Re-evaluate Evaluation

David Balduzzi, Karl Tuyls, Julien Perolat, Thore Graepel

Presented by Yuchen Lu

Motivation: Problem of Redundant Evaluation

Let's first use a common scenario in multi-task evaluation, we uniform average to rank the model.

Task	1	2	3	Mean	Rank
Model A	89	93	76	86	1st
Model B	85	85	85	85	2nd
Model C	79	74	99	84	3rd

Motivation: Problem of Redundant Evaluation

What if we add another task 4, which has similar bahavior as task 3...

Task	1	2	3	4	Mean	Rank
Agent A	89	93	**76**	**77**	83.75	3rd
Agent B	85	85	**85**	**84**	84.75	2nd
Agent C	79	74	**99**	**98**	87.5	1st

Our rank changes a lot, biasing toward task 3 and 4.

Motivation: Problem of Redundant Evaluation

Suppose we have the following evaluation result for a two-player game (chess, go, poker), where the number means the probability of row player winning against column player. The rule of thumb is to use Elo for ranking.

	A	B	C	Elo
A	0.5	0.9	0.1	0
B	0.1	0.5	0.9	0
C	0.9	0.1	0.5	0

Motivation: Problem of Redundant Evaluation

If we copy agent C to be the fourth agent, the resulting Elo rating would be changed...

	A	B	C	C'	Elo
A	0.5	0.9	0.1	0.1	-63
B	0.1	0.5	0.9	0.9	63
C	0.9	0.1	0.5	0.5	0
C'	0.9	0.1	0.5	0.5	0

It turns out, Elo can be viewed as taking uniform average at the logit space. We want to find the ranking or evaluation which could tackle with redundant data.

Motivation: Algebraic Property of Evaluation

The evaluation data can be viewed as an anti-symmetric matrix. $\boldsymbol{A}$ is symmetric iff. $\boldsymbol{A} + \boldsymbol{A}^T = \mathbf{0}$.

In AvA: Suppose the probability matrix is $\boldsymbol{P}$. Then we can set $= logit(\boldsymbol{P})$ where $logit(x) = \log \frac{x}{1-x}$. $\boldsymbol{A}$ is anti-symmetric because $p_{ij} + p_{ji} = 1$.

In AvT: Suppose $\boldsymbol{S} \in R^{m \times n}$ the performance matrix with m models and n tasks. Then we can construct the anti-symmetric matrix by treating each task as a player. So

$$A = \begin{bmatrix} \mathbf{0}_{m \times m} & \boldsymbol{S} \\ -\boldsymbol{S}^T & \mathbf{0}_{n \times n} \end{bmatrix}$$

Motivation: Algebraic Property of Evaluation

flow: Consider a fully connected graph with n vertex. Assign a flow $\boldsymbol{A}_{ij}$ to each edge of the graph. The flow in the opposite direction ji is $A_{ji} = A_{ij}$, so flows are just anti-symmetric matrices.

Motivation: Algebraic Property of Evaluation Matrix

divergence: Divergence of a flow, denoted as $div(\boldsymbol{A}) = \frac{1}{n}\boldsymbol{A} \cdot \mathbf{1}$, is essentially the row-average of $\boldsymbol{A}$. It is essentially what Elo and other uniform averaging scoring is doing.

gradient flow: Suppose you have a n-dimension vector $\boldsymbol{r}$. Then the gradient flow $\boldsymbol{A} = grad(\boldsymbol{r})$ such that $\boldsymbol{A}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$.

curl: The curl of a flow, denoted as $curl(\boldsymbol{A})$, is a three way tensor such that $curl(\boldsymbol{A})_{ijk} = \boldsymbol{A}_{ij} + \boldsymbol{A}_{jk} - \boldsymbol{A}_{ik})$. If $curl(\boldsymbol{A})_{ijk} = 0$, it means the comparison between i, j, k are transitive.

rotation: The rotation of a flow, denoted as $rot(\boldsymbol{A})$, is defined as $rot(\boldsymbol{A})_{ij} = \frac{1}{n}\sum_k curl(\boldsymbol{A})_{ijk}$.

Motivation: Algebraic Property of Evaluation

Paper-Rock-Scissor. Purely cyclic.:

$$C = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}, div = \mathbf{0}, curl \neq \mathbf{0}.$$

Modify paper to also beat scissor. Purely transitive:

$$T = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}, div = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, curl = \mathbf{0}.$$

Mixed: $\alpha C + \beta T$

Motivation: Algebraic Property of Evaluation

Gradient flow $grad(div(\boldsymbol{A}))$ and rotation flow $(rot(\boldsymbol{A}))$are two orthogonal component of the flow $\boldsymbol{A}$. That is

$$rot(grad(div(\boldsymbol{A}))) = \mathbf{0}$$

$$div(rot(\boldsymbol{A})) = \mathbf{0}$$

Hodge decomposition for each flow $\boldsymbol{A}$, there is an decomposition.

$$\boldsymbol{A} = grad(div(\boldsymbol{A})) + rot(\boldsymbol{A})$$

Uniform averaging or Elo, is only showing the divergence part of the story, and it does not fully explain the data. E.g., which part is dominant in our evaluation data?

Motivation: Summary

We want to have a evaluation which can

1. In-variance: The result does not change with redundant data.
2. Continuity: The result should be telling us how (non)transitive the evaluation data is, revealing the interaction dynamics.

