A Rank Aggregation Algorithm for Performance Evaluation in Modern Sports Medicine with NMR-based Metabolomics

V. Vigneron and H. Maaref

Univ. Evry, Université Paris-Saclay, IBISC EA 4526, Evry, France

Keywords: Deep Learning, Pooling Function, Rank Aggregation, LBP, Segmentation, Contour Extraction.

Abstract: In most research studies, much of the gathered information is qualitative in nature. This article focuses on items for which there are multiple rankings that should be optimally combined. More specifically, it describes a supervised stochastic approach, driven by a Boltzmann machine capable of ranking elements related to each other by order of importance. Unlike classic statistical ranking techniques, the algorithm does not need a voting rule for decision-making. The experimental results indicate that the proposed model outperforms two reference rank aggregation algorithms, ELECTRE IV and VIKOR, and it behaves more stable when encountering noisy data.

1 INTRODUCTION AND RELATED WORKS

In the last decades, the field of multiple criteria decision-making (MCDM) has received considerable attention in engineering, sciences, and humanities as they are extremely efficient in situations where policymakers need to decide priorities (Yazdani et al., 2017). There are optimal resolution procedures like linear programming or nonlinear optimization for solving problems governed by single criteria. But real-life situations demand the evaluation of a set of alternatives against multiple criteria and are typically structured as MCDM problems (Thakkar, 2021; Rahman et al., 2017). When a decision needs to be made - like choosing a movie, buying a car, selecting a stock portfolio, etc. - the choice should not be random or biased by someone’s suggestion. MCDM algorithms often produce conflicting results when compared together because of the choice of the function to optimize. This comes from the unavoidable trade-off between conflicting objectives as well as constraints. As a result, the optimal solution is not unique and corresponds to a so-called Pareto solution (Freund and Williamson, 2015).

On the opposite, learning to rank (LTR) is a class of approaches that apply supervised machine learning (ML) to resolve ranking problems. The training data for a LTR model consists of a list of samples and a "ground truth" score for each of those samples, manually labeled by experts, see (Li et al., 2017). The set of ranked data ("ground truth") becomes the data set that the system "trains" by minimizing some loss function to learn how best to rank automatically these items (Chaudhuri and Tewari, 2015).

The most common application of LTR is search engine ranking (Sharma et al., 2022). We propose to use them in the context of sports medicine because performance evaluation is a kind of fuzzy task. Existing LTR algorithms may be divided into 3 main classes: (a) pointwise methods which reduce the rating on each item to regression or classification (Blackburn and Ukhov, 2013) (b) pairwise methods which essentially formulate ranking on each document pair as a classification problem (Burges et al., 2007) (c) list-wise methods which optimize a measure-specific loss function, on all available items. See Chavhan et al. (Chavhan et al., 2021) for a review.

The pros and cons of using LTR vs. MCDM are (a) LTR is essentially a black box in terms of explainability. It’s hard to explain what exact effect specific inputs have on the outcome (b) LTR is
greedy (c) result relevance is metric-dependent. This work presents a new extension of the LTR based on a continuous restricted Boltzmann machine (CRBM) (Hinton, 2002). The CRBM is a generative stochastic artificial neural networks (ANN) that can learn a probability distribution over all possible permutations of its set of inputs. CRBMs have found applications in dimensionality reduction (Vrábel et al., 2020), classification (Yin et al., 2018) and collaborative filtering (Verma et al., 2019). They are excellent generative learning models for latent space extraction. Specifically, they can be trained into excellent ranking devices because of their flexible loss function and the associative memory captured in the transfer matrix \( W \) between the visible and the hidden layers, which is a promising advantage over other standard MCDM algorithms.

The paper is organized as follows: section 2 provides a detailed description of the problem and the metrics used for measuring aggregation of ranks. Section 3 presents the generative model. The experimental results are presented in Section 4. The last section concludes and outlines the way for future work.

**Notations.** Throughout this paper small Latin letters \( a, b, \ldots \) represent integers. Small bold letters \( \mathbf{a}, \mathbf{b} \) are put for vectors, and capital letters \( A, B \) for matrices or tensors depending on the context. The dot product between two vectors is denoted \( \mathbf{a} \cdot \mathbf{b} \). We denote by \( |\mathbf{a}| = \sqrt{\mathbf{a} \cdot \mathbf{a}} \), the \( l_2 \) norm of a vector. \( X_1, \ldots, X_n \) are non ordered variates, \( x_1, \ldots, x_n \) non ordered observations. "Ordered statistics" means either \( p_{(1)} \leq \ldots \leq p_{(n)} \) (ordered variates) or \( p_{(1)} \leq \ldots \leq p_{(n)} \) (ordered observations). The \( p_{(i)} \) are necessarily dependent because of the inequality relations among them.

