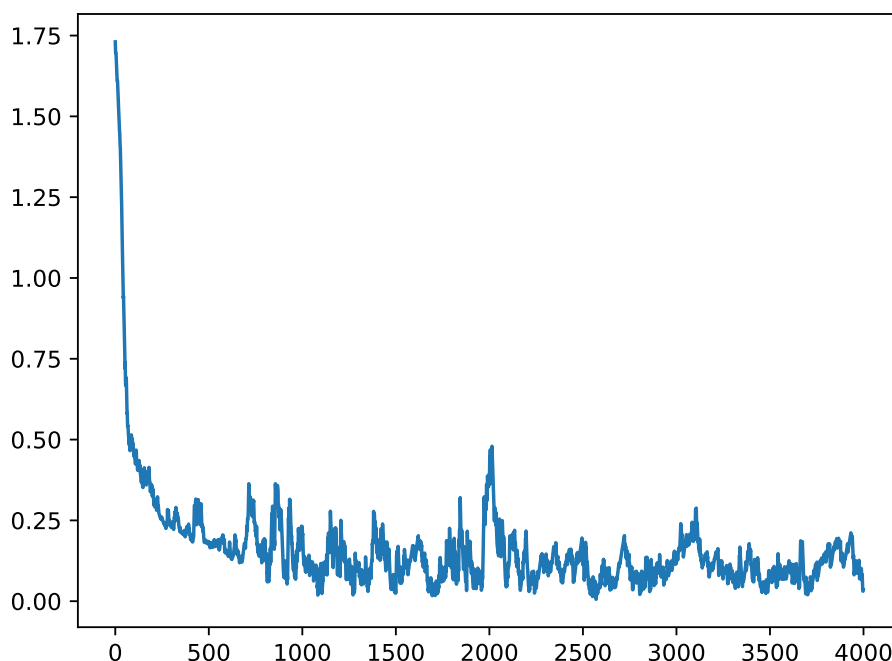


Figure 5. Diagram of the Frobenius norm between the empirical expected payoff matrix and the true expected payoff matrix at each step during the simulation. We can see that the mElo rating system does approximate the true expected payoff matrix as required, but there will always be noise present at a certain level.

In Figure 5, we show the Frobenius distance between the approximate expected payoff matrix \hat{P} and the true expected payoff matrix P . We can see that the mElo rating system is indeed approximating the true expected payoff matrix as required. We can also see that after approximately 1000 iterations, the mElo ratings stop improving. After this point, the mElo ratings just move around a set of mElo ratings that correspond to the true expected payoff matrix, because the step sizes after each iteration are bounded from above and below within a neighbourhood of the true values, the mElo rating system cannot obtain more precise results. Hence, the noisy estimates of P .

If we had used a learning rate less than $\eta = 0.1$, then it would take more iterations for the mElo ratings to find an approximation to P , but the approximation of P would be more precise. There is an inherent trade-off when selecting a learning rate for the mElo rating system. A smaller value of η leads to more precise estimates but slower convergence, and a larger value of η leads to faster convergence but less precise estimates. The optimal selection of η is an important problem, especially in situations where the true expected payoff matrix changes over time, and there is limited data available.

However, the variations can actually be a desirable property in the analysis of games because players' abilities in a sport (for instance) vary over time, and hence an iterative algorithm needs to adapt its estimates. Some noise is inevitable. In ML applications, we might wish to use mElo with a reducing learning rate schedule (much as is used in many other optimisation contexts, such as the temperature schedule used in simulated annealing) in order to converge to fixed estimates within some time threshold.

In our simulation, we ensure that all three players can play one another an unlimited number of times. This ensures that the games scheduled satisfy the properties in Section 6 and the expected payoff matrix estimated in this process is unique.

Even though the expected payoff matrix obtained was unique, the matrix C of mElo ratings was not. Since we randomly initialised our ratings at $C_1^{(0)} = [3, 0]^T$, $C_2^{(0)} = [1, 2]^T$, and $C_3^{(0)} = [-1, -1]^T$, we obtained a matrix of mElo ratings

$$C_{final} = \begin{bmatrix} 1.05632047, & 0.76079239, & -1.63813171 \\ 1.11819387, & -0.62869858, & -0.36933412 \end{bmatrix}$$

If we had started with the same mElo ratings but rotated θ degrees anti-clockwise, then we would end up with a matrix C in which the columns were also rotated θ degrees anti-clockwise. It is important to remember that both C matrices would be equally correct.

9. Advice to the Practitioner

This section summarises the results from this paper as a guide to the practitioner, where we provide a few instructions to the practitioner so they can be more sure that they are using the mElo rating system correctly.

When using the mElo rating system, we recommend that the practitioner does the following:

- Before fitting mElo ratings to the data use the list of games to determine which players are comparable with which other sets of players, using the sufficient condition given in Section 7. Only proceed if the condition is satisfied.
- When initialising the mElo ratings of each player, make sure they are initialised in such a way that the matrix C of ratings is of full rank. This can be done by randomly assigning players their initial ratings. If the initial ratings are sampled from a continuous probability distribution, then C will be of full rank with probability one.
- When updating mElo ratings, make sure to use the normalised version of the mElo update algorithm to ensure that the ratings stay robust.
- When assessing results (for instance from simulations), do not expect ratings vectors to match a preassigned set of "ground-truth" vectors. Assess the accuracy of the advantage or probability matrices.

