Recent Works on Feature Interaction

Xingzhi Sun

https://github.com/xingzhis/XAI

Wednesday 30th September, 2020

Table of Contents

Table of Contents

Non-additiveness

Post-hoc statistic for interaction

Non-additiveness

$F(\mathbf{x})$ cannot be written in the form of $F(\mathbf{x}) = f_j(x_j) + f_{\backslash j}(\mathbf{x}_{\backslash j})$

Non-additiveness

Post-hoc statistic for interaction

Non-additiveness

$F(\mathbf{x})$ cannot be written in the form of $F(\mathbf{x}) = f_j(x_j) + f_{\backslash j}(\mathbf{x}_{\backslash j})$

the effect of both variables

vs

the sum of effects of each variable

Non-additiveness

Post-hoc statistic for interaction

Non-additiveness

$F(\mathbf{x})$ cannot be written in the form of $F(\mathbf{x}) = f_j(x_j) + f_{\backslash j}(\mathbf{x}_{\backslash j})$

$$\boxed{\text{the effect of both variables}}$$
$$\text{vs}$$
$$\boxed{\text{the sum of effects of each variable}}$$

let $F_s(\mathbf{x}_s) = \mathbb{E}_{\mathbf{x}_{\backslash s}}\left[F\left(\mathbf{x}_s, \mathbf{x}_{\backslash s}\right)\right]$

Predictive learning via rule ensembles

$$H_{jk}^2 = \sum_{i=1}^N \left[\hat{F}_{jk}(x_{ij}, x_{ik}) - \hat{F}_j(x_{ij}) - \hat{F}_k(x_{ik})\right]^2 / \sum_{i=1}^N \hat{F}_{jk}^2(x_{ij}, x_{ik})$$

Model Expressiveness

Post-hoc statistic for interaction

the **accurate** model that contains the interaction

vs

the **accurate** model that bans the interaction

Model Expressiveness

Post-hoc statistic for interaction

$\boxed{\text{the **accurate** model that contains the interaction}}$

vs

$\boxed{\text{the **accurate** model that bans the interaction}}$

$F^*(x)$: Target function

$F(x)$: Highly accurate model

$R_{ij}(x)$: Highly accurate but ban interaction between x_i and x_j

$$\mathrm{stRMSE}(F(\mathbf{x})) = \frac{\mathrm{RMSE}(F(\mathbf{x}))}{\mathrm{StD}(F^*(\mathbf{x}))},$$

Detecting statistical interactions with additive groves of trees

$$I_{ij}(F(\mathbf{x})) = \mathrm{stRMSE}(F(\mathbf{x})) - \mathrm{stRMSE}\left(R_{ij}(\mathbf{x})\right)$$

Hessian - second derivative

Post-hoc statistic for interaction

The prediction has non-zero hessian over the interaction variables.

$$\frac{\partial F(\mathbf{x})}{\partial x_i \partial x_j} \neq 0$$

Hessian - second derivative

Post-hoc statistic for interaction

The prediction has non-zero hessian over the interaction variables.

$$\frac{\partial F(\mathbf{x})}{\partial x_i \partial x_j} \neq 0$$

Learning Global Pairwise Interactions with Bayesian Neural Networks

$$\mathrm{EAH}_g^{i,j}(\mathbf{W}) = \mathbb{E}_{p(\mathbf{x})}\left[\left|\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\right|\right]$$

$$\mathrm{AEH}_g^{i,j}(\mathbf{W}) = \left|\mathbb{E}_{p(\mathbf{x})}\left[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\right]\right|$$

Table of Contents

Model-specific depiction of interaction

- Explicit: parameters, such as weights of interaction terms.
- Implicit: Neuron networks, embedding vectors, etc.

