GT Explicabilité

Christophe Denis (EDF R&D, SU), Nicolas Maudet (LIP6, SU)

Journée commune MAFTEC - Explicabilité — GREYC, Caen
Motivation

- new regulations (eg. GDPR)
- raising concern in the society: making A.I. systems trustable!

Featured in mainstream press, related to prominent applications:

- automated decisions for autonomous vehicles
- loan agreements
- Admission Post Bac
Motivation

- new regulations (eg. GDPR)
- raising concern in the society: making A.I. systems trustable!

Featured in mainstream press, related to prominent applications:

- automated decisions for autonomous vehicles
- loan agreements
- Admission Post Bac
Motivation

- new regulations (eg. GDPR)
- raising concern in the society: making A.I. systems trustable!

Featured in mainstream press, related to prominent applications:

- automated decisions for autonomous vehicles
- loan agreements
- Admission Post Bac
Research trends

• Expert systems (eg. MYCIN)!
• DARPA XAI (Explainable A.I.) initiative
• IJCAI-2018 Federation of 4 workshops:
  • Explainable Artificial Intelligence (XAI)
  • Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)
  • Human Interpretability in Machine Learning (WHI)
  • Interpretable & Reasonable Deep Learning and its Applications (IReDLia)
• + ICAPS Explainable AI Planning / NIPS Interpretable ML / ...

...
Based on some interactions with a user (e.g. history of previous choices, attributes of the user, preference statements...), our A.I. system has to recommend a hotel in Paris.

Our recommendation algorithm is based on a cutting-edge weighted sum technique which combines your preferences about location and breakfast!
Based on some interactions with a user (e.g. history of previous choices, attributes of the user, preference statements...), our A.I. system has to recommend a hotel in Paris.

We recommend the yellow hotel because you’re a young researcher.
Based on some interactions with a user (e.g., history of previous choices, attributes of the user, preference statements...), our A.I. system has to recommend a hotel in Paris.

We recommend the yellow hotel because last time you came to Paris you went to a close-by cinema twice and you visited your good friend Joe who lives in the neighbourhood.
Based on some interactions with a user (e.g. history of previous choices, attributes of the user, preference statements...), our A.I. system has to recommend a hotel in Paris.

We recommend the yellow hotel because you liked the blue hotel and people who like the blue hotel also like the yellow hotel.
Based on some interactions with a user (e.g. history of previous choices, attributes of the user, preference statements...), our A.I. system has to recommend a hotel in Paris.

- We recommend the yellow hotel because you only stay one night. If you had stayed at least 3 nights we would have recommended the green hotel instead because they offer interesting discount.
Our recommendation algorithm is based on a cutting-edge weighted sum technique which combines your preferences about location and breakfast!

We recommend the yellow hotel...

... because you’re a young researcher.

... because last time you came to Paris you went to a close-by cinema twice and you visited your good friend Joe who lives in the neighbourhood.

... because you liked the blue hotel and people who like the blue hotel also like the yellow hotel.

... because you only stay one night. If you had stayed at least 3 nights we would have recommended the green hotel because they offer interesting discount.
The legal debate
A **right to explanation** has been put forward by some legislative texts, in particular the recent General Data Protection Regulation (GDPR). According to Goodman and Flaxman:

“In its current form, the GDPR’s requirements could require a complete overhaul of standard and widely used algorithmic techniques.”

Goodman and Flaxman. *EU regulations on algorithmic decision-making and a ‘right to explanation’*. ArXiv-2016.
However, in their examination of the legal status of the GDPR, Wachter et al. conclude that such a right does not exist yet. The right to explanation is only explicitly stated in a recital: 

*a person who has been subject to automated decision-making “should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision”*

However, recitals are not legally binding. It also appears to have been intentionally not included in the final text of the GDPR after appearing in an earlier draft.
Still, Article 13 and 14 about notification duties may provide a right to be informed about the “logic involved” prior to decision “existence of automated decision-making, including profiling [...] [and provide data subjects with] meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing.”

As it stands, only provides a (limited : secret of affairs, etc.) right to obtain ex-ante explanations about the model (which they call, ‘right to be informed’).

