Mining Unstructured Data

11. Interpretability

Javier Ferrando

Based on https://christophm.github.io/interpretable-ml-book/
Index

- Interpretability in Machine Learning
  - What is Interpretability
  - Interpretable models
  - Beyond validation metrics
- Interpretability in NLP
  - Saliency Maps
    - Gradients
    - Perturbations
- Interpretability in the Transformer
  - Analyzing Layers/Heads
  - Probing representations
- Ethics
- Ending Notes
Interpretability in Machine Learning
What is Interpretability?

- The degree to which a human can understand the cause of a decision
- The degree to which a human can consistently predict the model’s result
- The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made
- A model is better interpretable than another model if its decisions are easier for a human to comprehend than decisions from the other model
Interpretable models

- Linear Regression

\[
y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon
\]

\[
t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}
\]

|               | Weight | SE  | |t| |
|---------------|--------|-----|---|
| (Intercept)   | 2399.4 | 238.3 | 10.1 |
| seasonSPRING  | 899.3  | 122.3 | 7.4  |
| seasonSUMMER  | 138.2  | 161.7 | 0.9  |
| seasonFALL    | 425.6  | 110.8 | 3.8  |
| holidayHOLIDAY| -686.1 | 203.3 | 3.4  |
| workingdayWORKING DAY | 124.9  | 73.3  | 1.7  |
| weathersitMISTY| -379.4 | 87.6  | 4.3  |
| weathersitRAIN/SNOW/STORM | -1901.5 | 223.6 | 8.5  |
| temp          | 110.7  | 7.0  | 15.7 |
| hum           | -17.4  | 3.2  | 5.5  |
Interpretable models

- Linear Regression
- Logistic Regression

\[
\frac{P(y = 1)}{1 - P(y = 1)} = \text{odds} = \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)
\]

A change in a feature by one unit changes the odds ratio (multiplicative) by a factor of \(\exp(\beta_j)\)
Interpretable models

- Linear Regression
- Logistic Regression
- Decision Tree
Interpretable models

- Linear Regression
- Logistic Regression
- Decision Tree
- K-Nearest Neighbors
  - In classification, KNNs assigns the most common class of the nearest neighbors of an instance
  - In regression, it takes the average of the outcome of the neighbors
Why do we need Interpretability?

If a machine learning model performs well, why do we not just trust the model and ignore why it made a certain decision?

- “The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks.” (Doshi-Velez and Kim 2017)
- Knowing the ‘why’ of a prediction can help you learn more about the problem, the data and the reason why a model might fail.
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models

The [MASK] ran to the emergency room to see his patient.

Mask 1 Predictions:
- 36.5% doctor
- 12.7% man
- 2.8% boy
- 2.7% nurse
- 2.0% patient
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models

The [MASK] ran to the emergency room to see his patient.

Mask 1 Predictions:
- 36.5% doctor
- 12.7% man
- 2.8% boy
- 2.7% nurse
- 2.0% patient

The [MASK] ran to the emergency room to see her patient.

Mask 1 Predictions:
- 44.9% nurse
- 19.3% woman
- 7.4% doctor
- 5.3% girl
- 3.6% mother
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models

The [MASK] ran to the emergency room to see his patient.  

<table>
<thead>
<tr>
<th>Mask 1 Predictions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.5% <strong>doctor</strong></td>
</tr>
<tr>
<td>12.7% <strong>man</strong></td>
</tr>
<tr>
<td>2.8% <strong>boy</strong></td>
</tr>
<tr>
<td>2.7% <strong>nurse</strong></td>
</tr>
<tr>
<td>2.0% <strong>patient</strong></td>
</tr>
</tbody>
</table>

The [MASK] ran to the emergency room to see her patient.