Table of Contents

Predictive learning via rule ensembles

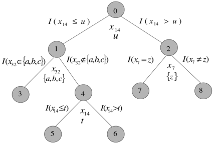
Tree-based ensembels

- Main effects: Linear
- Interaction: Rules derived from decision trees

Predictive learning via rule ensembles

Tree-based ensembels

- Main effects: Linear
- Interaction: Rules derived from decision trees



$$r_1(\mathbf{x}) = I(x_{14} \le u),$$
$$r_4(\mathbf{x}) = I(x_{14} \le u) \cdot I(x_{32} \notin \{a, b, c\}),$$
$$r_6(\mathbf{x}) = I(t < x_{14} \le u) \cdot I(x_{32} \notin \{a, b, c\}),$$
$$r_7(\mathbf{x}) = I(x_{14} > u) \cdot I(x_7 = z).$$

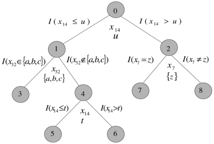
$$r_m(\mathbf{x}) = \prod_{s_{jm} \neq S_j} I(x_j \in s_{jm})$$

Figure: A desicion tree and its corresponding rule term

Predictive learning via rule ensembles

Tree-based ensembels

- Main effects: Linear
- Interaction: Rules derived from decision trees



$$r_1(\mathbf{x}) = I(x_{14} \leq u),$$
$$r_4(\mathbf{x}) = I(x_{14} \leq u) \cdot I(x_{32} \notin \{a, b, c\}),$$
$$r_6(\mathbf{x}) = I(t < x_{14} \leq u) \cdot I(x_{32} \notin \{a, b, c\}),$$
$$r_7(\mathbf{x}) = I(x_{14} > u) \cdot I(x_7 = z).$$

$$r_m(\mathbf{x}) = \prod_{s_{jm} \neq S_j} I(x_j \in s_{jm})$$

Figure: A desicion tree and its corresponding rule term

RuleFit (Friedman and Popescu, 2008)

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{k=1}^{K} \hat{a}_k r_k(\mathbf{x}) + \sum_{j=1}^{n} \hat{b}_j l_j(x_j)$$

Learn the model with regularized regression.

RuleFit (Friedman and Popescu, 2008)

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{k=1}^{K} \hat{a}_k r_k(\mathbf{x}) + \sum_{j=1}^{n} \hat{b}_j l_j\left(x_j\right)$$

RuleFit (Friedman and Popescu, 2008)

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{k=1}^{K} \hat{a}_k r_k(\mathbf{x}) + \sum_{j=1}^{n} \hat{b}_j l_j\left(x_j\right)$$

statistic for interaction

$$H_{jk}^2 = \sum_{i=1}^{N} \left[\hat{F}_{jk}\left(x_{ij}, x_{ik}\right) - \hat{F}_j\left(x_{ij}\right) - \hat{F}_k\left(x_{ik}\right)\right]^2 / \sum_{i=1}^{N} \hat{F}_{jk}^2\left(x_{ij}, x_{ik}\right)$$

Detect interaction with significant H_{jk}, whose distribution is obtained by bootstrapping.

the **accurate** model that contains the interaction

vs

the **accurate** model that bans the interaction

statistic for interaction

$$I_{ij}(F(\mathbf{x})) = \mathrm{stRMSE}(F(\mathbf{x})) - \mathrm{stRMSE}\left(R_{ij}(\mathbf{x})\right)$$

the **accurate** model that contains the interaction

vs

the **accurate** model that bans the interaction

statistic for interaction

$$I_{ij}(F(\mathbf{x})) = \mathrm{stRMSE}(F(\mathbf{x})) - \mathrm{stRMSE}\left(R_{ij}(\mathbf{x})\right)$$

- Obtain a highly accurate model through a tree ensemble that is later bagged: $F_0(x) = \sum_{i=1}^{K} T_i(x)$
- Restrict interaction by forbidding one of the interacting variables when growing a tree.

Detecting statistical interactions with additive groves of trees

Tree-based ensembels

the **accurate** model that contains the interaction

vs

the **accurate** model that bans the interaction

statistic for interaction

$$I_{ij}(F(\mathbf{x})) = \mathrm{stRMSE}(F(\mathbf{x})) - \mathrm{stRMSE}\left(R_{ij}(\mathbf{x})\right)$$

- Obtain a highly accurate model through a tree ensemble that is later bagged: $F_0(x) = \sum_{i=1}^{K} T_i(x)$
- Restrict interaction by forbidding one of the interacting variables when growing a tree.