10. Conclusions and Future Work

We hope that the results presented here will help the data science and ML communities use the mElo rating system correctly, when it is applicable. Further details of the mElo rating system can be found in [12]. Despite the limitations documented here, the mElo system allows the analysis of game outcomes (and ML comparisons) in a much richer manner than simple Elo, and it shows much promise.

Moreover, we suspect that the desire for such richness will lead many future authors to reach for systems that extend Elo into more complex vector spaces, and hence developing our theoretical understanding of such systems is important. The properties described here revolve around the specific mElo proposal, but the underlying understanding of how symmetries affect a rating system is applicable to almost any such system.

In the future, we aim to extend this understanding in several ways. Both the Elo and mElo rating systems are related to particle filters and Hidden Markov Models in that they are designed to track the movements of a partially observed stochastic system (here an expected payoff matrix) that is continuously changing over time. The time series aspect of the mElo rating system has not yet received much study. If the sampling rate multiplied by the learning rate is less than the speed at which the true parameter values are moving, then we hypothesise that the rating system will not work. In the future, we intend to derive mathematical conditions under which one can detect this phenomenon and know that the mElo rating system is applicable.

In addition, because mElo has a larger parameter space than Elo, it will require more data to converge to an optimum. In future work, we intend to place realistic bounds on how much data of

each type is required to establish a certain level of confidence in the results of the mElo rating system for a given value of k .

11. Conclusions

This section is not mandatory, but can be added to the manuscript if the discussion is unusually long or complex.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.”, please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code used to create simulated data for this paper can be found at <https://github.com/UofA-AdamHamilton/mEloSimulations>

References

1. Elo, A. *The Rating of Chess Players, Past and Present*; Ishi Press International, 1978.
2. Tenkanen, S. Rating National Hockey League teams: the predictive power of Elo rating models in ice hockey. Bachelor’s thesis, Aalto University, Espoo Finland, 2019. Available at <https://aaltodoc.aalto.fi/items/d163b9aa-bcdc-4d64-90d0-6efee35a3cf5>.
3. Thorrez, C. EsportsBench: A Collection of Datasets for Benchmarking Rating Systems in Esports, 2024. <https://github.com/cthorez/esports-bench>.
4. Mangaroska, K.; Vesin, B.; Giannakos, M. Elo-rating method: Towards adaptive assessment in e-learning. In Proceedings of the 2019 IEEE 19th international conference on advanced learning technologies (ICALT). IEEE, 2019, Vol. 2161, pp. 380–382.
5. Vaughan Williams, L.; Liu, C.; Dixon, L.; Gerrard, H. How well do Elo-based ratings predict professional tennis matches? *Journal of Quantitative Analysis in Sports* **2021**, *17*, 91–105.
6. Balduzzi, D.; Tuyls, K.; Perolat, J.; Graepel, T. Re-evaluating evaluation. In Proceedings of the Advances in Neural Information Processing Systems; Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; Garnett, R., Eds. Curran Associates, Inc., 2018, Vol. 31.
7. Czarnecki, W.M.; Gidel, G.; Tracey, B.; Tuyls, K.; Omidshafiei, S.; Balduzzi, D.; Jaderberg, M. Real World Games Look like Spinning Tops. In Proceedings of the Proceedings of the 34th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2020; NIPS’20.
8. Tang, S.; Wang, Y.; Jin, C. Is Elo Rating Reliable? A Study Under Model Misspecification, 2025, [\[arXiv:cs.LG/2502.10985\]](https://arxiv.org/abs/2502.10985).
9. Lazaridis, D. mElo, 2020.
10. Firth, D. Bradley-Terry Models in R. *Journal of Statistical Software* **2005**, *12*, 1–12. <https://doi.org/10.18637/jss.v012.i01>.
11. Tran, N.M. Pairwise ranking: choice of method can produce arbitrarily different rank order. *Linear Algebra and its Applications* **2013**, *438*, 1012–1024.
12. Hamilton, A. A Multi-Dimensional Extension of the Elo Rating System, 2023. PhD Thesis, University of Adelaide.
13. Bump, D. *Lie Groups*, 2nd ed. 2013. ed.; Graduate Texts in Mathematics, 225, Springer New York: New York, NY, 2013.
14. Zhang, Y.; Roughan, M.; Willinger, W.; Qiu, L. Spatio-temporal compressive sensing and internet traffic matrices. In Proceedings of the Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, New York, NY, USA, 2009; SIGCOMM ’09, p. 267–278. <https://doi.org/10.1145/1592568.1592600>.
15. Király, F.; Tomioka, R. A combinatorial algebraic approach for the identifiability of low-rank matrix completion, 2012. arXiv preprint arXiv:1206.6470.

16. Király, F.J.; Theran, L.; Tomioka, R. The algebraic combinatorial approach for low-rank matrix completion. *J. Mach. Learn. Res.* **2015**, *16*, 1391–1436.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.