<table>
<thead>
<tr>
<th>Mask 1 Predictions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.9% <strong>nurse</strong></td>
</tr>
<tr>
<td>19.3% <strong>woman</strong></td>
</tr>
<tr>
<td>7.4% <strong>doctor</strong></td>
</tr>
<tr>
<td>5.3% <strong>girl</strong></td>
</tr>
<tr>
<td>3.6% <strong>mother</strong></td>
</tr>
</tbody>
</table>

[CLS] The [MASK] ran to the **emergency** room to see **her** patient. [SEP]
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data

Test Example

Polar Bear ✗

Important Training Example

Polar Bear ✗
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them

**Article:** Super Bowl 50

**Paragraph:** "Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV."

**Question:** “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”

**Original Prediction:** John Elway

**Prediction under adversary:** Jeff Dean
Beyond validation metrics

- Find errors, bugs, and undesirable behavior in models
- Find errors and bugs in data
- Understand how models work so we can improve them
- Understand how models work so they are trusted (or not trusted)
Ok, but how?
Why did my model make this prediction?

Which parts of the input are responsible for this prediction?
Interpretability in NLP
Saliency Map Techniques in General

- Compute the relative *importance* of each token in the input
- Importance is, loosely: if you change or remove the token, how much is the prediction affected?

Examples of Saliency Maps:

**Sentiment**
- an *intelligent* fiction about learning through cultural *clash*.

**QA**
- What company won free *advertisement* due to QuickBooks *contest*?

**MLM**
- [CLS] The [MASK] ran to the *emergency* room to see *her* patient. [SEP]
● Saliency maps
  ○ generated using gradients
  ○ generated using perturbations
Gradients

- Given a function with 1 output and n inputs (scalar valued):

\[ f(x) = f(x_1, x_2, \ldots, x_n) \]

- Its gradient is a vector of partial derivatives with respect to each input:

\[ \frac{\partial f}{\partial x} = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right] \]
Given a (vector valued) function with m outputs and n inputs:

$$f(x) = [f_1(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)]$$

It’s Jacobian is an m x n matrix of partial derivatives:

$$\frac{\partial f}{\partial x} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}
\end{bmatrix}$$
Saliency Maps via Input Gradients

- Estimate importance of a feature using derivative of output w.r.t that feature

- i.e., with a “tiny change” to the feature, what happens to the prediction?

- We then visualize the importance values of each feature in a heatmap
Input Gradients in Computer Vision
Gradient-based Saliency Maps for NLP

For NLP, derivative of output w.r.t a feature = derivative of output w.r.t an input token

What to use as the output?
- Top prediction probability
- Top prediction logits
- Loss (with the top prediction as the ground-truth class)

What happens with multiple outputs?
- Text generation
- Tagging

Token is actually an embedding. How to turn gradient w.r.t embedding into a scalar score?
- Sum it?
- Take an L_p norm?
- Dot product with embedding itself?

Do we normalize values across sentence?

\[ \nabla e(t) \mathcal{L}_{\hat{y}} \cdot e(t) \]
Input Gradients in Computer Vision
Problems with Using Gradients for Saliency Maps

- Too “local” and thus sensitive to slight perturbations
Problems with Using Gradients for Saliency Maps

- Too “local” and thus sensitive to slight perturbations
- “saturated outputs” lead to unintuitive gradients

\[ y = x_1 + x_2 \quad \text{when} \quad (x_1 + x_2) < 1 \]
\[ 1 \quad \text{when} \quad (x_1 + x_2) \geq 1 \]

If \( x_1 \) or \( x_2 \) goes to 0, gradient is 0. Neither is important?
Problems with Using Gradients for Saliency Maps

- Too “local” and thus sensitive to slight perturbations
- “saturated outputs” lead to unintuitive gradients
- discontinuous gradients (e.g., thresholding) are problematic

\[ x = (10-\varepsilon, 10+\varepsilon) \]
Problems with Using Gradients for Saliency Maps

- Too “local” and thus sensitive to slight perturbations
- “saturated outputs” lead to unintuitive gradients
- Discontinuous gradients (e.g., thresholding) are problematic

How to mitigate these issues? Don’t rely on a single gradient calculation:
- SmoothGrad
- Integrated Gradients
Extensions of Vanilla Gradient