**Definition 1** ((Savage, 1956)). The rank order corresponding to the \( n \) distinct numbers \( x_1, \ldots, x_n \) is the vector \( t = (t_1, \ldots, t_n)^T \) where \( t_i \) is the number of \( x_j \)’s \( \leq x_i \) and \( i \neq j \).

The rank order \( t \) is always unambiguously defined as a permutation of the first \( n \) integers.

## 2 GENERAL FRAMEWORK

### 2.1 Rank-Aggregation

Let \( A = \{a_1, a_2, \ldots, a_n\} \) be a set of alternatives, candidates, individuals, etc. with cardinality \( |A| = n \) and let \( V \) be a set of voters, judges, criteria, etc. with \( |V| = m \). The data is collected in a \((n \times m)\) table \( T \) of general term \( t_{ij} \) crossing the sets \( A \) and \( V \) (Figure 1). \( t_{ij} \) can be marks \((t_{ij} \in \mathbb{N})\), value scales \((t_{ij} \in \mathbb{R})\), ranks (such that a voter can give \textit{ex-aequo} positions) or binary numbers \((t_{ij} \in \{0, 1\}\) such as opinion yes/no). \( T \) represents the ranking of the \( n \) alternatives under the form (see (Brüggemann and Patil, 2011) for a reminder on rank-aggregation). For ease of writing, in the following, \( t_{ij} = t_{ij}^{(1)} \).

![Figure 1: The data are collected in a \((n \times m)\) table \( T \).](image)

Solving a rank-aggregation problem means finding a distribution of values \( x^{*} \) attributed by a virtual judge to the \( n \) alternatives by minimizing the disagreements of opinions between the \( m \) judges (Benson, 2016), i.e.

\[
x^{*} = \arg \min_{x} \sum_{k=1}^{m} d(t, t^{(k)}), \quad \text{s.t.} \quad t \geq 0,
\]

where \( d(t, t^{(k)}) \) is a metric measuring the proximity between \( t \) and \( t^{(k)} \), chosen a priori, and \( t^{(k)} \) is the \( k \)th column of the table \( T \). Depending on the properties of \( d(\cdot) \), we will deal with a nonlinear optimization program with an explicit or implicit solution.

One could also stand the dual problem of the previous one, i.e., is there a distribution of rankings/marks that the \( m \) voters could have attributed to a virtual alternative ‘\( a \’ \ summarizing the behavior of the set of individuals \( A \) (Yadav and Kumar, 2015)? The first problem is linked to the idea of aggregating points of view, and the second to the concept of summarizing behaviors.
2.2 Explicit or Implicit Resolution

Eq. (1) defines a nonlinear optimization program whose solution is \( x^* \) (Yadav and Kumar, 2015). The distance \( d(t^{(k)}, t^{(k')}) \) between the ranking of voters \( k \) and \( k' \) can be chosen for instance as the Euclidean distance \( \sum_{i=1}^{n} (t_{ik} - t_{ik'})^2 \), the disagreement distance (Condorcet) \( \sum_{i=1}^{n} \text{sgn}(t_{ik} - t_{ik'}) \) or the order disagreement distance \( \sum \sum_{ij} |y_{ij}^{(k)} - y_{ij}^{(k')}| \) as \( t^{(k)} \) can be replaced by its permutation matrix \( Y^{(k)} \) (Figure 1). In the latter, \( y_{ij}^{(k)} = \mathbb{1}_{i < j} \) denotes the indicator matrix for which \( y_{ij}^{(k)} = 1 \) if the rank of the alternative \( a_i \) is less than the alternative \( a_j \) and 0 otherwise (Gehrlein and Lepelley, 2011). Note that \( y_{ii}^{(k)} = 0 \) and \( y_{ij}^{(k)} = 0 \) if \( i \) and \( j \) are ex-aequos.

In using matrix \( Y^{(k)} \), \( \frac{1}{2} \sum \sum_{ij} |y_{ij}^{(k)} - y_{ij}^{(k')}| = \frac{1}{2} \sum \sum_{ij} (y_{ij}^{(k)} - y_{ij}^{(k')})^2 \) since the expressions \( |y_{ij}^{(k)} - y_{ij}^{(k')}| \) are 0 or 1.

