2.2 ML Interpretability in Healthcare

Johnathan Crabbe
jc2133@cam.ac.uk

Why do we need interpretability?

- Example of a typical ML problem:
  - Patient predicting mortality with a model
  - Model can be DL network (common model)
    - If clinician is skeptical of the prediction
  - Neural network in mathematical expression is complex
    - Not interpretable even for a small network
    - Recent year, modern models are even bigger
      - E.g. Language, Gaming models
- Problem with complex models
  - Modern ML are complex (esp DNNs)
  - Opacity of those models causes difficulties to humans
    - **Model makers**: Does it generalize well? If not, how to fix it?
    - **Model users**: How does it work? What regime can it be used?
    - **Scientist**: Accordance with science? What can we learn?
  - Is opacity unavoidable for complex models?
    - Probably not! Think about brain neuron – producing meaningful explanations

What is interpretability?

- Setup:
  - For interpretable functions:
    - Restrict to models, can be directly analyzed by humans
      - Concise models (e.g. decision trees)
      - Models that contextualize predictions (e.g. attention-based models)
    - NB: restriction might impact model's performance
  - Post-Hoc interpretability
    - Add module on top of black-box function
      - Module aware of input features and black-box function
      - Create explanation to convince users
    - A parallel with human brain
- Problem with complex model:
  - Two approaches:
    - By design: simplification
    - Post-hoc:
      - Feature based
      - Example based
      - Concept based

Feature based

- $f(x_1, x_2) = (x_1)^2 + \exp(x_2)$

    - What Feature in the couple *(x_1, x_2)* contributes most to f?
        - For $x_2 \geq x_1 >> 0$: $\exp(x_2)^2 \Rightarrow x_2$ is more important
        - For x_1 , x_2 <<0 : $\exp(x_2) \approx 0 << (x_1)^2 \Rightarrow x_1$ is more important
        - …
    - If *f* is nonlinear, there is no global conclusion
    - Gets worse when *f* depends on many features that interact (DNNs, etc)
    - Importance scores $a_i(f,x)$ depend on blackbox *f* and input *x*
- Examples:
    - Lime (https://arxiv.org/abs/1602.04938)
    - SHAP (https://arxiv.org/abs/1705.07874)
    - Integrated Gradients (https://arxiv.org/abs/1703.01365)
- Limitations:
    - first order – no interactions
    - DNNs are nonlinear functions of the input – no global importance
- SHAP

$$a_i(f,x) = \sum_{s\subset[d_X]\backslash\{\}} \frac{|S|!\,(d_X - |S| - 1)!}{d_X!}(f(x_s \cup x_i) - f(x_s))$$

    - https://arxiv.org/abs/1705.07874
    - Idea: important features impact the prediction when added on top of other features
        - Features are “removed” through marginalization
    - Pros: well motivated theoretically, lots of implementation
    - Cons: extremely expensive to compute exactly -> approximation required
- Integrated Gradient
    - Feature is important if black box heavily depends on it
    - Computing gradient at each point between baseline and x
        - Gradient will be great for higher importance
    - Baseline should reflect the absence of information (e.g. black image)
    - Pros: Inexpensive to compute, lots of implementation
    - Cons: Heavily dependent on the baseline choice, requires gradient information
        - Only work for differentiable inputs
- Masks
    - Finding the most important features is an optimization problem
    - Ref: **Interpretable explanation of black boxes by meaning perturbation**
        - https://arxiv.org/abs/1704.03296
    - Pros: optimisation permits to surface more impactful features
    - Cons: Require structure data (e.g. image/ time-series)
- Dynamask: Feature importance of time series:
    - Time series data is pervasive in medicine & finance
    - Most of the previous methods don’t generalize beyond tabular/image data
    - Dynamask leverages time dependency

- ○ https://arxiv.org/abs/2106.05303

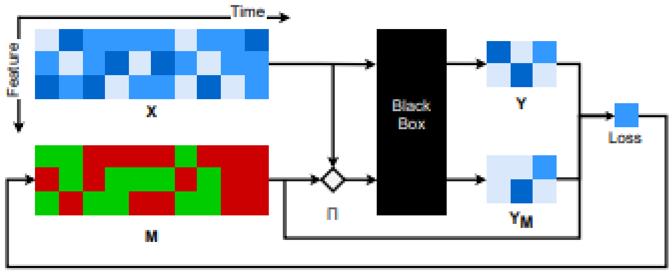

Figure 2. Diagram for Dynamask. An input matrix **X**, extracted from a multivariate time series, is fed to a black-box to produce a prediction **Y**. The objective is to give a saliency score for each component of **X**. In Dynamask, these saliency scores are stored in a mask **M** of the same shape as the input **X**. To detect the salient information in the input **X**, the mask produces a perturbed version of **X** via a perturbation operator Π. This perturbed **X** is fed to the black-box to produce a perturbed prediction $\mathbf{Y}_M$. The perturbed prediction is compared to the original prediction and the error is backpropagated to adapt the saliency scores contained in the mask.

