L5: Rankings from pairwise comparisons

Caterina De Bacco

May 5, 2025

1. Introduction

In the previous lecture we learned how to model networks with latent variables, focusing on community detection. However, community structure is not the only option in terms of what latent variable can model a network dataset.
Consider a set of pairwise comparisons, for instance matches between teams in sport or endorsements between institutions. This can also be represented as a network, a directed one where the direction and weight both represent the outcome of a pairwise comparison. The data can be encoded into an adjacency matrix A of dimension $N \times N$. Its entries $A_{ij} > 0$ denote how much i wins over j and can be of any type, e.g. real numbers, binary or discrete. For instance, in a match between teams i, j won by i, it can be the point difference (discrete), or the expected point difference (real). In general, A is not symmetric, as $A_{ij} \neq A_{ji}$, as the outcome of a win by i against j may be different to another match outcome where j wins instead.

In this context, rather than in communities, we may be interested in ranking individuals based on strength or prestige. Hence, we need to learn latent scores $s = (s_1, \dots, s_N)$ on nodes:

$$s_i \in \mathbb{R} \quad i = 1, \dots, N \quad . \tag{1}$$

These are real-valued quantities, that in turns imply an *ordinal* ranking $r = (r_1, \dots, r_N)$ such that:

$$s_i \geq s_j \implies r_i \leq r_j \quad , \tag{2}$$

where $r_i \in \mathbb{Z}$ denote who is first, second and so far until the last one in the ranking. Note that the signs in eq. (2) are arbitrarily set to have the lowest ranked to be the best individual.

Goal: learn latent scores s from a set of pairwise comparisons.

As an example, consider a set of matches in a sport. We would like to learn who is the strongest team, who is the weakest, and information like how closed are in strength team 4 and 5; in a match between team 1 and team 2, who is likely to win? How predictable is a league compared to another one?

To answer to these questions we need to learn the scores s from the data A. For this, we adopt a probabilistic perspective and set up a likelihood:

$$P(A|s) \quad , \tag{3}$$

that regulates the probability of observing the outcomes, given the scores. We can also add a prior on the scores $P(s)$ and setup a Bayesian model.
Our goal is to learn the posterior $P(s|A)$.

2. Modeling the direction of an outcome: Bradley-Terry model

An intuitive model to tackle this problem is the Bradley-Terry Bradley and Terry (1952), introduced by Zermelo (1929). It posits that the probability of i winning over j is:

$$p_{ij} = \frac{\pi_i}{\pi_i + \pi_j} \quad , \tag{4}$$

where $\pi_i = e^{s_i} \geq 0$, which were denoted by Zermelo as *Spielstärken* (playing strengths). Hence, we focus here on discrete outcomes where $A_{ij} \geq 0$ is the number of times that i wins over j (it is 0 if they never played, or if they only tie). We thus have:

$$P(A_{ij}|\pi_i, \pi_j) = p_{ij}^{A_{ij}} \quad . \tag{5}$$

Note that we do not have a term $(1 - p_{ij})$ as usual in Bernoulli distributions because we are not modeling the entry $A_{ij} = 0$, we only focus on what is probability of winning (or losing), *given* a match was played and that ties are not allowed.
The overall joint probability over all matches is then:

$$P(A|\pi) = \prod_{i,j} p_{ij}^{A_{ij}} = \prod_{i,j} \left(\frac{e^{s_i}}{e^{s_i} + e^{s_j}} \right)^{A_{ij}} \quad . \tag{6}$$

Question: Why do we need the s instead of simply using the π?

Because substituting the expression $\pi_i = e^{s_i}$ we see that what matters is a difference in score $s_i - s_j$!
In fact:

$$P(A_{ij}|\pi_i, \pi_j) = p_{ij}^{A_{ij}} = f(s_i - s_j)^{A_{ij}} = \left(\frac{1}{1 + e^{-(s_i - s_j)}} \right)^{A_{ij}} \quad , \tag{7}$$

i.e. $f(\cdot)$ is a logistic function.
Then, scores are obtained by MLE:

$$\hat{s} = \arg\max_s P(A|s) \quad , \tag{8}$$

or alternatively by MAP if we add priors on the s.

