IFT 6756 - Lecture 23
Evaluation of Multi-Agent Systems

This version of the notes has not yet been thoroughly checked. Please report any bugs to the scribes or instructor.

Scribes
Winter 2021: Raparthy Sharath Chandra, Kavin Patel

Instructor: Gauthier Gidel

1 Summary

Progress in machine learning is measured by careful evaluation on problems of outstanding common interest. Here, we consider two scenarios for evaluation : Agent vs Agent (AvA) where agents compete directly as in Go and Starcraft; and Agent vs Task (AvT) where algorithms are evaluated on suites of datasets or environments.

Performance of these algorithms are generally quantified using ELO ratings. However, due to the limitations of ELO ratings in intransitive and cyclic games, we need to look beyond ELO to quantify these algorithms. Hence, in this lecture, we will first try to understand to estimate the ELO rating, what are its limitations and a solution (Nash Averaging) that can overcome these limitations.

2 Antisymmetric (zero-sum) Game

In this section, we first recall some basic facts about antisymmetric matrices and then see how it can be used as a representation of functional form of zero-sum games.

Matrix Φ is antisymmetric if $\Phi + \Phi^T = 0$. Antisymmetric matrices have even rank and imaginary eigenvalues $\{\pm i\lambda_j\}_{j=1}^{rank(\Phi)/2}$. Any antisymmetric matrix Φ admits a real **Schur decomposition**:

$$\Phi = Q \cdot \wedge \cdot Q^T$$

where Φ, Q and $\wedge$ are $W \times W$ matrices. Q is orthogonal and $\wedge$ consists of zeros except for (2×2) diagonal-blocks of the form:

$$\wedge = \begin{bmatrix} 0 & \lambda_j \\ -\lambda_j & 0 \end{bmatrix}$$

The entries of $\wedge$ are real numbers, found by multiplying the eigenvalues of Φ by $i = \sqrt{-1}$.
In this context, the functional form of zero-sum game can be represented as the anti-symmetric payoff as :

$$\Phi(u, w) = -\Phi(w, u),$$

where w and u are policies of the agent.
The general intuition drawn from this form is, switching the roles switches the result. Meaning, winning a strategy for one player is equivalent to losing that strategy for its opponent. Strategic games like Chess, Go, Poker (with randomized initialization) can be represented in such manner.
This setting can also be generalized for non-zero sum settings (though it would be computationally heavy because of the two losses). For sake of simplicity, we will focus on the zero-sum game throughout this lecture.

3 Estimating Elo for AvA

In this section, we cover the detailed mechanics of Estimating the Elo rating of an agent, estimating it at a given time t and computing Elo for higher-order functions.

3.1 The Elo rating system

The Elo rating system is a method for calculating the relative skill levels of players in zero-sum two-player games. Initially started as a quanitifiable scores estimated in chess for all players, the Elo rating system is also used extensively in other games, including basketball, American football, rest-of-the-world football, baseball, Scrabble, and even video games such as Overwatch and PUBG.

Generally speaking, the performance in the ELO system is not measured in absolute terms. It is inferred from wins, losses, and draws against other players. Players' ratings depend on the ratings of their opponents and the results scored against them.

Elo's central assumption was that the (chess) performance of a player in each game is a random variable, and that it follows a normally distributed bell-shaped curve over time. Thus, while a player might perform significantly better or worse from one game to the next, the mean value of their performances (a reflection of their true skill) would remain the same. The assumption here is that this mean value of the performances for any given player only changes *slowly* over time.

Suppose n agents play a series of pairwise matches against each other. Elo assigns a rating to each player $i \in [n]$ based on their wins and losses, which we represent as an $n-$vector. The predicted probability of u beating w given their Elo ratings is:

$$P(u > w) = \frac{1}{1 + exp(\alpha \cdot (f(w) - f(u)))}$$

where $f(w)$ and $f(u)$ are the Elo ratings of w and u respectively.

In AvA, results are collated into a matrix, say $\Phi(u, w)$, of win-loss probabilities based on relative frequencies of u and w. Since logit functions are inverse of the sigmoid functions, one can state that matrix $\Phi(u, w)$ can be represented as:

$$\Phi(u, w) = \text{logit}(P(u > w)) = \alpha \cdot (f(u) - f(w))$$

This shows the antisymmetric payoff of matrix $\Phi(u, w)$ since $P(u > w) + P(w > u) = 1$

3.2 Online estimation of Elo

Let p_{ij} be the true probability and $\hat{p_{ij}}$ be the estimated probability of players i, j among n agents where $i, j \in [n]$. Given their respective Elo ratings f_i and f_j, the predicted probability of i beating j is:

$$\hat{p_{ij}} = \sigma(\alpha f_i - \alpha f_j)$$

where σ is a sigmoid function. The constant α is not important in what follows, so we assume $\alpha = 1$.