Loi pour une république numérique

L’administration communique à la personne faisant l’objet d’une décision individuelle prise sur le fondement d’un traitement algorithmique, à la demande de celle-ci, sous une forme intelligible et sous réserve de ne pas porter atteinte à des secrets protégés par la loi, les informations suivantes :

- Le degré et le mode de contribution du traitement algorithmique à la prise de décision ;
- Les données traitées et leurs sources ;
- Les paramètres de traitement et, le cas échéant, leur pondération, appliqués à la situation de l’intéressé ;
- Les opérations effectuées par le traitement.

Décret du 14 Mars 2017, cité et commenté dans :
Clarifying the notions
Transparency does not imply explainability

```python
(lambda __, __, __, __, __, __, __, __):
    setattr(
        __import__(True, __class__.__name__, [], [__class__.__name__, ])[0],
        __class__.__name__, lambda: lambda(__, __, __)
    )

    lambda __, __, __:
        chr(__ % __) + (__, __, __)  # if __ else
    (lambda __).func_code.co_inlinenumber,

    __<__
    (((((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))
    (((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))
    __<__
    (((((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))
    __<__
    (((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))
    __<__
    (((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))
    __<__
    (((((((__ << __) + __) << (__ << __)) + (__ << __)) + (__ << __)) + (__ << __)) + (__ << __))

*(lambda __, __, __: lambda(__, __, __))

(lambda __, __, __):
    ((__[[[lambda: __].func_code.co_nlocals]]) +
     __[[[lambda: __].func_code.co_nlocals]]) if __ else []

(lambda __: __.func_code.co_argcount,

    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
    lambda __: __,
transparency does not imply explainability

 prints Hello World! (by Ben Kurtovic, winner of a 2017 obfuscation contest)
Budish et al. claim that an explanation should allow to answer the following questions:

1. What were the main factors in a decision?
2. Would changing a given factor have changed the decision?
3. Why did two similar-looking cases get different conclusions, or vice-versa?

• Why did you do that?
  
  
  
  - issues of causality + understandable by humans

• Why didn’t you do something else (that I would have done)
  
  
  
  - demonstrating that the alternative action would prevent from finding a valid plan or would lead to a plan that is no better than the one found by the planner

• Why is what you propose to do more efficient/safe/cheap than something else (that I would have done)?
  
  
  
  - interesting case is when one wants to evaluate a plan using a metric which is different from the one used when searching
...in Explainable Planning?

- Why can’t you do that?
  - when a planner fails to find a plan for a problem
- Why do I need to replan at this point?
  - discovering what has diverged from expectation
- Why do I not need to replan at this point?
  - the observer has seen a divergence in expected behaviour and does not understand why it should not cause plan failure

Some reasons why we may question explainability

Devil’s advocate:

1. requiring explainable decisions may affect the efficiency of the system
Some reasons why we may question explainability

Devil’s advocate:

1. requiring explainable decisions may affect the efficiency of the system
2. providing an explanation may be costly
Some reasons why we may question explainability

Devil’s advocate:

1. requiring explainable decisions may affect the efficiency of the system
2. providing an explanation may be costly
3. if the explanation is too detailed, users may manipulate the system
Some reasons why we may question explainability

Devil’s advocate:

1. requiring explainable decisions may affect the efficiency of the system
2. providing an explanation may be costly
3. if the explanation is too detailed, users may manipulate the system
4. explanation may be used as a way to avoid “real” transparency
The explanation landscape is rich already

Option #1: Add explanation engines on top of existing systems:

- **model-agnostic** explanations, eg:
  - data-based explanations (incl. counterfactuals)
  - locally faithful approximations, surrogate models
- **model-specific** explanations, eg:
  - minimized traces (causality)
  - argumentation/explanation schemes

Option #2: Build systems explainable by design:

- add constraints or objective (capturing interpretability)
- restrict operators to argumentation schemes validated by the user.
Activités

- 04/12/17 : Journée thématique de lancement (Paris)
- 01/10/18 : Réunion du groupe de travail (Paris)
- 08/10/18 : Journée Machine Learning and Interpretability (Orléans)
- 01/04/19 : Journée Commune MAFTEC - Explicabilité (Caen)
- 27-28/05/19 : Journée de travail + Explicabilité en diagnostic médical

Outils

Site web du GT : https://gt-explication.gitlab.io/