Beats the Rulefit statistic in avoiding spurious interaction at sparse regions.

Table of Contents

- Main effects: GAM.
- Interaction: 2D bin functions f_{ij} on the residual of GAM.

- Main effects: GAM.
- Interaction: 2D bin functions f_{ij} on the residual of GAM.

FAST

Speed up the calculation of bin averages: pre-caluclate a CDF lookup table for reusing.

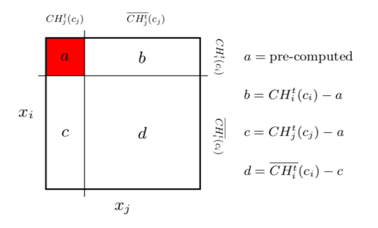


Table of Contents

Definition of feature interaction for smooth models

The prediction has non-zero hessian over the interaction variables.

$$\frac{\partial F(\mathbf{x})}{\partial x_i \partial x_j} \neq 0$$

Definition of feature interaction for smooth models

The prediction has non-zero hessian over the interaction variables.

$$\frac{\partial F(\mathbf{x})}{\partial x_i \partial x_j} \neq 0$$

In practice, we take the expectation of the hessian over the distribution of $\mathbf{x}$

Two ways of taking expectation

$$\mathrm{EAH}_g^{i,j}(\mathbf{W}) = \mathbb{E}_{p(\mathbf{x})}\left[\left|\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\right|\right]$$ lowest FNR, highest FPR.

$$\mathrm{AEH}_g^{i,j}(\mathbf{W}) = \left|\mathbb{E}_{p(\mathbf{x})}\left[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j}\right]\right|$$ lowest FPR, highest FNR.

- EAH avoids $(+,-)$ noise to cancel and could capture spurious interactions.
- AEH could have true interactions cancel out and fail to capture true interactions.

strike a balance: Group Expected Hessian (GEH)

$$M\text{-GEH}_g^{i,j}(\mathbf{W}) = \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M} |A_k|} \left| \mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)} \left[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j} \right] \right|$$

Partition the datapoints into M clusters, and expect the interaction is simmilar within each cluster, where only the noise is canceled out.

strike a balance: Group Expected Hessian (GEH)

$$M\text{-GEH}_g^{i,j}(\mathbf{W}) = \sum_{m=1}^{M} \frac{|A_m|}{\sum_{k=1}^{M}|A_k|} \left| \mathbb{E}_{p(\mathbf{x}|\mathbf{x}\in A_m)} \left[\frac{\partial^2 g^{\mathbf{W}}(\mathbf{x})}{\partial x_i \partial x_j} \right] \right|$$

Partition the datapoints into M clusters, and expect the interaction is simmilar within each cluster, where only the noise is canceled out.

Bayesian NN allows for the distribution of the M-GEH statistic and thus mean, std, confidence intervals, etc.

How does a neuron network capture an interaction?

- Features share units of the first hidden layer.
- The shared units are passed to the output through descendent edges.

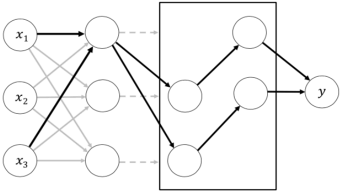


Figure: Interaction in an NN

Write in the form of NN weights:

NID (Tsang et al., 2018)

$$\begin{aligned} \mathbf{z}^{(\ell)} &= \left|\mathbf{w}^{y}\right|^{\top} \left|\mathbf{W}^{(L)}\right| \cdot \left|\mathbf{W}^{(L-1)}\right| \dots \left|\mathbf{W}^{(\ell+1)}\right| \\ \omega_i(\mathcal{I}) &= z_i^{(1)} \mu\left(\left|\mathbf{W}_{i,\mathcal{I}}^{(1)}\right|\right) \end{aligned}$$

Write in the form of NN weights:

NID (Tsang et al., 2018)

$$
\begin{aligned}
\mathbf{z}^{(\ell)} &= \left|\mathbf{w}^{y}\right|^{\top} \left|\mathbf{W}^{(L)}\right| \cdot \left|\mathbf{W}^{(L-1)}\right| \ldots \left|\mathbf{W}^{(\ell+1)}\right| \\
\omega_i(\mathcal{I}) &= z_i^{(1)} \mu\left(\left|\mathbf{W}_{i,\mathcal{I}}^{(1)}\right|\right)
\end{aligned}
$$

The algorithm is fast because only the features with top weights is considered in each iteration.