3. Modeling the existence and direction of an interaction: SpringRank model

So far we have always assumed that the *direction* of the interactions is affected by the status, prestige, or social position of the entities involved. But it is often the case that even the *existence* of an interaction, rather than its direction, contains some information about those entities' relative prestige. For example, in some species, animals are more likely to interact with others who are *close* in dominance rank Hobson and DeDeo (2015).

ASSUMPTION 1. *Interactions between similar individuals are more likely to take place.*

This suggests that we can infer the ranks of individuals in a social hierarchy using *both* the existence and the direction of their pairwise interactions.

Question: What physical system penalizes you for having distances too far or too close?

Springs!

SpringRank De Bacco *et al.* (2018) was introduced to tackle this problem. It models the dataset by imagining the network as a physical system where a directed interaction $i \to j$ is an oriented spring of resting length l and displacement $s_i - s_j$. Recall Hooke's law for the force of a spring:

$$F = -k\left(s_i - s_j - \ell\right) \quad , \tag{9}$$

where k is the spring constant, regulating how stiff or loose it is; ℓ is the rest length, and $s_i - s_j$ is the displacement from the relaxed position ℓ. From now on for simplicity we set $\ell = 1$.
This force increases when we either move the two end beads too close, compressing the spring and then feeling that it pusehs back; or when we pull them apart extending the spring and then feeling that the spring pulls you back.
This force gives an energy of the system of two entities:

$$H_{ij} = \frac{1}{2}\left(s_i - s_j - 1\right)^2 \quad , \tag{10}$$

which is quadratic, hence convex in the scores. It is minimized when $s_i - s_j = 1$.
The overall energy of the system is:

$$H(s) = \sum_{ij} A_{ij} H_{ij}(s) = \frac{1}{2}\sum_{ij} A_{ij}\left(s_i - s_j - 1\right)^2 \quad . \tag{11}$$

As it is convex, we can get set the derivative to zero and get the linear system:

$$\left[D^{out} + D^{in} - (A + A^T)\right] s = \left[D^{out} - D^{in}\right] 1_N \quad , \tag{12}$$

where D^{out} is a diagonal matrix containing the out-degree $d_i = \sum_j A_{ij}$, similarly for D^{in}; 1_N is a vector of all 1 of dimension N. Solving in s gives the SpringRank scores s^*.
One can also add an external field as a self-energy per node:

$$H_\alpha(s) = H(s) + \frac{\alpha}{2}\sum_{i=1}^{N} s_i^2 \quad , \tag{13}$$

which can be seen probabilistically as fixing a prior on each strength. The constant α tunes the relative magnitude of the two terms.
We can see an example of learned scores from soccer matches of different leagues and how this is correlated to actual attained standings in Fig. 1.

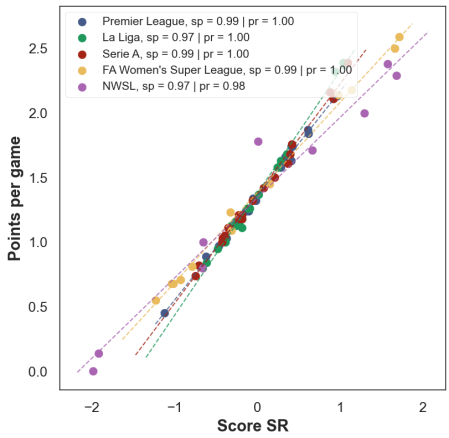

FIGURE 1. SpringRank and points per game in professional soccer matches. Scores inferred using SpringRank vs points per game in different league, we considered one example season per league (2015-16 for the men leagues, 2020-21 for FA WSL and 2019 for NWSL, female leagues). Values inside the legend are Pearson ("pr") and Spearman ("sr") correlations. Dashed lines are regression obtained by fitting the data. Each marker is a team, colors are leagues.