As \( y_{ij}^{(k)} = y_{ij}^{(k')} \), \( y_{ij}^{(k)} = 0 \) or 1, the function associated to order disagreement distance \( d \) is given by

\[
1 \left[ \sum_{i=1}^{n} \sum_{j=1}^{m} m_{ij}^{(k)} + \sum_{i=1}^{n} \sum_{j=1}^{m} \left( m_{ij}^{(k)} - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ij}^{(k)} \right) \right].
\]

Let \( a_{ij}^{(k)} = \sum_{i=1}^{n} y_{ij}^{(k)} \) the total number of voters preferring alternative \( a_i \) to \( a_j \) and define a matrix \( A = \{ a_{ij} \} \), summing the \( m \) matrices \( Y^{(k)} \) associated to the rankings \( t^{(k)} \) of the voters \( V^{(k)} \), Eq. (2) becomes:

\[
\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} m_{ij}^{(k)} + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^{(k)} - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^{(k)} y_{ij}^{(k)}.
\]

Finally, the search for a total order given by a matrix \( Y \) is the solution of the linear program

\[
\text{max } y \left( \sum_{i=1}^{n} \sum_{j=1}^{m} m_{ij}^{(k)} + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^{(k)} - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^{(k)} y_{ij}^{(k)} \right)
\]

s.t. \( a_{ij}^{(k)} = \sum_{k=1}^{p} y_{ij}^{(k)} \), \( y_{ij} = 1, i < j, y_{ii} = 0 \) \( y_{ij} + y_{ji} - y_{ik} \leq 1, i \neq j \neq k, y_{ij} \in \{0, 1\} \).

If the chi-2 metric is chosen, then the dependent variables cannot be separated as in Eq. (4): The resolution follows an implicit gradient-descent procedure as in (Vigneron and Tomazeli Duarte, 2018). Section 2.3 detailed how chi-2 distance is used for solving aggregation programs.

2.3 Chi-2 Metric

The distance between \( x \) and \( t^{(k)} \) is given by

\[
d(x, t^{(k)}) = \sum_{i} \frac{1}{f_i} \left( \frac{f_{ix} - f_{ik}}{f_{ik}} \right)^2,
\]

where

\[
f_i = \sum_{k} t_{ik} + x_i, \quad f_{ix} = \sum_{k} t_{ix} + x_i, \quad f_{ik} = \sum_{k} t_{ik} + x_i,
\]

\[
f_{ik} = \sum_{k} t_{ik} + x_i.
\]

Let \( n^2 = \sum_{k} t_{ik}, n^2 \tilde{x} = \sum_{k} x_i t_k = \sum t_{ik} \) and \( t_k = \sum t_{ik} \). Then, after some calculus, the optimal ranking \( x^* \) minimizes \( \sum_{i} d(x, t^{(k)}) \):

\[
n \sum_{i} \sum_{k} \frac{f_i + x_i}{t_k + x_i} \left( \frac{x_i}{n^2 \tilde{x}} \right)^2 = \left( n^2 - 2 \sum_{i} \left( x_i - \sqrt{x_i^2 - t_k} \right)^2 \right), \quad \text{assuming } t_k > x_i \sqrt{x_i^2 - t_k}.
\]

According to Eq. (7), the ranking is performed on the row profiles or column-profiles of the matrix \( T \) (see Fig. 1), each row being weighted by \( \sqrt{t_k} \).

So it is equivalent to compute profile matrix \( C \) whose entry is \( \frac{t_k}{\sqrt{t_k}} \), to consider the Euclidean distance between its rows and to heighten each row by \( \sqrt{t_k} \). A remark has to be made at this stage: two alternatives will be close if a large proportion of judges choose them simultaneously. For example, if there is a considerable amount of individuals chosen preferably by judges ‘A’ and ‘B,’ then we will say that judges A and B are close and that they “attract” each other. Eq. (7) is the well-known expression used in testing for independence in contingency tables.

