Game-Theoretic Interpretability for Temporal Modeling

Guang-He Lee [1] **David Alvarez-Melis** [1] **Tommi S. Jaakkola** [1]

Abstract

Interpretability has arisen as a key desideratum of machine learning models alongside performance. Approaches so far have been primarily concerned with fixed dimensional inputs emphasizing feature relevance or selection. In contrast, we focus on temporal modeling and the problem of tailoring the predictor, functionally, towards an interpretable family. To this end, we propose a co-operative game between the *predictor* and an *explainer* without any a priori restrictions on the functional class of the predictor. The goal of the explainer is to highlight, locally, how well the predictor conforms to the chosen interpretable family of temporal models. Our co-operative game is setup asymmetrically in terms of information sets for efficiency reasons. We develop and illustrate the framework in the context of temporal sequence models with examples.

1. Introduction

State-of-the-art predictive models tend to be complex and involve a very large number of parameters. While the added complexity brings modeling flexibility, it comes at the cost of transparency or interpretability. This is particularly problematic when predictions feed into decision-critical applications such as medicine where understanding of the underlying phenomenon being modeled may be just as important as raw predictive power.

Previous approaches to interpretability have focused mostly on fixed-size data, such as scalar-feature datasets (Lakkaraju et al., 2016) or image prediction tasks (Selvaraju et al., 2016). Recent methods do address the more challenging setting of sequential data (Lei et al., 2016; Arras et al., 2017) in NLP tasks where the input is discrete. Interpretability for continuous temporal data has remained mostly unexplored (Wu et al., 2018).

[1]MIT Computer Science and Artificial Intelligence Laboratory. Correspondence to: Guang-He Lee <guanghe@mit.edu>.

The 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2018) at ICML 2018, Stockholm, Sweden, 2018.

In this work, we propose a novel approach to model interpretability that is naturally tailored (though not limited to) time-series data. Our approach differs from interpretable models such as interpretation generators (Al-Shedivat et al., 2017; Lei et al., 2016) where the architecture or the function class is itself explicitly constrained towards interpretability, e.g., taking it to be the set of linear functions. We also differ from post-hoc explanations of black-box methods through local perturbations (Ribeiro et al., 2016; Alvarez-Melis & Jaakkola, 2017). In contrast, we establish an intermediate regime, game-theoretic interpretability, where the predictor remains functionally complex but is encouraged during training to follow a locally interpretable form.

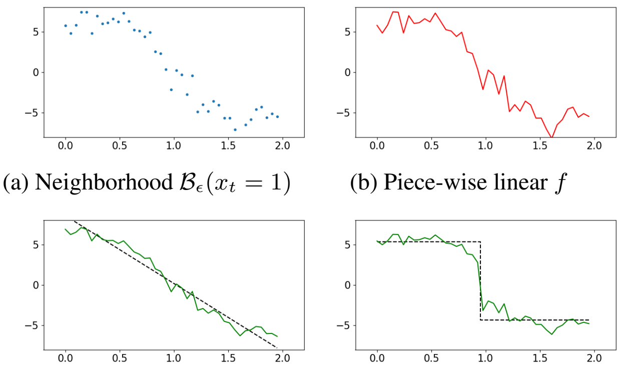
At the core of our approach is a game-theoretic characterization of interpretability. This is set up as a two-player co-operative game between *predictor* and *explainer*. The predictor remains a complex model whereas the explainer is chosen from a simple interpretable family. The players minimize asymmetric objectives that combine the prediction error and the discrepancy between the players. The resulting predictor is biased towards agreeing with a co-operative explainer. The co-operative game equilibrium is stable in contrast to GANs (Goodfellow et al., 2014).

The main contributions of this work are as follows:

- A novel game-theoretic interpretability framework that can take a wide range of prediction models without architectural modifications.
- Accurate yet explainable predictors where the explainer is trained coordinately with the predictor to actively balance interpretability and accuracy.
- Interpretable temporal models, validated through quantitative and qualitative experiments, including stock price prediction and a physical component modeling.

2. Methodology

In this work, we learn a (complex) predictive target function $f \in \mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ together with a simpler function $g \in \mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$ defined over an axiomatic class of interpretable models $\mathcal{G}$. We refer to functions f and g as the predictor and explainer, respectively, throughout the paper. Note that we need not make any assumptions on the function class $\mathcal{F}$, instead allowing a flexible class of predictors. In contrast,



(a) Neighborhood $\mathcal{B}_\epsilon(x_t = 1)$ (b) Piece-wise linear f

(c) $g_{x_t} \in \mathcal{G}_{\text{Linear}}$ (d) $g_{x_t} \in \mathcal{G}_{\text{Decision Stump}}$

Figure 1. Examples of fitting a neighborhood $\mathcal{B}_\epsilon(x_t = 1)$ (1a) with piece-wise linear predictor (1b). When playing with different families of explainers (Figure 1c and 1d; dashed lines), the resulting predictor (in solid green) behaves differently although they admit the same prediction error (mean squared error = 1.026).

the family of explainers $\mathcal{G}$ is explicitly and intentionally constrained such as the set of linear functions. As any $g \in \mathcal{G}$ is assumed to be interpretable, the family $\mathcal{G}$ does not typically itself suffice to capture the regularities in the data. We can therefore hope to encourage the predictor to remain close to such interpretable models only locally.