**SmoothGrad**: add gaussian noise to input and average the gradient
Extensions of Vanilla Gradient

Integrated Gradients: average gradients along path from zero to input

\[ p(y|x) \]
Summary of Gradient-based Saliency Methods

Positives:
- Fast to compute: single (or a few) calls to \texttt{backward()}
- Visually appealing: spectrum of importance values

Negatives:
- Needs white-box (gradient) access to the model
- Not “customizable”
  - small changes in an individual “token” are not necessarily meaningful
  - distance is implicitly Euclidean ($L_2$)
- Gradients can be unintuitive with saturated or thresholded values
- Difficult to apply to non-classification tasks
● Saliency maps
  ○ generated using gradients
  ○ generated using perturbations
Alternative: Generating Saliency Maps Using Input Perturbations

Goal is the same: saliency map over the input

However, these methods:

- are black-box (completely model-agnostic)

- allow input perturbations/neighborhoods to be defined
  - i.e., we can use different “units of explanations”
    - words, phrases, sentences
    - for multimodal inputs: image, environment, etc.
## Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

<table>
<thead>
<tr>
<th>Question</th>
<th>Confidence</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.78</td>
<td></td>
</tr>
</tbody>
</table>
Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed.

<table>
<thead>
<tr>
<th>Question</th>
<th>Confidence</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td><strong>What</strong> did Tesla spend Astor’s money on?</td>
<td>0.67</td>
<td><strong>What</strong></td>
</tr>
</tbody>
</table>
Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

<table>
<thead>
<tr>
<th>Question</th>
<th>Confidence</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.67</td>
<td>What</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.72</td>
<td>did</td>
</tr>
</tbody>
</table>
Leave-one-out: remove tokens and look at confidence

- Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

<table>
<thead>
<tr>
<th>Question</th>
<th>Confidence</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.67</td>
<td>What</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.72</td>
<td>did</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.66</td>
<td>Tesla</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.74</td>
<td>spend</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.76</td>
<td>Astor’s</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.48</td>
<td>money</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.72</td>
<td>on</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on?</td>
<td>0.73</td>
<td>?</td>
</tr>
</tbody>
</table>
Simplest method is leave-one-out: define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed.

<table>
<thead>
<tr>
<th>Question</th>
<th>Confidence</th>
<th>Highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.67</td>
<td>What</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.72</td>
<td>did</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.66</td>
<td>Tesla</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.74</td>
<td>spend</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.76</td>
<td>Astor’s</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.48</td>
<td>money</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.72</td>
<td>on</td>
</tr>
<tr>
<td>What did Tesla spend Astor’s money on ?</td>
<td>0.73</td>
<td>?</td>
</tr>
</tbody>
</table>

What did Tesla spend Astor’s **money** on ?
Concern with Leave-one-out

Problem: Leave-ONE-out

The movie is mediocre, maybe even bad.   Negative 99.8%

The movie is mediocre, maybe even bad.   Negative 98.0%

The movie is mediocre, maybe even bad.   Negative 98.7%

What we really need:

The movie is mediocre, maybe even bad.   Positive 63.4%
LIME: Intuition Behind the Approach

- Look at model’s predictions for a bunch of nearby inputs
- Closer points are more important than further points
- Fit a linear model. Its weights are the feature importances
The movie is mediocre, maybe even bad.

Negative 99.8%
Negative 98.0%
Negative 98.7%
Positive 63.4%
Positive 74.5%
Negative 97.9%

The movie is mediocre, maybe even bad.
Problems with Leave-one-out and LIME

- Customizing perturbations and distances is difficult to define intuitively
- More importantly, difficult for users to understand
  - different explanations with different perturbations?
- How do we define “distance” between sentences? Lexical similarity?
- Generating interpretations is expensive (many calls to underlying model)
- Difficult to apply to non-classification tasks
Interpretability in the Transformer
What do attention heads learn?
What do self-attention heads learn?
What do self-attention heads learn?