Eq. (7) derives from the Bhattacharyya directed divergence between two discrete probability distributions \( P = \{ p_i \} \) and \( Q = \{ q_i \} \) defined as

\[
BD = -\ln(\sum_{i} \sqrt{p_i q_i}) \quad \text{(Nielsen, 2022)}\text{ if } p_i = \frac{x_i}{\sqrt{t_i}}, q_i = \frac{y_i}{\sqrt{t_i}}.
\]

Note that \( n^2 \) is useless in the ratio and will be removed in the entries of the continuous restricted Boltzmann machine. Implicit methods are natural for LTR algorithms that are usually fed by an incoming data stream, \( n \) constantly varying. Section 3 proposes a learning model in which the rank probabilities take the form of a Boltzmann distribution.
3 METHODOLOGY

3.1 Continuous Restricted Boltzmann Machine

Chen and Murray proposed another Boltzmann machine (BM) approach with continuous neuron in (Chen and Murray, 2003): the CRBM, a restricted Boltzmann machine using the neuron structure depicted in figure 2. In the CRBM, the activation function is unique for each neuron and given by:

\[ s_j = \phi_j(x_j) = \theta_L + (\theta_U - \theta_L) \frac{1}{1 + \exp(-a_j x_j)} \] (8)

where \( \theta_L \) and \( \theta_U \) are, respectively, the function’s lower and upper bounds. \( a_j \) is a slope parameter of \( \phi_j(.) \). The continuous behavior for the hidden units allows us to capture more information than binary units.

![Diagram of Continuous Restricted Boltzmann Machine](image)

We note \( W \in \mathbb{R}^{(m \times l)} \) the transfer matrix between the two layers and \( \xi_v \) and \( \xi_h \) the bias vectors of, respectively, the visible layer and the hidden layer. The energy function of the CRBM is:

\[ E(v,h) = -v^T Wh - v^T \xi_v - h^T \xi_h + \sum_j \frac{1}{m} \int_0^1 \phi^{-1}(\mu) \text{d} \mu \] (9)

with \( \phi^{-1}(\cdot) \) the inverse of the activation for a coefficient slope \( \alpha_i = 1 \). The energy \( E(s) \) of a CRBM is associated with the joint probability of the state of the neurons \( P_{\text{CRBM}}(s) \) defined as

\[ P_{\text{CRBM}}(s) = \frac{1}{Z} \exp(-E(s)), \] (10)

where \( Z \) is a marginalization constant. Training a CRBM is performed in minimizing the energy function Eq. (9), which itself requires sampling the hidden units.

The CRBM training uses the contrastive divergence algorithm (see (Hinton, 2012)). The training set \( D \) is composed of \( n \) observations used to find the best set of parameters \( \mathcal{F} = \{ W, \xi \} \), \( \xi \) regrouping visible and hidden bias vectors.

Minimizing directly the joint log-likelihood \( \sum_{t=1}^{n} \log P_{\text{CRBM}}(v^t) \) to update the parameters is difficult due to the presence of the constant \( Z \). Then it is replaced by the minimization of the contrastive divergence (MCD) (Hinton, 2002) that minimizes the contrast \( D \) between two successive Kullback-Leibler (KL)-divergences:

\[ D = KL(P^0(v), P^\infty(v)) - KL(P^1(v), P^\infty(v)), \] (11)

where \( P^0(v), P^\infty(v), P^1(v) \) are the distribution function of the visible units over respectively the training set, the equilibrium state and after \( q \) steps of Gibbs sampling (Hinton, 2012) (Fig. 3).

![Figure 3: An intuitive idea is to minimize the KL divergence between \( P^0(v)P^\infty(v) \). But \( P^\infty(v) \) is intractable. We prefer to minimize \( D \). If \( D = 0 \), then \( P^0(v) = P^1(v) \) and then : \( P^0(v) = P^\infty(v) \).](image)

An important observation is that any linear combination of measures of discrepancy with positive coefficients is also a measure of discrepancy.

\[ KL(P^0(v), P^\infty(v)) - KL(P^1(v), P^\infty(v)) + \lambda (BD^0(v) - BD^1(v)), \] (12)

with \( \lambda \) a regularization parameter. And thus, Eq. (12) can be used as a measure of discrepancy.

In particular, the observations are normalized: \( v = (\frac{x^1_1}{\tilde{x}^1_1}, \ldots, \frac{x^m_l}{\tilde{x}^m_l})^T \) (see section 2.3).

In the next section, a CRBM driven by the loss function (12) ranks rugby players according to their performances measured by metabolomics.
Endurance is a widely practiced sporting activity, from novice to champion. It is defined as maintaining an effort for a prolonged period. This effort originates from significant physiological and metabolic stress leading to organism adaptations. If this effort is too great, it can cause metabolic and locomotor disorders. The objective is to optimize training methods that will protect the health of athletes, young or old, efficient or less efficient. We adopt an integrative approach that simultaneously studies the physiological responses to exercise and the molecular and metabolic signals. From multivariate statistical analysis of biofluids such as urine, serum, plasma, saliva, sweat, etc., it is possible to generate metabolomic profiles or biomarkers. See Khoramipour et al. (2022) for a review of metabolomics practice in sports medicine.

