Explainable AI: Current state and some thoughts for the Future

Debasis Ganguly [1]

[1] University of Glasgow, Glasgow, UK

May 17, 2023

Overview

1. Feature to Data driven
2. Trustability of AI systems
3. Types of Explanations
4. How does an Explanation Mechanism work
5. LIME: A Case Study
6. Explanations: A Step Towards (Real) Thinking?
7. Local Explanations for Ranking
8. Explaining ranking functions
9. Societal Impacts

Explanations in AI

Morty: Which movie should I watch?

Don't answer like Rick!

Stacked layers of feed-forward layers over convolutions of word embeddings.

How does that help me to know if a movie is good or bad from the reviews.

Feature to Data

Evolution of AI models from feature-driven to data-driven

- Feature-driven models:
  - Relied on human perceived abstract representations of the data.
  - Hence more clarity about what happens and how.
  - Easy to include/exclude features based on intuition.
- Data-driven models:
  - Relies on machine-generated abstractions, e.g. 1D convolution for text, 2D convolution for images etc.
  - How do we control the predictions?
  - How do we convince others, e.g. think about convincing a medical person about an automated diagnosis.

Trust Issues

1. If users do not trust a model or a prediction, they will not use it.
2. Two notions of trust:
    1. Trusting a **prediction**, i.e. whether a user trusts an individual prediction sufficiently to take some action based on it.
    2. Trusting a **model**, i.e. whether the user trusts a model to behave in reasonable ways if deployed.
3. Both are directly impacted by how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Levels of Trust

Prediction-level

- Explain the predictions of any classifier (or a regression model) by approximating it locally with an *interpretable* model.
- The explanations are usually in the form of *importance weights* assigned to different features.

Model-level

- Obtain per-instance *explanations* or feature weights and then choose a set of representative instances.

Different explanation units for different application domains

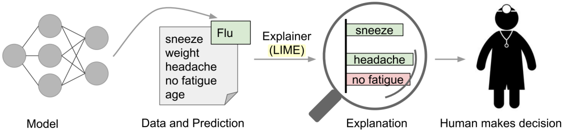
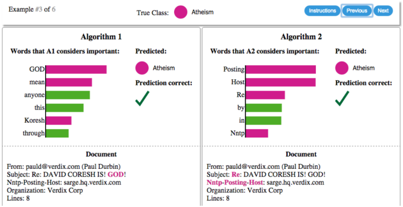


Figure 1: Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights which symptoms in the patient's history led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about the model's prediction.



Example #3 of 6

True Class: ● Atheism

Algorithm 1

Words that A1 considers important:

Predicted: ● Atheism

Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Algorithm 2

Words that A2 considers important:

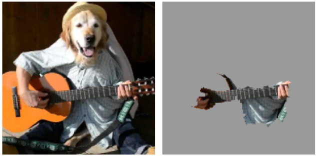
Predicted: ● Atheism

Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

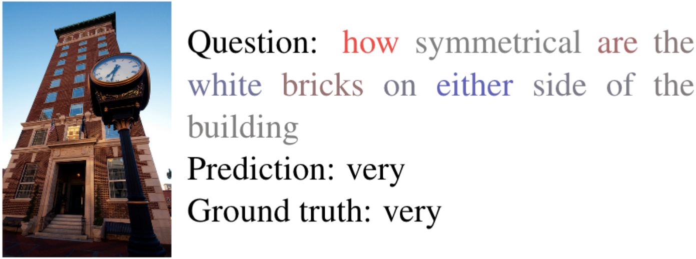
Figure 2: Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text.



(a) Original Image (b) Explaining *Electric guitar*

Images taken from (Ribeiro et. al., KDD'16)

Different explanation units for different application domains



Question: how symmetrical are the white bricks on either side of the building

Prediction: very

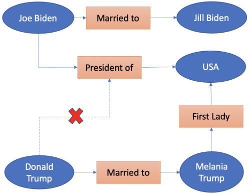
Ground truth: very

Red: high, Blue: low, Gray: close to 0.