**Head 7-6**
- **Possessive pronouns** and apostrophes attend to the head of the corresponding NP
- 80.5% accuracy at the poss relation

**Head 4-10**
- **Passive auxiliary verbs** attend to the verb they modify
- 82.5% accuracy at the auxpass relation
What do self-attention heads learn?

**Head 9-6**

- **Prepositions** attend to their objects
- 76.3% accuracy at the pobj relation

**Head 5-4**

- **Coreferent** mentions attend to their antecedents
- 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent
What do cros-attention heads learn?

Source-target word alignment
Understanding representations by probing

Hypothesis:

- Neural models, especially large ones like BERT, perform well without any explicit linguistic supervision in part because they learn similar notions themselves.

Question:

- Do neural networks’ internal representations encode linguistic notions of structure, like parts-of-speech, dependency trees, named entities?
Probing: supervised analysis of representations

- Does my network make task (e.g. part-of-speech) labels accessible?

Choose a function family to decode the task

Train a function representation task

Interpret accuracy on test-set
Ethics
Robustness

Robustness: Sentiment Classification fails just with typos

<table>
<thead>
<tr>
<th>Connoisseurs of Chinese film will be pleased to discover that Tian's meticulous talent has not withered during his enforced hiatus.</th>
<th>Prediction: <strong>Positive (77%)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aonnoisseurs of Chinese film will be pleased to discover that Tian's meticulous talent has not withered during his enforced hiatus.</td>
<td>Prediction: <strong>Negative (52%)</strong></td>
</tr>
</tbody>
</table>
Robustness

Major flaws in facial recognition systems revealed: Bizarre 'face stealing' specs can fool them into thinking you are someone else (and can even turn a man into Milla Jovovich)

- Glasses allow wearer to dodge recognition or impersonate another person
- Method disrupts the system's ability to accurately read pixel colouration
- In experiments, it allowed a man to impersonate actress Milla Jovovich
- Researchers say it highlights the ways attackers might evade technology
COMPAS is an assistive (biased) software and support tool used to predict recidivism risk

The prediction fails differently for the black defendants:
Racial disparities in automated speech recognition

© Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartery, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel

PNAS April 7, 2020 117 (14) 7684-7689; first published March 23, 2020; https://doi.org/10.1073/pnas.1915768117

Edited by Judith T. Irvine, University of Michigan, Ann Arbor, MI, and approved February 12, 2020 (received for review June 22, 2019)
Gender Shades showed face recognition is much less accurate on black people
AI TayTweets learnt from conversations held on social media and it turned to be racist.

Taylor Swift 'tried to sue' Microsoft over racist chatbot Tay

© 10 September 2019
Github's Copilot emitting keys
Gender Bias in Translations

Malay - detected
Dia bekerja sebagai jururawat.
Dia bekerja sebagai pengaturcara.

English
She works as a nurse.
He works as a programmer.

Didn’t specify gender
Environmental Costs

Common carbon footprint benchmarks

in lbs of CO2 equivalent

- Roundtrip flight b/w NY and SF (1 passenger) 1,984
- Human life (avg. 1 year) 11,023
- American life (avg. 1 year) 36,156
- US car including fuel (avg. 1 lifetime) 126,000
- Transformer (213M parameters) w/ neural architecture search 626,155

Chart: MIT Technoloov Review • Source: Strubell et al. • Created with Datawrapper
Ending Notes
Classical NLP vs Deep Learning NLP
What tools are there for doing NLP stuff?

https://huggingface.co/docs/transformers/index

```python
>>> pipe = pipeline("text-classification")
>>> pipe("This restaurant is awesome")
[{'label': 'POSITIVE', 'score': 0.9998743534088135}]
```
Want to learn more?


https://web.stanford.edu/~jurafsky/slp3/

https://jalammar.github.io/
Thank you for your attention

Javier Ferrando
Universitat Politècnica de Catalunya
javier.ferrando.monsonis@upc.edu