For expository purposes, we will develop the framework in a discrete time setting where the predictor maps $x_t \in \mathcal{X}$ to $f(x_t) \in \mathcal{Y}$ for $t = \{1, 2, \dots\}$. The data are denoted as $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots\}$. We then instantiate the predictor with deep sequence generative models, and the explainers with linear models.

2.1. Game-Theoretic Interpretability

There are many ways to use explainer functions $g \in \mathcal{G}$ to guide the predictor f by means of discrepancy measures. However, since the explainer functions are inherently weak such as linear functions, we cannot expect that a reasonable predictor would be nearly linear globally. Instead, we can enforce this property only locally. To this end, we define *local* interpretability by measuring how close f is to a family $\mathcal{G}$ over a local neighborhood $\mathcal{B}_\epsilon(x_t)$ around an observed point x_t. One straightforward instantiation of such a neighborhood $\mathcal{B}_\epsilon(x_t)$ in temporal modeling will be simply a local window of points $\{x_{t-\epsilon}, \dots, x_{t+\epsilon}\}$. Our resulting local discrepancy measure is

$$\min_{g \in \mathcal{G}} d_{x_t}(f, g) = \min_{g \in \mathcal{G}} \frac{\sum_{x' \in \mathcal{B}_\epsilon(x_t)} d(f(x'), g(x'))}{|\mathcal{B}_\epsilon(x_t)|}, \quad (1)$$

where $d(\cdot, \cdot)$ is a deviation measurement. The minimizing explainer function, $\hat{g}_{x_t}$, is indexed by the point x_t around which it is estimated. Indeed, depending on the function f, the minimizing explainer can change from one neighborhood to another. If we view the minimization problem game-theoretically, $\hat{g}_{x_t}$ is the *best response strategy* of the local explainer around x_t.

The local discrepancy measure can be subsequently incorporated into an overall regularization problem for the predictor either symmetrically (shared objective) or asymmetrically (game-theoretic) where the goals differ between the predictor and the explainer.

Symmetric criterion. Assume that we are given a primal loss $\mathcal{L}(\cdot, \cdot)$ that is to be minimized for the problem of interest. The goal of the predictor is then to find f that offers the best balance between the primal loss and local interpretability. Due to the symmetry between the two players, the full game can be collapsed into a single objective

$$\sum_{(x_t, y_t) \in \mathcal{D}} \Bigg[\mathcal{L}(f(x_t), y_t) + \frac{\lambda}{|\mathcal{B}_\epsilon(x_t)|} \sum_{x' \in \mathcal{B}_\epsilon(x_t)} d(f(x'), g_{x_t}(x')) \Bigg] \quad (2)$$

to be minimized with respect to both f and g_{x_t}. Here, λ is a hyper-parameter that we have to set.

To illustrate the above idea, we generate a synthetic dataset to show a *neighborhood* in Figure 1a with a perfect piece-wise linear predictor $f \in \mathcal{F}_{\text{Piece-wise Linear}}$ in Figure 1b. Clearly, f does not agree with linear explainers within the neighborhood, despite its piece-wise linear nature. However, when we establish a game between f and a linear explainer $g_{x_t=1} \in \mathcal{G}_{\text{Linear}}$ in Figure 1c, it admits lower functional deviation (and thus stronger linear interpretability). We also show in Figure 1d that different explainer family would induce different outcomes of the game.

Asymmetric Game. The symmetric criterion makes it simple to understand the overall co-operative objective, but solving it is inefficient computationally. Different possible sizes of the neighborhood $\mathcal{B}_\epsilon(x)$ (e.g., end-point boundary cases) makes it hard to parallelize optimization for f (note that this does not hold for g_{x_t}), which is problematic when we require parallel training for neural networks. Also, since f is reused many times across neighborhoods in the discrepancy measures, the value of f at each x_t may be subject to different functional regularization across the neighborhoods, which is undesirable.