It can be observed that only the difference between Elo ratings affects win-loss predictions. We can therefore impose that Elo ratings sum to zero i.e. $f^T 1 = 0$, without loss of generality. Hence, the cross-entropy loss can be defined as:

$$l(\hat{p_{ij}}, p_{ij}) = -p_{ij} \log(\hat{p_{ij}}) - (1 - p_{ij}) \log(1 - \hat{p_{ij}}),$$

where $\hat{p_{ij}} = \sigma(f_i - f_j)$

Now, suppose that the t^{th} match pits player i against j, with outcome $S_{ij}^t = 1$ if i wins and $S_{ij}^t = 0$ if i loses. Online stochastic gradient descent update with the above loss can be obtained as :

$$f_i^{t+1} = f_i^t - \eta \nabla_{f_i} l(\hat{p_{ij}}, S_{ij}^t) = f_i^t + \eta(S_{ij}^t - \hat{p_{ij}})$$

In general setting, the step-size η in the above equation should be reducing in order to get estimated probability equivalent to true probability. However, since the agents are always evolving over time with constant change in their Elo estimate, one cannot impose a reducing/vanishing step-size. Hence, in order to get the optimization perspective on Elo, it is necessary to standardize the stochastic gradient descent (SGD) with constant step-size. This can also ensure a faster adaptive behaviour among the agents. The optimization equation of agents with Elo score f_i and f_j in the population can be stated as:

$$\min_{f_i} \mathbb{E}_{f_j \sim \text{pop}} l(\sigma(f_i - f_j), \mathbb{P}(i > j))$$

Here, it is to be noted that SGD with constant step-size does not converge. At most, it can only converge to a neighbourhood proportional to the variance times the step-size (also referred to as noise ball).

3.3 Estimation of Elo at a given time

So far, we saw the general concept of the Elo and how it can be estimated in the online setting. In this section, we will see how one can estimate the Elo in an "offline" setting. Compared to the online mode, the offline setting is more settled since it is always assumed that the player does not get any better with time. That way, the chances of badly estimating the opponent's Elo score is ruled out. The general rule to estimate the Elo rating in offline mode is simple : If we have estimates of all match-ups, then we can estimate the Elo of an individual player correctly.

Let $\mathbb{B}$ be the population of agents with their Elo ratings u_i where $i \sim \mathbb{B}$. Thereafter, one can represent all the match-ups of the agents in the form of a payoff matrix as $\mathbb{A}_{\mathbb{B}}$. Hence, the simultaneous match-up of agents i and j can be represented as :

$$[\mathbb{A}_{\mathbb{B}}]_{ij} = \Phi(u_i, u_j)$$

So now, we have the payoff matrix with all simultaneous match-ups of the agents in the offline setting. Now, how can we estimate a good Elo score at a given time t?

Getting Elo from A:
To answer this question, let us assume that we have good estimates of i winning against j. Hence, we follow the intuition of A_{ij} as :

$$A_{ij} = f_i - f_j$$

Hence, by algebraic manipulation, the estimates of match-ups of agents in $\mathbb{A}$ can be written as :

$$\mathbb{A} = \begin{bmatrix} f_1 & \dots & f_1 \\ \vdots & \ddots & \vdots \\ f_n & \dots & f_n \end{bmatrix} - \begin{bmatrix} f_1 & \dots & f_n \\ \vdots & \ddots & \vdots \\ f_1 & \dots & f_n \end{bmatrix} = f1^T - 1f^T$$

This equation proves that one can estimate the agent's elo score by just summing the rows of other agents' elo scores. Meaning, an agent's average score against every other agents' score gives a good estimate of that agent's elo score. Hence, in mathematical form, one can state the following :

$$f_i - \bar{f} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{A}_{ij}$$

where f_i is the individual elo score of the agent i and $\bar{f}$ is the average elo score of all the agents.

Therefore, one can say that in a finite tournament, an agent's Elo score is basically that agent's average performance. However, it is to be noted that this is only true under the assumption that we live in a perfect world where $\mathbb{A}_{ij} = f_i - f_j$ holds true.