(Visual) Question Answering

- Does the process of arriving at the answer correlate with human perception (Mudrakarta et. al., ACL'18)?
- Does the model put emphasis correctly on the correct set of words to generate the answer?

Different explanation units for different application domains



Who is the first lady of USA?

Knowledge base Completion (KBC)

- A triple of the form (h, r, t), h: a head entity (subject), r: a relation, and t : a tail entity (object).
- Useful for answering factoid type questions.
- An explanation in KBC could be generating inference rules, i.e. $(e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \implies (e_1, r_3, e_3)$, e.g. '*Jill* is the *first lady* **because** Jill is the *spouse of* Joe and Joe is the *president*'.

Counterfactual Explanations

What if things were different?

- What would the model have predicted if the input is slightly different?
- Useful when the users are prepared to *understand* the system predictions analyzing the predictions on alternatives for certain choices.
  - Administrative policy making - Will unemployment rate decrease if college education costs are decreased?
  - Movie recommendation - M recommends ‘*Scarface*’ because I watched ‘*Godfather*’ and ‘*Irish man*’; if I hadn’t watched *Irish man*, M would recommend me ‘*Godfather 2*’.

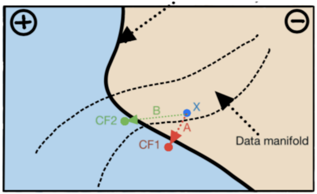


Figure 1: Two possible paths for a datapoint (shown in blue), originally classified in the negative class, to cross the decision boundary. The end points of both the paths (shown in red and green) are valid counterfactuals for the original point. Note that the red path is the shortest, whereas the green path adheres closely to the manifold of the training data, but is longer.

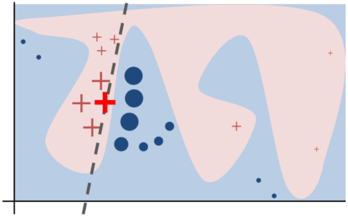
Desirable Characteristics

1. **Interpretable**: Explanation units should have a correlation with human-perceived features.
   - Text: Words or phrases within a document.
   - Regions from images.
2. **Local Fidelity**: Must be *locally faithful* - i.e. it must correspond to how the model behaves in the vicinity of the instance being predicted.
3. **Model-Agnostic**: The explainer algorithm should not make any assumptions about the underlying working principle of the model.
   - Else one would need a different explainer for each new model!

Transforming Data to Explanations

- *Transform* each input instance $\vec{x} \in \mathbb{R}^d$ to a different space representing its feature importance or explanation vector.
- Given: a particular data instance $\vec{x}$ and a parameterized (trained) model $\theta : \vec{x} \mapsto y$.
- Samples other similar data instances, $\vec{z} \in N(\vec{x})$, from its neighborhood.
- Learns a *local view* of the global model by leveraging the predictions of θ on these *local* instances.

Local Approximation to estimate feature weights



- Objective: Estimate the (soft attention) weights of each feature of this instance (the red colored cross).
- Sample other points from around this point.
- Fit a simple (linear) classifier on this subset.
- This simple classifier *approximates* the behaviour of the complex decision boundary locally.
- Explain the current point with the parameters of this linear classifier.
- $\vec{x} \in \mathbb{R}^d \mapsto \phi(x) \in \mathbb{R}^p$.

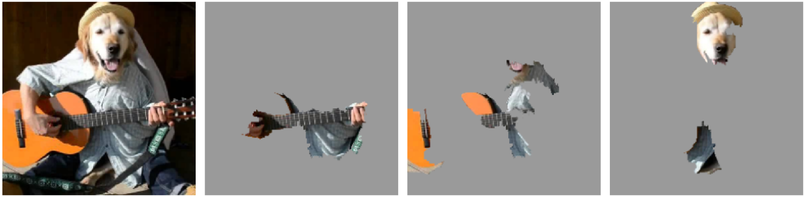
Objective Function

1. The instance-wise explanation objective is to make the local model $\theta_{\vec{x}}$ *closely approximate* the global model θ.
2. Minimize a loss of the form

$$\mathcal{L}(\vec{x}, \theta; \phi) = \sum_{\vec{z} \in N(\vec{x})} (\theta(\vec{z}) - \phi \cdot \vec{z})^2, \ N(\vec{x}) = \{\vec{x} \odot \vec{u} : \vec{u} \sim \{0, 1\}^d\}, \quad (1)$$