In principle, we would like to impose a uniform functional regularization for every x_t, where the regularizer is established on a local region $\mathcal{B}_\epsilon(x_t)$ basis. This new modeling framework leads to an asymmetric co-operative game, where the information sets are asymmetric between predictor f and local explainers g_{x_t}. Accordingly, each local best response explainer $\hat{g}_{x_t}$ is minimized for local interpretability (1) within $\mathcal{B}_\epsilon(x_t)$, thus relying on f values within this region. In contrast, the predictor f only receives feedback in terms of resulting deviation at x_t and thus only sees $\hat{g}_{x_t}(x_t)$. From the point of view of the predictor, the best response strategy is obtained by

$$\min_{f \in \mathcal{F}} \sum_{(x_t, y_t) \in \mathcal{D}} \mathcal{L}(f(x_t), y_t) + \lambda \cdot d(f(x_t), \hat{g}_{x_t}(x_t)). \quad (3)$$

Discussion. We can analyze the scenario when the loss and deviation are measured in squared error, the explainer is in constant family, and the predictor is non-parametric. Both games induce a predictor that is equal to recursive convolutional average of y_t, where the decay rate in each recursion is the same $\frac{\lambda}{1+\lambda}$ for both games, but the convolutional kernel evolves twice faster in the symmetric game than in the asymmetric game.

The formulation involves a key trade-off between the size of the region where explanation should be simple and the overall accuracy achieved with the predictor. When the neighborhood is too small, local explainers become perfect, inducing no regularization on f. Thus the size of the region is a key parameter in our framework. Another subtlety is that solving (1) requires optimization over explainer family $\mathcal{G}$, where specific deviation and family choices matter for efficiency. For example, L_2 and affine family lead to linear regression with closed-form local explainers. Finally, a natural extension to solving (1) is to add regularizations.

We remark that the setup of f and $\mathcal{G}$ leaves considerable flexibility in tailoring the predictive model and the explanations to the application at hand — which is not limited to temporal modeling. Indeed, the derivation is for temporal models but extends naturally to others as well.

3. Examples

Conditional Generative Model. The basic idea of sequence modeling can be generalized to conditional sequence generation. For example, given historical data of a stock's price, how will the price evolve over time? Such mechanism allows us to inspect the temporal dynamics inside the problem of interest to assist in long-term decisions, while a conditional generation allows us to control the progression of generation with different settings of interest.

Formally, given an observation sequence $x_1, \dots, x_t \in \mathbb{R}^N$, the goal is to estimate the probability $p(x_{t+1}, \dots, x_T | x_1, \dots, x_t)$ of future events $x_{t+1}, \dots, x_T \in \mathbb{R}^N$. For notational simplicity, we will use $x_{1:t}$ to denote the sequence of variables $x_1, \dots, x_t$. A popular approach to estimate this conditional probability is to train a conditional model by maximum likelihood (ML) (Van Den Oord et al., 2016). If we model the conditional distribution of x_{i+1} given $x_{1:i}$ as a multivariate Gaussian distribution with mean $\mu(\cdot)$ and covariance $\Sigma(\cdot)$, we can define the asymmetric game on $\mu(\cdot)$ by minimizing

$$\sum_{i=t}^{T-1} \Big[-\log \mathcal{N}(x_{i+1}; \mu(x_{1:i}), \Sigma(x_{1:i})) + \lambda \left\| \mu(x_{1:i}) - g_{x_{1:i}}(x_{1:i}) \right\|_2^2 \Big], \quad (4)$$

with respect to $\mu(\cdot)$ and $\Sigma(\cdot)$, both parametrized as recurrent neural networks. For the explainer model $g_{x_{1:i}}(\cdot)$, we use the neighborhood data $\{(x_{1:i-\epsilon}, \mu(x_{1:i-\epsilon})), \cdots, (x_{1:i+\epsilon}, \mu(x_{1:i+\epsilon}))\}$ to fit a K-order Markov autoregressive (AR) model:

$$g_{x_{1:i}}(x_{1:i}) = \sum_{k=0}^{K-1} \theta_{k+1} \cdot x_{i-k} + \theta_0, \quad (5)$$

where $\theta_k \in \mathbb{R}^{N \times N}, k = 1, \dots, K$ and $\theta_0 \in \mathbb{R}^N$. AR model is a generalization of linear model to temporal modeling and thus admits a similar analytical solution. The choice of Markov horizon K makes this model flexible and should be informed by the application at hand.

Explicit Interpretability Game. In some cases, we wish to articulate interpretable parts explicitly in the predictor f. For example, if we view the predictor as approximately locally linear, we could explicitly parameterize f in a manner that highlights these linear coefficients. To this end, in the temporal domain, we can explicate the locally linear assumption already in the parametrization of μ:

$$\mu(x_{1:i}) = \sum_{k=0}^{K-1} \hat{\theta}(x_{1:i})_{k+1} \cdot x_{i-k} + \hat{\theta}_0(x_{1:i}), \quad (6)$$

which we can write as $\hat{\theta}_{\text{AR}}(x_{1:i}) + \hat{\theta}_0(x_{1:i})$, where $\hat{\theta}$ and $\hat{\theta}_0$ are learned as recurrent networks. However, this explicit parameterization is relevant only if we further encourage them to act their part, i.e., that the locally linear part of $\mu(x_{1:i})$ is really expressed by $\hat{\theta}(x_{1:i})_k, k = 1, \dots, K$.