Nash Averaging: Intuition

Intuition:

1. Cast the evaluation as a 2 player zero-sum game. You pick the hardest task/opponent. I pick the best model.
2. Let's all be rational and play the best move by finding maximum entropy Nash Equilibrium.
3. Report evaluation score as weighted average using maxent nash weights of tasks.

Comments:

- There exists a maxent nash for each 2-player zero-sum game. (Berg et al., 1999)

Nash Averaging: Invariance

Let's revisit the example in the beginning. We have

$$A = \begin{bmatrix} 0 & 4.6 & -4.6 \\ -4.6 & 0.0 & 4.6 \\ 4.6 & -4.6 & 0.0 \end{bmatrix} \text{ and } A' = \begin{bmatrix} 0 & 4.6 & -4.6 & -4.6 \\ -4.6 & 0.0 & 4.6 & 4.6 \\ 4.6 & -4.6 & 0.0 & 0 \\ 4.6 & -4.6 & 0.0 & 0 \end{bmatrix}$$

The maxent nash for A is $p_A^* = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. nash scores $[0, 0, 0]$, uniform scores $[0, 0, 0]$.

The maxent nash for A is $p_A^* = [\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}]$. nash scores $[0, 0, 0, 0]$, uniform scores $[-4.6, 4.6, 0, 0]$.

Nash Averaging: Continuity

Let $\boldsymbol{C} = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$, $\boldsymbol{T} = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}$, and $\boldsymbol{A} = \boldsymbol{C} + \epsilon \boldsymbol{T}$.

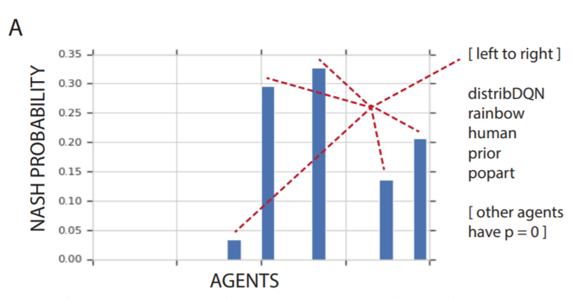
The maxent nash weights are

$$p^*_{\boldsymbol{A}} = \begin{cases} (\frac{1+\epsilon}{3}, \frac{1-2\epsilon}{3}, \frac{1+\epsilon}{3},) & 0 \leq \epsilon \leq \frac{1}{2} \\ (1, 0, 0) & \epsilon > \frac{1}{2} \end{cases}$$

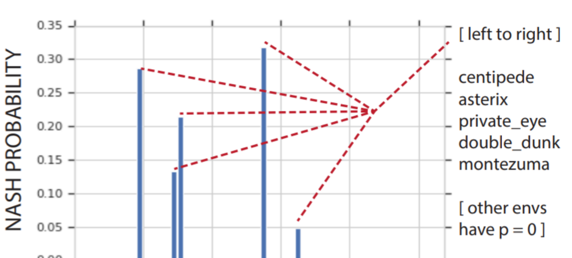
The scores are

$$scores = \begin{cases} (0, 0, 0) & 0 \leq \epsilon \leq \frac{1}{2} \\ (0, -1-\epsilon, 1-2\epsilon) & \epsilon > \frac{1}{2} \end{cases}$$

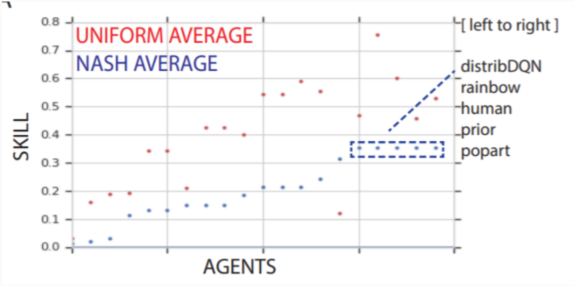
Re-evaluat Atari



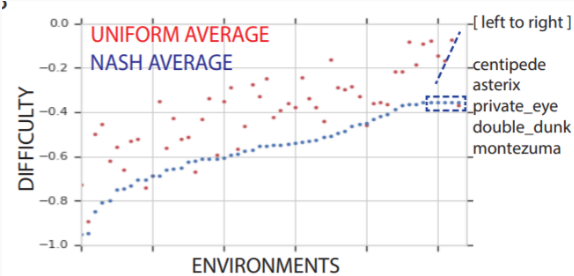
(a) Nash weight for Algo



Re-evaluat Atari



(c) Nash Score for Algo



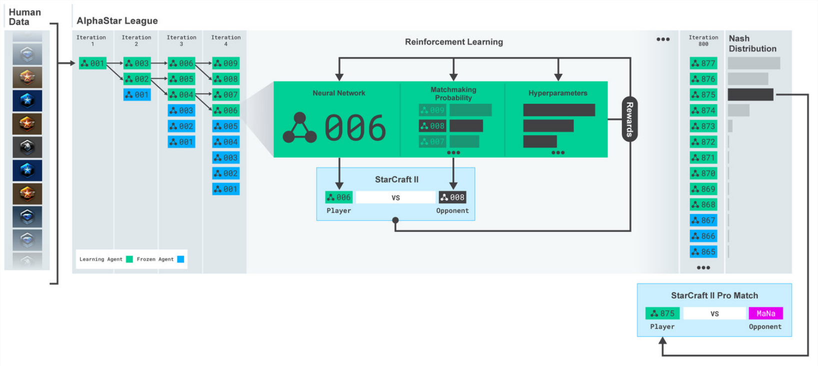


Figure 1: Alpha Star Training Pipeline

Starcraft: Nash League