- The first hidden layer captures first-order interactions.
- The following layers capture higher-order interactions.
- But they entangle and could contain spurious interaction: $x_1x_2 + x_3x_4 \rightarrow x_1, x_2, x_3, x_4$ instead of x_1, x_2 and x_3, x_4

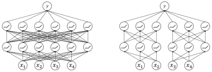


Figure: Entangled vs disentangeled

Neural Interaction Transparency: Disentangling Learned Interactions for Improved Interpretability

Neuron Networks

- The first hidden layer captures first-order interactions.
- The following layers capture higher-order interactions.
- But they entangle and could contain spurious interaction: $x_1x_2 + x_3x_4 \rightarrow x_1, x_2, x_3, x_4$ instead of x_1, x_2 and x_3, x_4

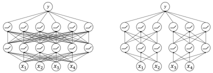
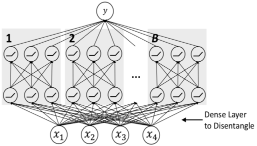


Figure: Entangled vs disentangeled

Solution: Add a penalty on the weight matrix to control maximum allowed order of interaction.



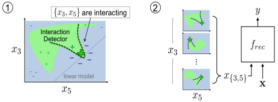
Given a black-box model, how to interprete global interaction?

- detect local interaction
- count the local interactions.
- interactions with many occurances are interpreted as global.

Recent Works on Feature Interaction
Xingzhi Sun
Outline
Definitions of feature interaction
Post-hoc statistic
Model-specific
Models
Tree-based ensembels
GA^{2}M
Neuron Networks
Factorization Machines
Hybrid models with neuron networks
Summary

Given a black-box model, how to interprete global interaction?

- detect local interaction
- count the local interactions.
- interactions with many occurances are interpreted as global.



GLIDER (Tsang et al., 2020)

1. For each data instance $\mathbf{x}$, purturb to get a local dataset, and predict on that dataset.
2. on the local dataset, detect interaction with NID.
3. count and rank occurances of each interaction.

Table of Contents

Factorization machines

Factorization Machines

Background: Highly-sparse categorical data with one-hot coding. e.g. Shopping history, movie reviews.

Factorization machines

Factorization Machines

Background: Highly-sparse categorical data with one-hot coding. e.g. Shopping history, movie reviews.

FM (Rendle, 2010)

$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i x_j$
learn the parameters with gradient descent

Each feature x_i corresponds to an *embedding vector* $\mathbf{v}_i$.
Interaction: the inner products of the vectors $< \mathbf{v}_i, \mathbf{v}_j >$.

Factorization machines

Factorization Machines

Background: Highly-sparse categorical data with one-hot coding. e.g. Shopping history, movie reviews.

FM (Rendle, 2010)

$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$
learn the parameters with gradient descent

Each feature x_i corresponds to an *embedding vector* $\mathbf{v}_i$.
Interaction: the inner products of the vectors $< \mathbf{v}_i, \mathbf{v}_j >$.

Advantages over regression (polynomial kernel SVM):

- Generalizes to instances that do not appear in the training set.
- linear complexity.

Variants of FM

Factorization Machines

Field-aware Factorization Machines for CTR Prediction

1. Categorize features into fields: Clothes, Food, Electronics,...
2. For each feature, instead of one embedding vector, learn a vector for each field.
3. When calculating interaction, use the vectors matching each other's field to take inner product.
   $\phi_{\mathrm{FFM}}(\boldsymbol{w}, \boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\boldsymbol{w}_{i,f_2} \cdot \boldsymbol{w}_{j}, f_1\right) x_i x_j$

Attentional Factorization Machines

Instead of taking the inner product of the embedding vectors, take weighed outer product with weights learned by an Attention Neuron Network.