To this end, we formulate a refined game that defines the discrepancy measure for the explainer in a coefficient specific manner, separately for $\hat{\theta}_{\text{AR}}$ and $\hat{\theta}_0$ so as to locally mimic the AR and constant family, respectively. The objective of local explainer at $x_{1:i}$ with respect to $\hat{\theta}_{\text{AR}}$ and $\hat{\theta}_0$ then becomes

$$\min_{g_{x_{1:i}} \in \mathcal{G}_{\text{AR}}} \frac{1}{2\epsilon+1} \sum_{i'=i-\epsilon}^{i+\epsilon} \| \hat{\theta}_{\text{AR}}(x_{1:i'}), g_{x_{1:i}}(x_{1:i'}) \|_2^2 + \min_{\bar{g}_{x_{1:i}} \in \mathcal{G}_C} \frac{1}{2\epsilon+1} \sum_{i'=i-\epsilon}^{i+\epsilon} \| \hat{\theta}_0(x_{1:i'}), \bar{g}_{x_{1:i}} \|_2^2, \quad (7)$$

where $\mathcal{G}_{\text{AR}}$ is the family of AR models (which does not include any offset/bias, consistent with $\hat{\theta}_{\text{AR}}$) and $\mathcal{G}_C$ is simply the set of constant vectors. The objective for f is defined analogously, symmetrically or asymmetrically. For simplicity, our notation doesn't include end-point boundary cases with respect to the neighborhoods.

Dataset	# of seq.	input len.	output len.
Stock	15,912	30	7
Bearing	200,736	80	20

Table 1. Dataset statistics

Stock	Error	Deviation	TV
AR	1.557	0.000	0.000
game-implicit	1.478	0.427	0.000
deep-implicit	1.472	0.571	0.000
game-explicit	1.479	0.531	73.745
deep-explicit	1.475	0.754	91.664
Bearing	**Error**	**Deviation**	**TV**
AR	9.832	0.000	0.000
game-implicit	8.309	3.431	5.706
deep-implicit	8.136	4.197	7.341
game-explicit	8.307	4.177	27.533
deep-explicit	8.151	6.134	29.756

Table 2. Performance. All units are in 10^{-2}.

4. Experiments

We validate our approach on two real-world applications: a bearing dataset from NASA (Lee et al., 2016) and a stock market dataset consisting of historical data from the S&P 500 index[1]. The bearing dataset records 4-channel acceleration data on 4 co-located bearings, and stock dataset records daily prices (4 channels in open, high, low, and close). Due to the diverse range of stock prices, we transform the data to daily percentage difference. We divide the sequence into disjoint subsequences and train the sequence generative model on them. The input and output length are decided based on the nature of dataset. The bearing dataset has a high frequency period of 5 points and low frequency period of 20 points. On the stock dataset, we used 1 month to predict the next week's prices. The statistics of the processed dataset is shown in Table 1. We randomly sample 85%, 5%, and 10% of the data for training, validation, and testing.

We set neighborhood ϵ and Markov order K to be 6 and 7 to impose sequence-level coherence in stock dataset; and 9 and 2 for smooth variation in bearing dataset. We parametrize $\mu(\cdot)$ and $\Sigma(\cdot)$ jointly by stacking 1 layer of CNN, LSTM, and 2 fully connected layers. We use Ridge regression[2] with default parameter in `scikit-learn` (Pedregosa et al., 2011) to implement the AR model. For efficiency, 10% of the sequences are sampled for regularization in each batch.

We compare our asymmetric game-theoretic approach ('game') against the same model class without an explainer ('deep'). We use 'implicit' label to distinguish predictors from those 'explicitly' written in an AR-like form. Evaluation involves three different types of errors: 'Error' is the root mean squared error (RMSE) between greedy autoregressive generation and the ground truth; 'Deviation' is RMSE between the model prediction $\mu(x_{1:i})$ and the explainer $g_{x_{1:i}}(x_{1:i})$, estimated also for 'deep' that is not guided by the explainer; and 'TV' is the average total variation of $[\hat{\theta}, \hat{\theta}_0]$ over any two consecutive time points. For testing, the explainer is estimated based on a greedy autoregressive generative trajectory. TV for implicit models is based on the parameters of the explainer g_{x_t}. For explicit models, deviation RMSE is the sum of AR and constant deviations as in Eq. (7), thus not directly comparable to the implicit 'Deviation' based only on output differences.