- Applications of feature importance:
    - Isolating most important features helps to highlight model weakness
    - ML models are lazy and will exploit hidden confounders (e.g. Ribeiro et al., 2016)
    - Information can be exploited to benchmark treatment effect models (Crabbe et al. 2022)
    - Discovering patterns that are far from obvious for humans
    - Feature importance narrows down the study of those patterns (Davies et al., 2021)
        - https://www.nature.com/articles/s41586-021-04086-x

Example-based Explanations

- What: identify most important training examples for black-box predictions
- How: Attribute an importance score a^n to each training example (x^n, y^n) for black-box f
- Example:
    - Influence functions (https://proceedings.mlr.press/v70/koh17a.html)
    - TraceIN (https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html)
    - SimplEx (https://proceedings.neurips.cc/paper/2021/hash/65658fde58ab3c2b6e5132a39fae7cb9-Abstract.html)
- Limitations:
    - Approximation required for large datasets
    - Scores isolate individual examples -> ignore interactions
- Influence functions
    - https://proceedings.mlr.press/v70/koh17a.html
    - Inverse Heissian inner product simulates removal of training examples (x^n, y^n)
    - Important training examples increase loss when removed
    - Pros: does not require retraining, well motivated theoretically (asymptotic statistics)
    - Cons: Computing the Hessian inverse is expensive – approximation

- SimplEx
    - https://proceedings.neurips.cc/paper/2021/hash/65658fde58ab3c2b6e5132a39fae7cb9-Abstract.html
    - Adapt case-based reasoning to neural network
    - I.e. doctor diagnosis of new patients
    - Weights are computed by using the examples latent representations
        - Cutting network at representation layer
        - Mapping everything in latent space and see how it changes
    - Pros: no need to retrain model, much faster
    - Cons: requires access to the models latent representation
- Example-based explanations applications
    - Model's mistakes on training should be taken into account
    - If relevant training examples are misclassified, beware (e.g. from Crabbe et al. 2021 https://arxiv.org/abs/2203.01928)
    - Collecting training examples has a cost
    - Seller side: Compensate the data sellers appropriately
        - Jia et al., 2019
    - Buyer side: Quantitatively identify good data vendors

Concept-based explanations

- Nutshell:
    - Idea: investigate the black-box manipulates human concepts to make predictions
    - How: attribute an importance score a^c to each concept *c*
    - Example:
        - TCAV
        - TCAR (improvement of TCAV)
    - Limitations:
        - Need to provide many examples to illustrate a concept
        - Only works with neural networks
- TCAV
    - Idea: investigate how concepts are distributed in a model's representation space
    - https://mlconf.com/sessions/interpretability-beyond-feature-attribution-quant/
    - E.g. concept positive (stripe images) and concept negative (non stripe images) for a Zebra identification model
        - Cut at representation space and separate concepts at the hyperplane
        - Localize the concepts if its is represented in the space
    - Pros: user defined concepts and as long as examples are defined
    - Cons: assume the concepts sets are linearly separable in representation space
- TCAR
    - What if concept sets are not linearly separable?
    - Concepts are represented by region rather than vectors
- Applications
    - Scientific assessment of ML models requires to manipulate scientific concepts
    - Prostate cancer models recover the prostate grading system
        - If the grading system is encoded in the model – Yes

Future of interpretability

- Challenges:

- Interpretability does not protect against our own biases
    - How should we use these tools by avoiding e.g. confirmation bias?
- Most of the methods are designed in supervised setting
    - Early extensions to unsupervised setting (ICML Crabbe et al., 2022)
- Find new use-cases where interpretability helps
    - Examples form this talk just tip of the ice

- Not covered in the talk:
  - Counterfactual explanations
  - Rule-based explanations
  - Symbolic regression
  - Language explanations
- Links:
  - Code:https://github.com/vanderschaarlab/interpretability
  - Papers:https://www.vanderschaar-lab.com/interpretable-machine-learning/
  - Website: https://github.com/JonathanCrabbe

Q&A