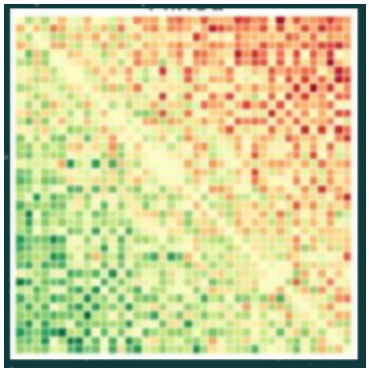

Figure 1: The following plot shows that the bottom left players are consistently better than bottom top players. This plot is consistent with the players' individual score calculated in the offline setting.

In real world scenario, this is usually not the case. The transitive component $[f1^T - 1f^T]$ is not the only factor that can summarize the real-world games. It is important to include the cyclic interactions along with transitive component for non-transitive games (which is usually seen in practice. In case of just transitive games, we say that the cyclic component is zero). Hence, we can say that the payoff matrix $\mathbb{A}$ can be stated as :

$$\mathbb{A} = (\text{Transitive Component}) + (\text{Cyclic Component})$$

$$\therefore \mathbb{A} = (f1^T - 1f^T) + \mathbb{B}$$

Hence, for perfect world, we say that the cyclic component $\mathbb{B} = 0$. Transitive component computes the average performance of the players in the tournament. In comparison to this, the value of B should be very negligible.

So, the takeaway is, there is a natural way to decompose $\mathbb{A}$ as a sum of transitive component and a cyclic component. This decomposition always exists under any assumptions. It can help us identify what is the type of the game.

3.4 Limitations of Elo

Elo is useful to predict the win-loss probability under the assumption that the game is transitive. That is, the game is of the form :

$$\mathbb{P}(i > j) = \sigma(f_i - f_j)$$

However, the win-loss probabilities predicted by Elo ratings can fail in simple (non-transitive) cases. For example, simple cyclic games like rock, paper and scissors will all receive the same Elo ratings. Elo's predictions are $\mathbb{A}_{ij} = \frac{1}{2}$ for all i, j – and so Elo has no predictive power for any given pair of players. Hence, Elo is meaningless in cyclic (non-transitive) games. For non-transitive games, higher-order Elo is necessary.

3.5 Higher order Elo

Elo ratings bake-in the assumption that relative skill is transitive. However, there is no single dominant strategy in games like rock-paper-scissors or StarCraft. Elo's predictive failures are due to the cyclic component that uniform averaging ignores. Rating systems that can handle non-transitive abilities are therefore necessary.

The fundamental algebraic structure of tournaments and evaluation is antisymmetric. Techniques specific to antisymmetric matrices are less familiar than approaches like PCA that apply to symmetric matrices and are typically correlation-based. However, we can apply Schur Decomposition we saw in Section 2 to antisymmetric matrices and get the results.

Consider the non-transitive game contained the payoff matrix $\mathbb{A}$ as :

$$\mathbb{A} = f1^T - 1f^T + \mathbb{B}$$

Handling non-transitive abilities requires learning an approximation to the cyclic component $\mathbb{B}$. Schur decomposition allows to construct low-rank approximations that extend Elo. Note, antisymmetric matrices have even rank. Consider:

$$\mathbb{B} = O \begin{bmatrix} 0 & \lambda_1 & \dots \\ -\lambda_1 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} O^T$$

where O and O^T are orthogonal matrices and $\lambda_1 \geq \cdots \geq \lambda_p$.

The above equation can further be estimated and simplified (by algebraic manipulations) as :

$$\mathbb{B} \approx \lambda_1 O_{n\times 2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} O^T_{n\times 2}$$

Hence, with this decomposition of the cyclic component, we can finally say that the performance of the agent i depends on two quantities : Skills (general ELO Rating) f_i; and strategy (Cyclic vector) (O_{i1}, O_{i2}).

Finally, the win-loss prediction $\hat{p}_{ij}$ of agent i beating agent j in a non-transitive game setting contained in payoff matrix $\mathbb{A}$ is :

$$\hat{p}_{ij} = \sigma(\mathbb{A}_{ij}) \approx \sigma(f_i - f_j + \lambda_1(O_{i1}O_{j2} - O_{i2}O_{j1}))$$

In this equation, $(f_i - f_j)$ shows the difference in skills of agents i and j. and $(O_{i1}O_{j2} - O_{i2}O_{j1})$ shows the strategy (cyclic component) employed by both agents. λ_1 being a real number tells us how much cyclic the game is. $\lambda_1 \approx 0$ means the game is transitive.

4 Agents v/s Tasks

5 Desired Properties

5.1 Maximum entropy Nash-Equilibrium

5.2 Atari Results Discussion

References