[1] www.kaggle.com/camnugent/sandp500/data

[2] Ridge is used to alleviate degenerate cases of linear regression.

$\times 10^{-2}$	$\lambda = 0.$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$	AR
Error	8.136	8.057	8.309	9.284	9.832
Deviation	4.197	4.178	3.431	1.127	0.000
TV	7.341	7.197	5.706	1.177	0.000

Table 3. Implicit game on bearing dataset when λ varies.

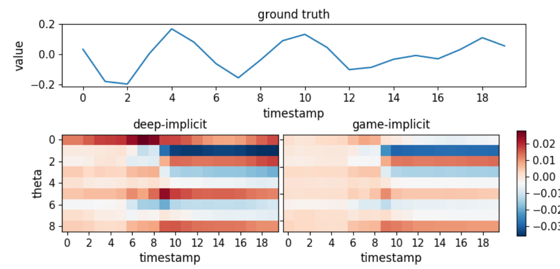


Figure 2. Visualization of weight vectors for predicting the first channel along each autoregressive timestamp in bearing dataset. The y-axis from 0 to 8 denotes $(\theta_0)_1$, $(\theta_1)_{1,1:4}$ and $(\theta_2)_{1,1:4}$.

The results are shown in Table 2. TV is not meaningful for implicit formulation in stock dataset due to the size of the neighborhood. The proposed game reduces the gap between deep learning model and AR model on deviation and TV, while retaining promising prediction accuracy.

We also show results of implicit setting on bearing dataset for different λ in Table 3. The trends in increasing error and decreasing deviation and TV are quite monotonic with λ. When $\lambda = 0.1$, the 'game' model is even more accurate than 'deep' model due to regularization effect of the game.

We visualize the explanations from implicit setting ($[\theta_0, \theta]$ in explainer model) over autoregressive generative trajectories in Figure 2. The explanation from the 'game' model is more stable. Compared to the ground truth, different temporal pattern after the 9th point is captured by the explanations.

5. Conclusion

We provide a novel game-theoretic approach to interpretable temporal modeling. The game articulates how the predictor accuracy can be traded off against locally agreeing with a simple axiomatically interpretable explainer. The work opens up many avenues for future work, from theoretical analysis of the co-operative games to estimation of interpretable unfolded trajectories through GANs.

Appendix

To analyze the behavior of game, we assume f is non-parametric, $d(\cdot,\cdot)$ and $\mathcal{L}(\cdot,\cdot)$ are squared error. Then here we do some analysis of the game with respect to linear $\mathcal{G}_{\text{Linear}}$ and constant $\mathcal{G}_C$ family.

To simplify derivation, we use a discrete time function notation: $\mathcal{D} = \{\cdots, (x_i, y_i), \cdots\}$. The derivation for a general setting is similar with minor adjustment to the definition of neighborhood. Then the objective for f in the symmetric game can be described as:

$$\min_f \sum_i (f(x_i) - y_i)^2 + \frac{\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} (f(x_t) - g_i(x_t))^2, \quad (8)$$

where $N = 2\epsilon + 1$ is the size of neighborhood. Since $f(x_t)$ is non-parametric and the objective in convex in it, we can treat it as a distinct variable, and use derivative to find its optimum:

$$f_S^*(x_i) = \frac{1}{1+\lambda}(y_i + \frac{\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i)) \quad (9)$$

The objective for f in the asymmetric game can be described as:

$$\min_f \sum_i (f(x_i) - y_i)^2 + \lambda(f(x_i) - g_i(x_i))^2, \quad (10)$$

The corresponding optimum is:

$$f_A^*(x_i) = \frac{1}{1+\lambda}(y_i + \lambda g_i(x_i)) \quad (11)$$

For both games, the objective for g_i can be described as:

$$\min_{g_i} \frac{\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} (f(x_t) - g_i(x_t))^2, \quad (12)$$

5.1. Linear Family $\mathcal{G}_{\text{Linear}}$

If $\mathcal{G}$ is a linear family $\mathcal{G}_{\text{Linear}}$, then $g_i(x_t) = x_t^T \theta_i$. We use X_i to denote the matrix $[x_{i-\epsilon}, \cdots, x_{i+\epsilon}]^T$, and $f(X_i)$ to denote the vector $[f(x_{i-\epsilon}), \cdots, f(x_{i+\epsilon})]^T$. Then we know the solution of g_i is the following:

$$g_i^* = X_i^\dagger f(X_i), \quad (13)$$

where † denotes pseudo-inverse. By feeding (13) into (9) and (11), we obtain a recursive optimization procedure to optimize (9) and (11). The form suggests that despite the local nature of regularization, the game affects the solution of $f(x_i)$ in a rate of $\frac{\lambda}{1+\lambda}$ from its neighborhood recursively throughout the whole space in both symmetric and asymmetric game.