3. $\phi \cdot \vec{z}$ is a parameterized linear representation of the local function $\theta_{\vec{x}}$.
4. Neighborhood function is approximated by selecting arbitrary subsets of features of the current instance $\vec{x}$
5. These are of the form $\vec{x} \odot \vec{u}$, where $\vec{u}$ is a random bit vector of size d.

Different explanations for the same instance



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$)**, "Acoustic guitar"** ($p = 0.24$) **and "Labrador"** ($p = 0.21$)

- Recall that the objective depends on the predicted class label, $\theta(\vec{x})$.
- Hence the explanation weights can be different for different predicted labels.

From Per-instance Explanations to Model Explanation

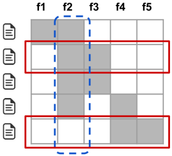


Figure 5: Toy example $\mathcal{W}$. Rows represent instances (documents) and columns represent features (words). Feature f2 (dotted blue) has the highest importance. Rows 2 and 5 (in red) would be selected by the pick procedure, covering all but feature f1.

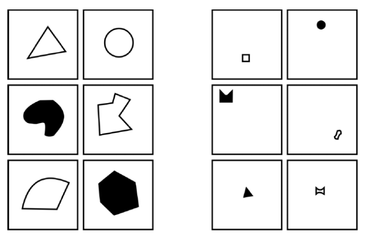
- What are the most important features (overall) for a model?
- Transform each $\vec{x} \mapsto \phi(x)$, i.e. estimate soft-attention weights for each.
- Pick those instances which cover maximally the feature space.

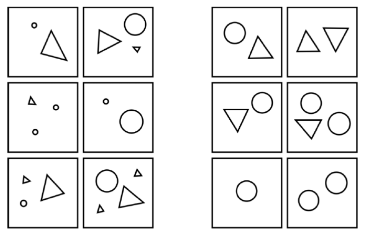
Bongard Problems

- Invented by the Russian computer scientist Mikhail Moiseevich Bongard.
- Popularized by Douglas Hofstadter in his Pulitzer prize winner - *Godel, Escher and Bach*.

Task

- *Explain* (in language), *why* the images on the left different from those in the right.
- Tests the abstract thinking capacity.



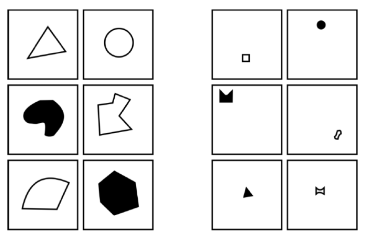


Bongard Problems

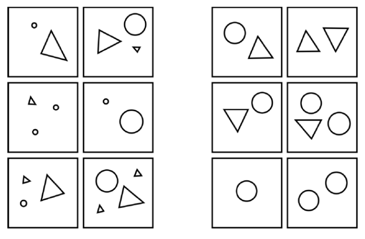
- Invented by the Russian computer scientist Mikhail Moiseevich Bongard.
- Popularized by Douglas Hofstadter in his Pulitzer prize winner - *Godel, Escher and Bach*.

Task

- *Explain* (in language), *why* the images on the left different from those in the right.
- Tests the abstract thinking capacity.



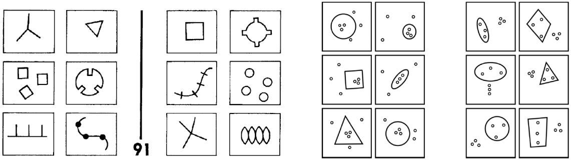
Large figures vs. Small figures



Small figure present vs. No small figure present

Characteristics of (Human) Intelligence

- Different levels of abstraction.
  - What combinations of attributes to use to define an object.
  - Some are more fine-grained (e.g., number of corners, lines etc.) than others (e.g., convexity).
- Moving back and forth between these representations to define how are objects similar and how are they dissimilar, specific to a task.