3.1. Probabilistic formulation

Question: There is no probability here. How do we get a probabilistic interpretation?

With Boltzmann distributions!
In fact, by introducing an inverse temperature parameter $\beta \geq 0$, we can set:

$$P(s) \propto \exp\{-\beta H_\alpha(s)\} \tag{14}$$
$$= \mathcal{N}\left(s; s^*, \Sigma\right) \quad , \tag{15}$$

where the covariance matrix is:

$$\Sigma = \frac{1}{\beta}\left[D^{out} + D^{in} - (A + A^T + \alpha\mathbb{I})\right]^{-1} \quad . \tag{16}$$

We can now see how β controls the noise of the system. In the limit $\beta \to \infty$, the scores are sharply peaked around the ground state s^*, the maximum likelihood estimate of the scores. On the opposite side for $\beta \to 0$, the scores are noisy and do not depend on the data A. Hence, β can be used to assess the strength of the hierarchy, the larger β the stronger the hierarchy and games are more likely to end with an outcome aligned with the score difference.

Question: How do we simulate new matches?

For this we need a generative model for the A, given the s. Using what learned above, we can set the expected value as:

$$\mathbb{E}[A_{ij}] = c \exp\left\{-\beta H_{ij}\right\} \quad , \tag{17}$$

where c is a constant that controls the sparsity of the dataset. We then have to determine $P_{ij}(\beta)$ as a proper distribution with expected value as in eq. (17). For instance, for discrete outcomes we can use a Poisson distribution with $\lambda_{ij} = \mathbb{E}[A_{ij}]$. By playing with β we can generate outcome more (high β) or less (small β) correlated with the scores. It can be shown that the MLE estimate of the scores in this Poisson distribution, in the limit of large β (strong hierarchy), approach the SpringRank ground state solving eq. (12).

Question: How do we predict who wins a match?

This is a question about the conditional probability of i winning over j, given we observe a match between them. We are thus asking the probability of the direction, not of the existence, of an edge. For this, we can consider:

$$P_{ij}(\beta) = \frac{e^{-\beta H_{ij}}}{e^{-\beta H_{ij}} + e^{-\beta H_{ji}}} = \frac{1}{1 + e^{-2\beta(s_i - s_j)}} \quad , \tag{18}$$

where β here can be learned using cross-validation, given some desired performance metric.

4. Depth and luck

We have asked before what league is the most hierarchical, but it is not really clear how to define this precisely. We can maybe reframe by asking two different and separate questions.

Question: 1. What league has more *depth*?

This question touches upon the idea that certain leagues have larger gaps between teams, so that distinguishing between strength of players is easier. We can see an example of this in Fig. 2. We can see that the NWSL has a larger depth, measured by $s_{max} - s_{min}$, as distances between top and bottom teams are much larger than that of other leagues. This is one possible way to measure depth. To better compare different leagues, one possible way is to use the parameter β, as this is a factor multiplied to differences $s_i - s_j$, as shown in eq. (18) and it determines shape of the distribution $P_{ij}(\beta)$.

4.1. Rescaling the scores

Let's first find a common unit of measure to compare leagues (or datasets). For this, we can use probability. Specifically, we can consider a difference $\Delta s = s_i - s_j$ between scores and translates this into a probability of i winning over j, so that we can map a given Δs to a unit of probability q that has the same meaning, regardless the league or dataset. By

fixing this unit q to some value, we can get from eq. (18) the corresponding value Δs that gives a probability q of the node with higher score to win:

$$\Delta_\beta = \frac{1}{2\beta} \log\left[\frac{q}{1-q}\right] \quad . \tag{19}$$

For instance, fixing $q = 75\%$ and assuming that the $\beta = 1.6$, we get a $\Delta_\beta = 0.34$. Each league has a different β and thus a different Δ_β for a given fixed q. Hence, we can rescale the scores per league as $\hat{s}_i = s_i/\Delta_\beta$ so that now a difference $\hat{s}_i - \hat{s}_j = 1$ of one unit means the same across leagues: it is the score difference that results in a probability q of i to win against j. We can now compare plots of different leagues using the same units! You can see an example in Fig. 2. Note that this rescaling should be done only for visualization and interpretation purposes, not for calculating probabilities of winning, as a rescaling alters the shape of the probability in eq. (18).