If we denote Y_i as $[y_{i-\epsilon}, \cdots, y_{i+\epsilon}]^T$ and further assume $(\frac{\lambda}{1+\lambda})^2 \approx 0$, then the optimum of asymmetric game is

$$f_A^*(x_i) = \frac{y_i}{1+\lambda} + \frac{\lambda}{(1+\lambda)^2}(X_i^\dagger Y_i)x_i. \quad (14)$$

Similarly, the optimum of symmetric game is

$$f_S^*(x_i) = \frac{y_i}{1+\lambda} + \frac{\lambda}{N(1+\lambda)^2}(\sum_{t=1-\epsilon}^{1+\epsilon} X_t^\dagger Y_t)x_i. \quad (15)$$

5.2. Constant Family $\mathcal{G}_{\mathbf{C}}$

The optimal $g_i(\cdot)$ is

$$g_i(\cdot) = \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} f(x_t). \quad (16)$$

In this case, the optimal asymmetric game becomes:

$$f_A^*(x_i) = \frac{y_i}{1+\lambda} + \frac{\lambda}{1+\lambda}\frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} f_A^*(x_t) \quad (17)$$

$$= \frac{y_i}{1+\lambda} + \frac{\lambda}{(1+\lambda)^2}\frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} y_t + \frac{\lambda^2}{(1+\lambda)^3}\frac{1}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} y_{t_2} + \frac{\lambda^3}{(1+\lambda)^4}\frac{1}{N^3} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} \sum_{t_3=t_2-\epsilon}^{t_2+\epsilon} y_{t_3} + ..., \quad (18)$$

which is a resursive convolutional average with a decay rate of $\frac{\lambda}{1+\lambda}$.

Similarly, for symmetric game, the optimum is:

$$f_S^*(x_i) = \frac{y_i}{1+\lambda} + \frac{\lambda}{1+\lambda}\frac{1}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} f_S^*(x_{t_2}) \quad (19)$$

$$= \frac{y_i}{1+\lambda} + \frac{\lambda}{(1+\lambda)^2}\frac{1}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} y_{t_2} + \frac{\lambda^2}{(1+\lambda)^3}\frac{1}{N^4} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} \sum_{t_3=t_2-\epsilon}^{t_2+\epsilon} \sum_{t_4=t_3-\epsilon}^{t_1+\epsilon} y_{t_4} + ... \quad (20)$$

By comparing (18) with (20), we know both formulation establish a recursive convolutional average with the same decay rate, but symmetric game has a convolutional filter that evolves twice faster than the asymmetric game. As a result, the symmetric game has a smoother solution than the asymmetric game, as applying convolutional averaging will always make the input smoother.

5.3. Symmetry Recovery for Asymmetric Game

Constant Class. We note that if we properly adjust the construction of local deviation, the adjusted asymmetric game recover the original symmetric game for constant family. Here we define the (weighted) local deviation for adjust asymmetric game as:

$$\min_{g_i} \frac{\lambda}{N^2} \sum_{t=i-2\epsilon}^{i+2\epsilon} (2\epsilon + 1 - |t-i|)(f(x_t) - g_i(x_t))^2 \tag{21}$$

$$= \min_{g_i} \frac{\lambda}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} (f(x_t) - g_i(x_t))^2. \tag{22}$$

If the explainer is in constant family, the optimal explainer becomes

$$g_i(x_t) = \frac{1}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} f(x_{t_2}), \tag{23}$$

The optimal predictor for adjusted asymmetric game becomes

$$f_A^*(x_i) = \frac{y_i}{1+\lambda} + \frac{\lambda}{1+\lambda} \frac{1}{N^2} \sum_{t_1=i-\epsilon}^{i+\epsilon} \sum_{t_2=t_1-\epsilon}^{t_1+\epsilon} f_A^*(x_{t_2}), \tag{24}$$

which clearly recovers the original symmetric game in (20).

Alternative Perspective The key factor to induce the above equivalence is achieved by introducing weighting on the local data to fit the explainer. Alternatively, we can establish an *asymmetric strategies* to achieve the same result. Here we further generalize constant family to an abstract family $\mathcal{G}$

For each local explainer, the strategy is still to recover the best explanation to f as the original local deviation formulation (12). However, for the predictor, the strategy is to achieve a balance between ground truth label and an *aggregated* explanations:

$$\min_f \sum_i \left[(f(x_i) - y_i)^2 + \lambda(f(x_i) - \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i))^2 \right]. \tag{25}$$

As a result, the best response strategy of f at any x_i becomes:

$$f_A^*(x_i) = \frac{1}{1+\lambda} y_i + \frac{\lambda}{1+\lambda} (\frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i)), \tag{26}$$

which recovers the original symmetric game (9). In the following sections, we based on the same principle of *asymmetric strategies* to recover symmetric game.