Table of Contents

Wide & Deep Learning for Recommender Systems

Hybrid models with neuron networks

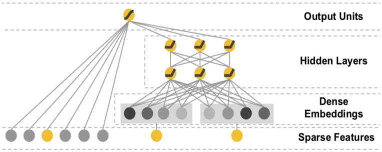


Figure: Wide & Deep

A **jointly-trained** hybrid model:

- Linear crossing model: *manual* low-order interactions.
- Deep model: *automatic* high-order interactions.

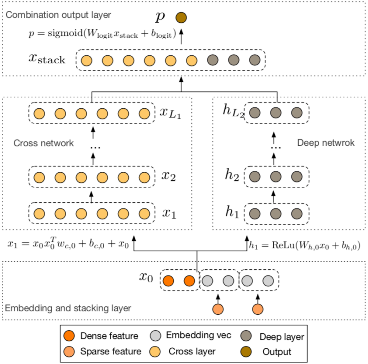


Figure: Deep & Cross

A **jointly-trained** hybrid model:

- Crossing network: *automatic* low-order interactions.
- Deep network: *automatic* high-order interactions.

DeepFM: A Factorization-Machine based Neural Network for CTR Prediction

Hybrid models with neuron networks

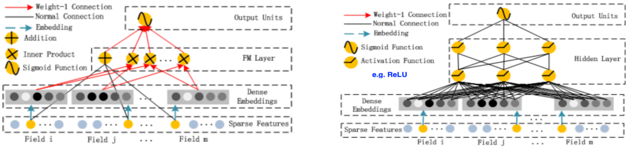


Figure: DeepFM

A **jointly-trained** hybrid model:

- FM: *automatic* low-order interactions.
- Deep model: *automatic* high-order interactions.

xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems

Hybrid models with neuron networks

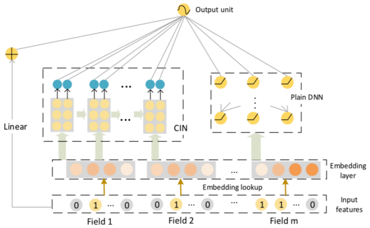


Figure: xDeepFM

A **jointly-trained** hybrid model:

- Linear model: main effects.
- Compressed Interaction Network: *automatic* low-order interactions: Interaction of each order goes to the output.
- Deep neuron network: *automatic* high-order interactions.

Deep Interest Network for Click-Through Rate Prediction

Attention Neuron Networks

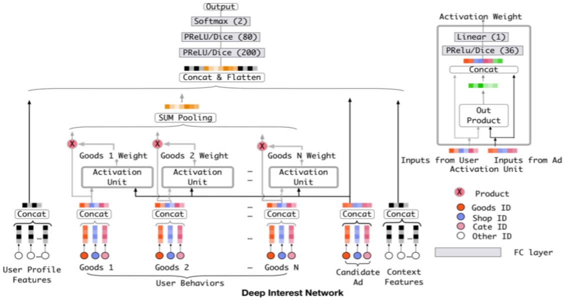


Figure: Deep Interest Network

The network comprises Activation Units, an attention network that leverages the user preference history and identify the true interest from diverse interests.

AutoInt: Automatic feature interaction learning via self-attentive neural networks

Attention Neuron Networks

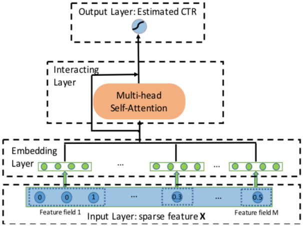


Figure: AutoInt

Each *multi-head self-attentive* layer captures interaction by learning interaction weights.
The model is a black-box with all the interactions entangled.

Summary

Models for capturing feature interactions

Models for capturing feature interactions:

- post-hoc
  - Tree Ensembles: non-additiveness, expressiveness
  - Neuron Networks: hessian, weights
- ad-hoc
  - Regularized NN: additive terms
  - GA2M: simple, fast
  - FM: sparse data, generalizaton, fast
  - Non-Deep and Deep hybrids: low-order and high-order