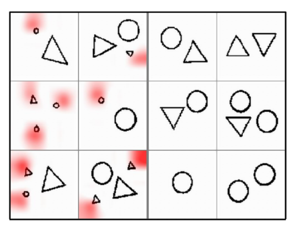


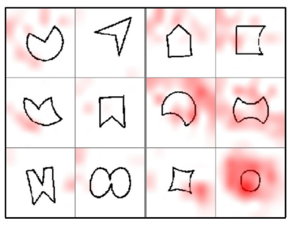
  - **Left:** BP denotes an *abstract property* for the understanding of numbers 3 and 4. More fine-grained concepts of corners, lines, wedges don't work. **Right:** An abstract concept of density is required.

The Road Ahead

Explanation technology today

A heatmap of attention weights overlaid on objects (images in this example).





Future

Clear descriptions of the inference procedure, expressed as human perceivable *concepts*.

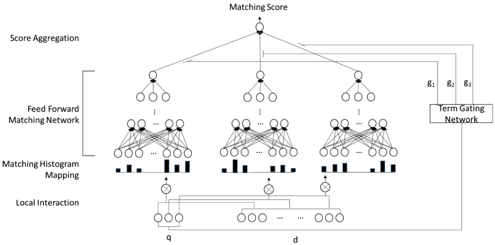
Small figure present vs. No small figure

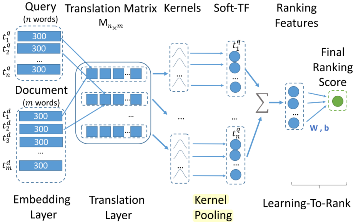
Angle directed inwards vs. No inward angles

Feature-driven to Data-driven

- Information Retrieval: Given a query Q, objective is retrieve a ranked list of documents that are relevant to Q.
- Statistical models employ term weightings such as tf-idf to compute the scores $s(D, Q)$.
- Deep learning to rank models employ characterizing the patterns between the matching between query and document terms.
  - Pairwise training: Given a query Q and a document pair D_1 and D_2, predict a binary response 1 if D_1 is relevant and D_2 is not, or 0 otherwise.
  - Point-wise prediction (during testing): Given a query and a document D predict its score.

Two Example Deep Learning-to-Rank Models





- DRMM (left) uses histograms of word pair similarities (between doc and query) terms as inputs to a feed-forward network.
- The model seeks to utilize inherent patterns in these histograms to distinguish relevance from non-relevance.
- KNRM (right) does not need to rely on histograms. Instead it applies 1D convolution.

Explanations for IR Models

- Some models may not be easy to 'explain' to a search engine user, who may have questions such as 'Why does a search engine retrieve document D at rank k?'
- Motivation to capture local effects on subsamples and predict a distribution of term importance potentially capturing an IR model's inherent term weighting characteristics.
- Differences from classification task explainers (e.g. LIME):
  - IR models involve a set of interacting pairs (query and document) rather than one single instance.
  - It makes sense to take sub-samples with respect to a document D, where a sub-sample is a pseudo-document formed by randomly sampling terms from D.

Objective Function

- A ranking function can be represented as

$$S(D,Q) = \sum_{t \in D \cap Q} w(t,D)$$

  - $w(t,D)$: term weight of term t in a document D.

$$\begin{aligned}\mathcal{L}(D,Q,\sigma;\vec{\Theta}) &= \sum_{i=1}^{M} \rho(D,D'_i)\left(S(D,Q) - S_{\vec{\Theta}}(D'_i,Q)\right)^2 + \alpha|\vec{\Theta}| \\ &= \sum_{i=1}^{M} \rho(D,D'_i)\left(S(D,Q) - \sum_{j=1}^{p} \theta_j w(t_j,D'_i)\right)^2 + \alpha|\vec{\Theta}|.\end{aligned}$$