5.4. When Non-parametric does not hold

We note that the above analysis are based on an assumption that f is non-parametric. Here are analyze scenarios when the property does not hold with concrete machine learning models. In addition, we further relaxed the assumption that the loss is squared error.

Decision Tree. If f refers to a decision tree with limited depth that cannot perfectly fit the data. Without loss of generality, we can assume that there are at least two x_i, x_j such that $f(x_i)$ and $f(x_j)$ belongs to the same split with $y_i \neq y_j$. Note that such assumption applies to both classification and regression trees, and we discuss the optimal $f(x_i)$ and $f(x_j)$ under such split. The conclusion can be trivially generalized to more points within a split.

As a result, we can share the two variables $f(x_i) = f(x_j)$ with the same value $\bar{f}$, and derive the equilibrium under the symmetric game and adjusted asymmetric game. The symmetric game for $\bar{f}$ can be presented as

$$\min_{\bar{f}} \mathcal{L}(\bar{f}, y_i) + \mathcal{L}(\bar{f}, y_j) + \frac{\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} (\bar{f} - g_t(x_i))^2 + \frac{\lambda}{N} \sum_{l=j-\epsilon}^{j+\epsilon} (\bar{f} - g_l(x_j))^2. \tag{27}$$

The adjusted asymmetric game can be presented as

$$\min_{\bar{f}} \mathcal{L}(\bar{f}, y_i) + \mathcal{L}(\bar{f}, y_j) + \lambda(\bar{f} - \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i))^2 + \lambda(\bar{f} - \frac{1}{N} \sum_{l=j-\epsilon}^{j+\epsilon} g_l(x_j))^2, \tag{28}$$

which clearly recovers the same equilibrium as (27).

Differentiable Function

We assume f to be a differentiable function parametrized by θ: $f_\theta(\cdot)$. Then the optimality condition of symmetric game is

$$\nabla_\theta \sum_i \mathcal{L}(f_\theta(x_i), y_i) + \frac{\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} (f_\theta(x_t) - g_i(x_t))^2 \quad (29)$$

$$= \sum_i \Big[\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$$

$$+ \frac{2\lambda}{N} \sum_{t=i-\epsilon}^{i+\epsilon} (f_\theta(x_t) - g_i(x_t)) \nabla_\theta f_\theta(x_t) \Big] = 0. \quad (30)$$

Similarly, for adjusted asymmetric game, the optimality condition of the adjusted asymmetric game is

$$\nabla_\theta \sum_i \Big[\mathcal{L}(f_\theta(x_i), y_i) + \lambda (f_\theta(x_i) - \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i))^2 \Big] \quad (31)$$

$$= \sum_i \Big[\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$$

$$+ 2\lambda (f_\theta(x_i) - \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i)) \nabla_\theta f_\theta(x_i) \Big] = 0. \quad (32)$$

By noting that

$$\sum_i 2\lambda (f_\theta(x_i) - \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i)) \nabla_\theta f_\theta(x_i) \quad (33)$$

$$= \sum_i \sum_{t=i-\epsilon}^{i+\epsilon} \frac{2\lambda}{N} (f_\theta(x_i) - g_t(x_i)) \nabla_\theta f_\theta(x_i) \quad (34)$$

$$= \sum_t \sum_{i=t-\epsilon}^{t+\epsilon} \frac{2\lambda}{N} (f_\theta(x_i) - g_t(x_i)) \nabla_\theta f_\theta(x_i) \quad (35)$$

$$= \sum_i \sum_{t=i-\epsilon}^{i+\epsilon} \frac{2\lambda}{N} (f_\theta(x_t) - g_i(x_t)) \nabla_\theta f_\theta(x_t), \quad (36)$$

we conclude the adjusted asymmetric game recovers the symmetric game with squared deviation.

Sufficient Condition to Symmetry Recovery

To generalize the above solution to general class of deviation function, we remark a sufficient condition for adjusted asymmetric game to recover symmetric game is to compose an alternative deviation function d' and learning target $\bar{g}(\cdot)$ as follows:

$$\nabla_\theta d'(f_\theta(x_i), \bar{g}(x_i)) \quad (37)$$

$$= \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} \nabla_\theta d(f_\theta(x_i), g_t(x_i)), \quad (38)$$

while $d' = d$ and $\bar{g}(x_i) = \frac{1}{N} \sum_{t=i-\epsilon}^{i+\epsilon} g_t(x_i)$ in the case of squared deviation.