Since the 00’s metabolomics investigates quantitatively the metabolome of living systems in response to pathophysiological stimuli or genetic modification (Amara et al., 2022).

Nuclear magnetic resonance (NMR) is traditionally used to elucidate molecular structures. It takes advantage of the energy transition of nuclear spins in a strong magnetic field to identify and elucidate the structure of organic molecules and specific metabolites. Metabolites are intermediate organic compounds resulting from metabolism. To understand the metabolic changes induced by endurance exercise and training of rugby players according to the intensity and duration of the activity, we study the physiological modulation of rugby players according to their positions.

The study focuses on the activity variability between the forwards - more intense and intermittent efforts - and the rears - greater distance covered, more running, more rest time, etc. (Paul et al., 2022). It aims to answer whether, during matches, (a) the urinary metabolites are identical before and after 80 minutes of a match? (b) this metabolomic modulation is of the same order depending on the player’s position?

No study to date has investigated how to predict physiological exertion in rugby or how to classify a player according to its physiological parameters.

The experimental protocol is as follows: the urine of 80 players (40 forwards and 40 rears) is analyzed by NMR to identify the metabolites present in the two situations described above. The variations in metabolism explain the variations in physiological parameters as a function of the time and position factors. NMR spectra contain more than 10,000 values. See Fig 4 for an example of NMR spectrum.

19 variables represent the physiological variables, among which: forward/backward position of the player during the match, body mass index, experience, playing time, distance covered on the playground but also plasmatic metabolite rates in phenylalanine, tyrosine, glucose, creatinine, β-hydroxybutyrate, lactate, pyruvate, N-acetyl glycoprotein, lipids (Table 1). This set of variables constitutes the criteria that are used to rank the rugby players.

There are 70 samples in the training set and 10 in the test set. The structure of the CRBM is: 19 visible and 3 hidden units. Due to the small training and test data, we did not divide the data into mini-batches during the experiment. All the data were divided into eight groups for the seven-fold cross-validation method. Seven groups were selected as the training set each time, and the remaining group was the test set. This process was repeated until each group became a test set. The number of iterations was 200 for each CRBM. For training, the CRBMs were initialized with small random weights and zero bias parameters. The learning rate was $\eta = 0.1$ when training with CD in Eq. (12) and $\lambda = 0.005$. CRBMs models
Table 1: Data description. Notice that variable 17 precises the position of the players. Forwards: pillars, hooker, 2nd lines. Backward: 3rd lines, 9 and 10, backs.

<table>
<thead>
<tr>
<th>Variables</th>
<th>type (if real: ( \mu \pm \text{std.err.} ))</th>
<th>physiological signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 term</td>
<td>binary</td>
<td>1st term=1, 2nd term=0</td>
</tr>
<tr>
<td>P</td>
<td>binary</td>
<td>position forward=1, backward=0</td>
</tr>
<tr>
<td>A</td>
<td>28.4 ± 0.97</td>
<td>Age</td>
</tr>
<tr>
<td>H</td>
<td>179.95 ± 1.0</td>
<td>height</td>
</tr>
<tr>
<td>W</td>
<td>89.75 ± 2.81</td>
<td>weight</td>
</tr>
<tr>
<td>BMI</td>
<td>27.65 ± 0.756</td>
<td>body-mass index</td>
</tr>
<tr>
<td>X</td>
<td>41.3 ± 1.01</td>
<td>game experience</td>
</tr>
<tr>
<td>Ty</td>
<td>143.9 ± 1.89</td>
<td>tyrosine</td>
</tr>
<tr>
<td>Glu</td>
<td>269.5 ± 2.59</td>
<td>glucose</td>
</tr>
<tr>
<td>( \beta )</td>
<td>414.1 ± 3.22</td>
<td>( \beta )-hydroxybutyrate</td>
</tr>
<tr>
<td>Creat</td>
<td>6.57 ± 0.22</td>
<td>creatinine</td>
</tr>
<tr>
<td>Gly</td>
<td>5069 ± 11.25</td>
<td>glycoprotein</td>
</tr>
<tr>
<td>D</td>
<td>2.238 ± 7.48</td>
<td>distance covered on playground</td>
</tr>
<tr>
<td>L</td>
<td>51.6 ± 1.13</td>
<td>lipids</td>
</tr>
<tr>
<td>R</td>
<td>( {1,2,3} )</td>
<td>rolling position</td>
</tr>
<tr>
<td>PHC0</td>
<td>-86.98 ± 15.73</td>
<td>phenylalanine 0</td>
</tr>
<tr>
<td>PHC1</td>
<td>23.36 ± 9.19</td>
<td>phenylalanine 1</td>
</tr>
<tr>
<td>SNR</td>
<td>41.03 ± 3.56</td>
<td>signal to noise ratio</td>
</tr>
</tbody>
</table>