Explaining the Objective Function

- $D'_i = \sigma_i(D)$: i^{th} sample extracted from a document D comprised of p unique terms.
- α: L1 regularization term.
- $\vec{\Theta} \in \mathbb{R}^p$: a vector of p real-valued parameters used to approximate the score of the sub-sample D'_i with respect to the query Q.
- Weight of the loss $\rho(D, D'_i)$ is a similarity between the document D and its sub-sample D'_i.
- ρ: Kernel function of the form

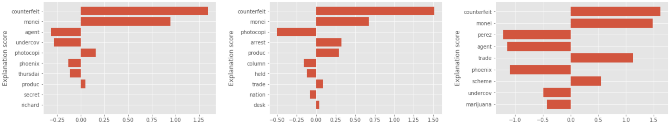
$$\rho(D, D') = \exp(-\frac{x^2}{h}),\ x = \arccos(D, D')$$

  - $\arccos(D, D')$: cosine-distance (angle) between a document D and a sub-document sampled from it
  - h: width of a Gaussian kernel.

- The weighted loss function predicts $S(D, Q)$ using the given samples.
- Since a retrieval model computes the score of an entire document and also the scores of its sub-samples, the predicted vector $\vec{\hat{\Theta}} \in \mathbb{R}^p$ estimates the importance of each term.
- E.g. the j^{th} component of $\vec{\hat{\Theta}}$ denotes the likelihood of term t_j in contributing positively to the overall score $S(D, Q)$.
- Weights in $\vec{\hat{\Theta}}$ that correspond to a query term should have larger weights (denoting higher importance).
- Non-query terms with high weights in $\vec{\hat{\Theta}}$ are potentially the ones that are semantically related to the query and hence are likely to be relevant to its underlying information need.
- A visualization of these terms may then provide the desired explanation of an observed score of a document D with respect to Q (high or low).

Different Sampling Strategies

- **Uniform Sampling:**
  - Sample terms with a uniform likelihood (with replacement).
  - No bias towards term selection leading to likely generation of a diverse set of samples for a document.
- **Biased Sampling:**
  - Set the sampling probability of a term proportional to its tf-idf weight seeking
  - Generate sub-samples with informative terms.
- **Masked Sampling:**
- Extract segments of text from a document, somewhat analogous to selecting regions from an image.
- Specify a segment size, say k, and then segment a document D (comprised of $|D|$ tokens) into $\frac{|D|}{k}$ number of chunks.
- A chunk is then made visible in the sub-sample with probability v (a parameter).



(a) Uniform sampling **(b) Tf-idf sampling** **(c) Masked samples ($v = 0.1$, $k = 5$)**

Figure 4: Visualization of explanation vectors $\hat{\Theta}(Q, D)$ estimated for a sample (relevant) document 'LA071389-0111' (D) and query (Q) 'counterfeiting money' (TREC-8 id 425). The Y-axis shows explanation terms, while the X-axis plots their weights.

- Positive weights obtained for query terms (e.g. 'counterfeit').
- For non-query terms, the explanation weights vary across sampling methods
  - Choice of sampling method can considerably impact the quality of the local explanations generated by any LIRME.
- Terms (output as negative weights): not relevant to the information need of the example query, e.g. 'phoenix', 'agent' etc.
- Positive weights are likely to help a user discover the associated relevant sub-topics within it.
- Negative weights indicate potential non-relevant aspects.

Fundamental building blocks of a ranking model

1. *term frequency* of a term in a document,
2. *document frequency* or collection frequency of a term (an estimate of its global informativeness measure),
3. *length* of a document.