Boundary Case

The above analysis is based on an implicit assumption that the sequence has infinite length from $-\infty$ to ∞. However, when the sequence length is limited, it requires further adjustment to ensure symmetry recovery. Here we still focus on the case of squared deviation for simplicity.

We use $B_\epsilon(i)$ to denote the neighborhood of i, and use N_i to denote $|B_\epsilon(i)|$. The optimality condition for symmetric game is

$$\nabla_\theta \sum_{i=1}^{T} \mathcal{L}(f_\theta(x_i), y_i) + \frac{\lambda}{N_i} \sum_{t \in B_\epsilon(i)} (f_\theta(x_t) - g_i(x_t))^2 \quad (39)$$

$$= \sum_{i=1}^{T} \Big[\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$$

$$+ \frac{2\lambda}{N_i} \sum_{t \in B_\epsilon(i)} (f_\theta(x_t) - g_i(x_t)) \nabla_\theta f_\theta(x_t) \Big] = 0. \quad (40)$$

In this case, to recover symmetric game, we design a different asymmetric deviation measures that captures the boundary case. We use $\bar{N}_i$ to denote $\sum_{t \in B_\epsilon(i)} \frac{1}{N_t}$. Then the asymmetric deviation measure for *predictor* is (note that explainer always use the original deviation measure):

$$d(f(x_i), \bar{g}(x_i)) = \frac{1}{\bar{N}_i} (\bar{N}_i f(x_i) - \sum_{x_t \in B_\epsilon(x_i)} \frac{g_t(x_i)}{N_t})^2 \quad (41)$$

As a result, we can see that the optimality condition for adjusted asymmetric game becomes:

$$\nabla_\theta \sum_{i=1}^{T} \Big[\mathcal{L}(f_\theta(x_i), y_i) + \frac{\lambda}{\bar{N}_i} (\bar{N}_i f(x_i) - \sum_{x_t \in B_\epsilon(x_i)} \frac{g_t(x_i)}{N_t})^2 \Big] \quad (42)$$

$$= \sum_{i=1}^{T} \Big[\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$$

$$+ 2\lambda (\bar{N}_i f_\theta(x_i) - \sum_{t \in B_\epsilon(i)} \frac{g_t(x_i)}{N_t}) \nabla_\theta f_\theta(x_i) \Big] \quad (43)$$

$$= \sum_{i=1}^{T} \Big[\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$$

$$+ \sum_{t \in B_\epsilon(i)} \frac{2\lambda}{N_t} (f_\theta(x_i) - g_t(x_i)) \nabla_\theta f_\theta(x_i) \Big] = 0. \quad (44)$$

Since by swapping the iteration orders of i and t, we have:

$$\sum_{i=1}^{T} \sum_{t \in B_\epsilon(i)} \frac{2\lambda}{N_t} (f_\theta(x_i) - g_t(x_i)) \nabla_\theta f_\theta(x_i) \tag{45}$$

$$= \sum_{t=1}^{T} \sum_{i \in B_\epsilon(t)} \frac{2\lambda}{N_t} (f_\theta(x_i) - g_t(x_i)) \nabla_\theta f_\theta(x_i), \tag{46}$$

which suggests the equivalence of optimality condition between symmetric game (40) and asymmetric game (44) Hence, we conclude that the symmetric game can be recovered by adjusted asymmetric game.

References

Al-Shedivat, Maruan, Dubey, Avinava, and Xing, Eric P. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.

Alvarez-Melis, David and Jaakkola, Tommi S. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *Proceedings of EMNLP*, 2017.

Arras, Leila, Horn, Franziska, Montavon, Grégoire, Müller, Klaus-Robert, and Samek, Wojciech. " What is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Lakkaraju, Himabindu, Bach, Stephen H, and Leskovec, Jure. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1675–1684. ACM, 2016.

Lee, J., Qiu, H., Yu, G., Lin, J., and Rexnord Technical Services (2007). IMS, University of Cincinnati. Bearing data set. *NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*, 7(8), 2016.

Lei, Tao, Barzilay, Regina, and Jaakkola, Tommi. Rationalizing Neural Predictions. In *EMNLP 2016, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016. URL http://arxiv.org/abs/1606.04155.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL http://arxiv.org/abs/1602.04938http://doi.acm.org/10.1145/2939672.2939778.

Selvaraju, Ramprasaath R, Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, and Batra, Dhruv. Grad-cam: Visual explanations from deep networks via gradient-based localization. *See https://arxiv.org/abs/1610.02391 v3*, 7(8), 2016.

Van Den Oord, Aaron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Wu, Mike, Hughes, Michael C., Parbhoo, Sonali, Zazzi, Maurizio, Roth, Volker, and Doshi-Velez, Finale. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16285.