Table 2: Test results comparing CRBM with classic MCDMs algorithms VIKOR and ELECTRE IV. Float numbers are issued from the models and the ranks in sorting these numbers.

<table>
<thead>
<tr>
<th>Players</th>
<th>VIKOR</th>
<th>ELECTRE IV</th>
<th>CRBM+Bhattacharyya</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.608</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5.123</td>
<td>1</td>
<td>6.324</td>
</tr>
<tr>
<td>3</td>
<td>4.639</td>
<td>6</td>
<td>5.094</td>
</tr>
<tr>
<td>4</td>
<td>5.378</td>
<td>8</td>
<td>4.923</td>
</tr>
<tr>
<td>5</td>
<td>5.811</td>
<td>3</td>
<td>6.147</td>
</tr>
<tr>
<td>6</td>
<td>4.033</td>
<td>2</td>
<td>3.401</td>
</tr>
<tr>
<td>7</td>
<td>3.468</td>
<td>4</td>
<td>3.567</td>
</tr>
<tr>
<td>8</td>
<td>4.254</td>
<td>10</td>
<td>3.411</td>
</tr>
<tr>
<td>9</td>
<td>5.6220</td>
<td>9</td>
<td>6.2400</td>
</tr>
<tr>
<td>10</td>
<td>5.8110</td>
<td>5</td>
<td>6.3240</td>
</tr>
</tbody>
</table>

The results obtained with VIKOR, ELECTRE IV, and the CRBM are gathered in Table 2 for comparison. For each MCDM algorithm, the first column measures the discrepancy of the model Eq. (12) and the second column the rank of the rugby player (in bold). They provide an interpretation that the CRBM method is closer to the actual results and far from the ELECTRE IV method (Roy, 1985). VIKOR is a MCDMs, ranking preferences among a set of alternatives in the presence of conflicting criteria under the concept of group regrets (Guiwu et al., 2020). ELECTRE IV assumes that all requirements (actually pseudo-criteria) have the same importance.

At first glance, the rankings are not similar but not so different either. The rankings order the players according to the sportive qualities recorded in Tab. 1. For example, ELECTRE IV and CRBM would select player one first, then player 10. But ELECTRE IV would prefer player 6 in the third place while CRBM would choose player 5.

The discriminant representation provided by the hidden layer is the most determining factor in favor of CRBM against VIKOR or ELECTRE. The players were ranked by the first hidden neuron of the CRBM. The scatter-plot Figure 5 uses the values of the first two hidden neurons. This means the hidden neurons capture a latent representation capable of discriminating between the two classes, "rears" and "forwards."
5 CONCLUSION AND DISCUSSIONS

CRBMs are domain-independent feature extractor that transforms raw data into latent variables. The most relevant questions are: how to dimension the hidden layer $h$ optimally? And how do the neurons interact?

Our generative network is relatively small. Hence to compute $p(h|v)$ is an affordable problem for small RBM, but once we have a large number of hidden neurons, it becomes impossible to compute all possible $p(h|v)$. The more neurons, the more computational efforts are needed: massive networks should not be the only way to reduce the modeling error. The choice of dimension remains today an unsolved issue.

In addition, for each configuration $v$, some hidden neurons have a probability close to 0 or 1, meaning that for each $v$, some states of $h$ are irrelevant.

Besides being energy-consuming, a significant dimension network requires much time to learn. In many papers, authors focus on comparing the performance between models but barely reach computational efforts between models. The issue of computational efforts can have a significant impact, particularly in the real-time system. Still, it is enormously dependent on the data, the application, and the used hardware.

To reduce the bias between the data distribution $P_{data}(x)$ and the estimated data distribution $P_{model}(x)$ the cost function was modified (Eq. 12).

The next step is constructing a deeper network, such as deep belief network (DBN), that may provide more explainability hints.

REFERENCES