Aggregation over per-term similarity scores

$$s(Q,D) = \sum_{q \in Q \cap D} s(q,D),\ s(q,D) = \Phi(\phi_x(q,D), \phi_y(D), \phi_z(q)),$$

Fundamental building blocks of a ranking model

Function units

1. $\phi_x(q, D) : \mathbb{Z} \mapsto \mathbb{R}$: transforms an integer raw frequency of a term q in a document D, $x = f(q, D) \in \mathbb{Z}$, to a real number, e.g. $\phi_x(x) = \{\sqrt{x}, \log(x) \ldots\}$.
2. $\phi_y(D) : \mathbb{Z} \mapsto \mathbb{R}$: transforms the integer value of the length of a document $y = |D|$, to a real number, e.g. $\phi_y(y) = y^{-1}$ etc.
3. $\phi_z(q) : \mathbb{Z} \mapsto \mathbb{R}$: transforms the number of documents across the whole collection in which the term t occurs , $z = c(q)$ (an integer), to a real, e.g. $\phi_z(z) = z^{-1}$ etc.

Different functional units for different ranking models

BM25

$$s(q, D) = \frac{N}{log(c(q))} \frac{f(q, D)(k_1 + 1)}{f(q, D) + k_1(1 - b + b\frac{|D|}{|\hat{D}|})}$$

- N: number of documents
- $|\hat{D}|$: average document length in the collection.
- Parameters k_1 and b: trade-off between term frequency and document lengths.

Functional Units for BM25

- $\phi_x(x) = x$
- $\phi_y(y) \propto y^{-1}$
- $\phi_z(z) \propto z^{-1}$

Lack of Transparency in Neural Models

- Neural models such as DRMM employ pairwise learning to rank.
- A triplet loss to predict a higher score for a relevant document D^+ with respect to a non-relevant one (D^-).

$$\mathcal{L}(q, D^+, D^-; \Theta) = \max(0, 1 - s(q, D^+) + s(q, D^-))$$

- Binned histograms of term-match score distributions (weighted by inverse document frequencies).
- Difficult to see what effects does term frequency, ϕ_x, and document length functions, ϕ_y play in the overall predicted score of DRMM.

Explaining the ranks

Why does a model M rank a document D at position $r > 1$ (say 5) instead of retrieving it at rank 1

1. M is a statistical model
   1. compute the values of term frequencies and collection statistics of matched terms in D
   2. compare these values with the top-ranked document retrieved D_{top}.

Example: M is BM25 with b set to a high value

1. $|D| < |D_{top}| \implies$ why would the rank of D be 5.
2. It is not possible to conduct this analysis for neural models, such as DRMM.

Linear Regression

- Similarity score represented as a 3-dimensional vector of coefficients of the components, $\phi = (\phi_x, \phi_y, \phi_z)$.
- $\phi_x(w, D) \stackrel{\text{def}}{=} f(w, D)$
- $\phi_y(D) \stackrel{\text{def}}{=} |D|$
- $\phi_z(w) \stackrel{\text{def}}{=} c(w)$
- For a neural model an additional dimension - $\phi_\omega(w, t, D) = \vec{w} \cdot \vec{t}$: similarity between the embedded vector representations of terms w and t in a document D.

Fit a regressor on the top-doc scores

Parameter vector $\theta = (\theta_0, \theta_x, \theta_y, \theta_z) \in \mathbb{R}^4$: learned by minimizing the L_2-regularized square loss function with gradient descent.

$$\mathcal{L}(\vec{\theta}) = (s(w, D) - (\theta_x x + \theta_y y + \theta_z z + \theta_0))^2 + |\vec{\theta}|_2 \tag{2}$$

Societal Impacts

Is a model fair?

- Can a model be *made to understand* cognitive biases (and hence get rid of those), e.g.,?
  - men are better programmers than women (Bolukbasi et. al., NIPS'17);

Explanations can help

- Visualize and perceive the way the models construct abstractions from the data.
- Potentially control these abstractions to mitigate biases.

Summary and Conclusions

- Explanations in AI:
  - A medium to build trustworthiness with end-users.
  - A tool for practitioners develop more effective models.

- Current state of AI explanation:
  - Still limited to attention weights over fine-grained units (pixels or words).
  - Lacks true test of *comprehension*.
- Road to the Future:
  - Come up with more abstract and easy to understand explanations (e.g. natural language).
  - Apply under interactive decision-making environments, e.g. reinforcement learning.
  - Apply for unsupervised/semi-supervised learning settings, e.g. Bongard problem type pattern matching.