4.2. β and depth of competition

With this mapping between score difference and probabilities, we can see what is the role played by β in shaping eq. (18). Suppose we divide the scores into level, each separated by a gap Δ_β as in eq. (19). A league with many such levels has more depth, meaning that there are more matches where one team have at least q probability of winning, and the larger this probability the larger the level gap between the two teams.

> *Question*: How do we calculate the number of levels? (and thus the depth of a competition)

This depends on the distribution of the difference $\Delta_{ij} = s_i - s_j$ over pairs of nodes, as well as on β. In the example above, for a reference level $\Delta_\beta = 0.157$ (corresponding to 75% probability of winning), we can have 1 level if all the scores are within $\Delta_{ij} < \Delta_\beta$. This suggest to get a reference difference $\hat{\Delta}$ and define:

$$n_{levels} = \frac{\hat{\Delta}}{\Delta_\beta} = \frac{\hat{\Delta} 2\beta}{\log q/(1-q)} \propto \hat{\Delta}\,\beta \quad , \tag{20}$$

where we have highlighted the proportionality between the number of levels vs the reference difference and β. In the example above, we have implicitly used as reference the max difference $\hat{\Delta} = s_{max} - s_{min}$, where $s_{max} = \max_i s_i$ and similarly for s_{min}. Alternatively, one can use the average distance $\hat{\Delta} = \sum_{i<j} |s_i - s_j|/(N(N-1)/2)$ or the standard deviation of s over the nodes to use as reference a typical pair of players. Regardless this choice, we can see how β appears as a proportionality factor, meaning that a higher β leads to a higher number of levels, and thus a deeper competition.
In the soccer datasets in Fig. 2, we can take directly $n_{levels} = \hat{\Delta} = s_{max} - s_{min}$ as the markers are the rescaled $\hat{s}$ for a $\Delta_\beta = 1$. With this choice for $\hat{\Delta}$, we can see how the NWSL is the deepest league, with around 4 levels, while the men leagues are shallower with around 2 levels or less.
In Fig. 3 we show values for additional types of datasets.

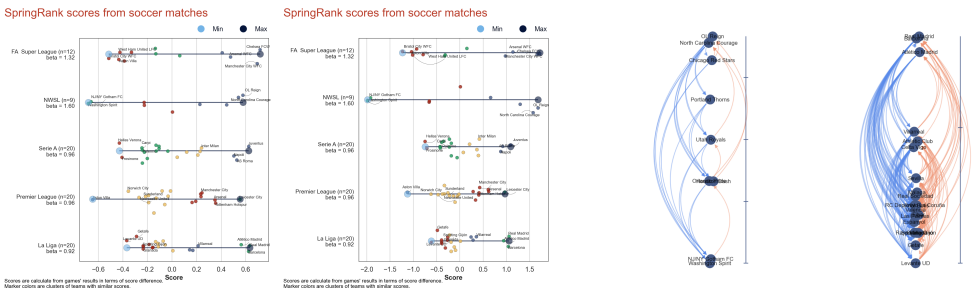

FIGURE 2. SpringRank score distribution in different soccer leagues. Markers are inferred scores s, colors are obtained by clustering (in one-dimension) the scores to highlight teams of different tiers. Left) scores are the original ones learned from solving the linear system in eq. (12). Center) Rescaled scores $\hat{s}$ such that a unit of $s_i - s_j = 1$ corresponds to $q = 75\%$ probability of i winning against j. Right) NWSL and La Liga scores and outcome directions. Blue arrows go down the hierarchy, red ones go against the hierarchy as a team with lower score wins against a team with higher score. Ticks in the y-axis are separated by $\Delta s = 1$, to highlight a $q = 75\%$ probability of winning.

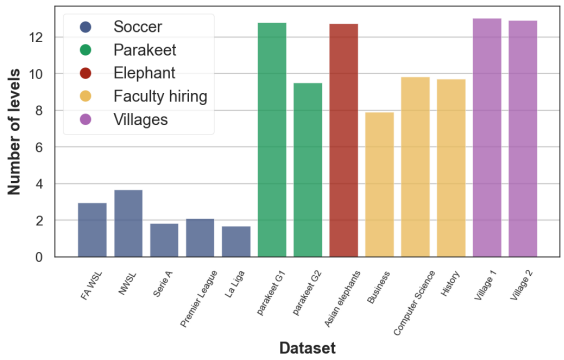

FIGURE 3. Depth of competition. We report the number of levels computed as in eq. (20) for different datasets. Reference probability unit per level is $q = 75\%$.

4.3. β and luck

Now we are ready to ask the second question.

> *Question*: 2. In what league does *luck* play a bigger role?

This is a question about unpredictability of an outcome, about how often do we expect to observe an "upset" win, where a very weak player wins against a very strong one. This can be easily confused with the level of noise in the system, which is controlled by the inverse temperature β in the SpringRank algorithm. We saw in the discussion around eq. (17) that a very high temperature, i.e. $\beta \to 0$, leads to outcomes A_{ij} independent from the scores s_i, s_j. Hence, a possible way to define luck is the level of correlation that score difference have with the outcome. In this sense, β as defined in that paragraph can be used as parameter to measure the role of luck, similarly to what done in statistical physics to measure noise with the temperature. A similar behavior applies to the β in eq. (18), as wehn $\beta \to 0$ the the dependence between probability of a win and score difference goes to

zero, as we get a probability 0.5 for both competitors to win. One possible alternative way to capture luck in a Bradley-Terry type of model is to inflate the 1/2 probability as done in Jerdee and Newman (2024). This is done by extending the logistic function as:

$$f_\alpha(s) = \frac{1}{2}\alpha + (1-\alpha)\frac{1}{1+e^{-s}} \quad , \tag{21}$$

where the parameter $\alpha \in [0,1]$ tunes the level of inflation, the higher the α the higher the probability that the outcome is 50-50, regardless the scores of the players. With this formulation though, we need to have deep competitions to be able to clearly distinguish the role of α, because in shallow competitions where scores are very close to each other it is quite likely to observed upsets, without having to take into account an explicit α. Also, the shape of this curve for large α, is similar to the shape of a standard BT model with $\beta \to 0$.

5. Model selection

How do we evaluate the performance of the model? How do we decide which is the best model? We can use prediction tasks. In this context, one relevant question is how does a score s help to predict a match outcome. Hence, we can use cross-validation to hide part of the datasets, learn from the training and test on the heldout set.
This can also be used to select optimal hyperparameters as α and β.

Question: What performance metric should we use?

There are many options for this: accuracy, upsets, log-likelihood, cross-entropy, etc.

5.1. Learning latent rankings from pairwise comparisons: summary

- Learning ranking from pairwise comparison require asking what variables do we expect scores to control: direction, existence, both?
- Depth of competition can be measured by mapping units of score difference into units of probabilities
- Luck is trickier to measure, as noise is already inherently accounted for in a probabilistic model. What do we expect luck to do that noise cannot alreday capture?
- Determining what is the best scoring system requires model selection criteria. In this context, often one looks at prediction tasks

References

R. A. Bradley and M. E. Terry, Biometrika **39**, 324 (1952).
E. Zermelo, Mathematische Zeitschrift **29**, 436 (1929).
E. A. Hobson and S. DeDeo, PLoS computational biology **11**, e1004411 (2015).
C. De Bacco, D. B. Larremore, and C. Moore, Science advances **4**, eaar8260 (2018).
M. Jerdee and M. Newman, Science Advances **10**, eadn2654 (2024).