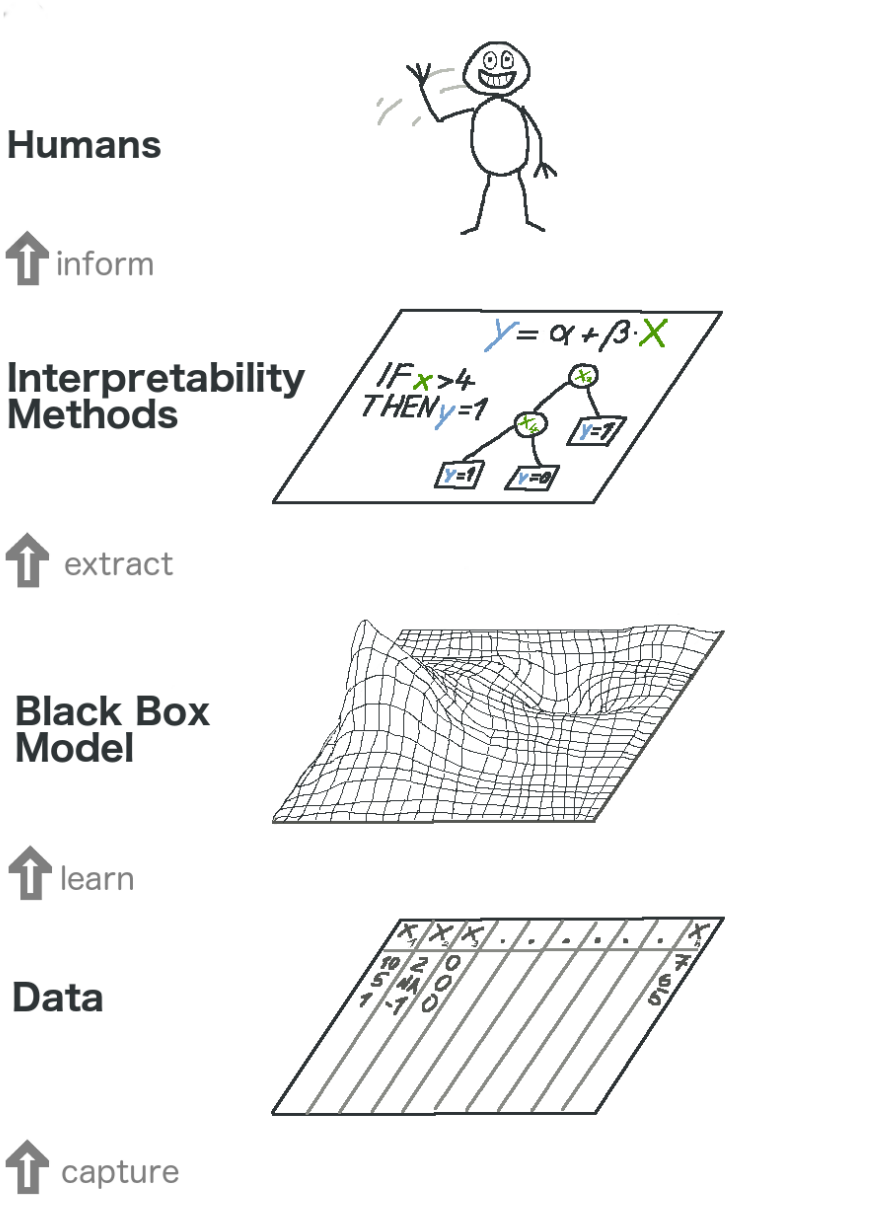
explainable ML section 3

Kristin Chen

10/15/2020

Introduction

- machine learning: An algorithm trains a model that produces the predictions.
  - We capture the world by collecting data, and abstract it further by learning to predict the data (for the task) with a machine learning model. Interpretability is just another layer on top that helps humans understand.



Taxonomy of interpretability

- intrinsic vs post-hoc: interpretability is achieved
  - intrinsic: by restricting the complexity of the machine learning model that are considered interpretable due to their simple structure, such as short decision trees or sparse linear models.
  - post-hoc: by applying methods that analyze the model after training
- results of interpretation methods:
  - feature summary statistic: provide summary statistics for each feature. Some methods return a single number per feature, such as feature importance, or a more complex result, such as the pairwise feature interaction strengths, which consist of a number for each feature pair.
  - feature summary visualization: some feature summaries are actually only meaningful if they are visualized. Partial dependence plots are curves that show a feature and the average predicted outcome.
  - Model internals: The interpretation of intrinsically interpretable models, such as weights in linear models or the learned tree structure (the features and thresholds used for the splits) of decision trees.
  - data point: includes all methods that return data points (already existent or newly created) to make a model interpretable, i.e. counterfactual explanations, which refers to explain the prediction of a data instance, the method finds a similar data point by changing some of the features for which the predicted outcome changes in a relevant way (e.g. a flip in the predicted class).

Taxonomy of interpretability (cont)

- model-specific or model-agnostic?
  - Model-specific interpretation tools are limited to specific model classes. The interpretation of regression weights in a linear model is a model-specific interpretation, since – by definition – the interpretation of intrinsically interpretable models is always model-specific.
  - Model-agnostic tools can be used on any machine learning model and are applied after the model has been trained (post hoc). These agnostic methods usually work by analyzing feature input and output pairs.
- local vs global
  - local: the interpretation method explain an individual prediction. Locally, the prediction might only depend linearly or monotonically on some features, rather than having a complex dependence on them.
  - global: entire model behavior. the global level of interpretability is about understanding how the model makes decisions, based on a holistic view of its features and each of the learned components such as weights, other parameters, and structures. Which features are important and what kind of interactions between them take place? Global model interpretability helps to understand the distribution of your target outcome based on the features.

example dataset

- bike rentals (regression)
- risk factors for cervical cancer (classification)

model-agnostic + global (average out the y-hat): PDP (plot of partial dependence; average marginal effect)

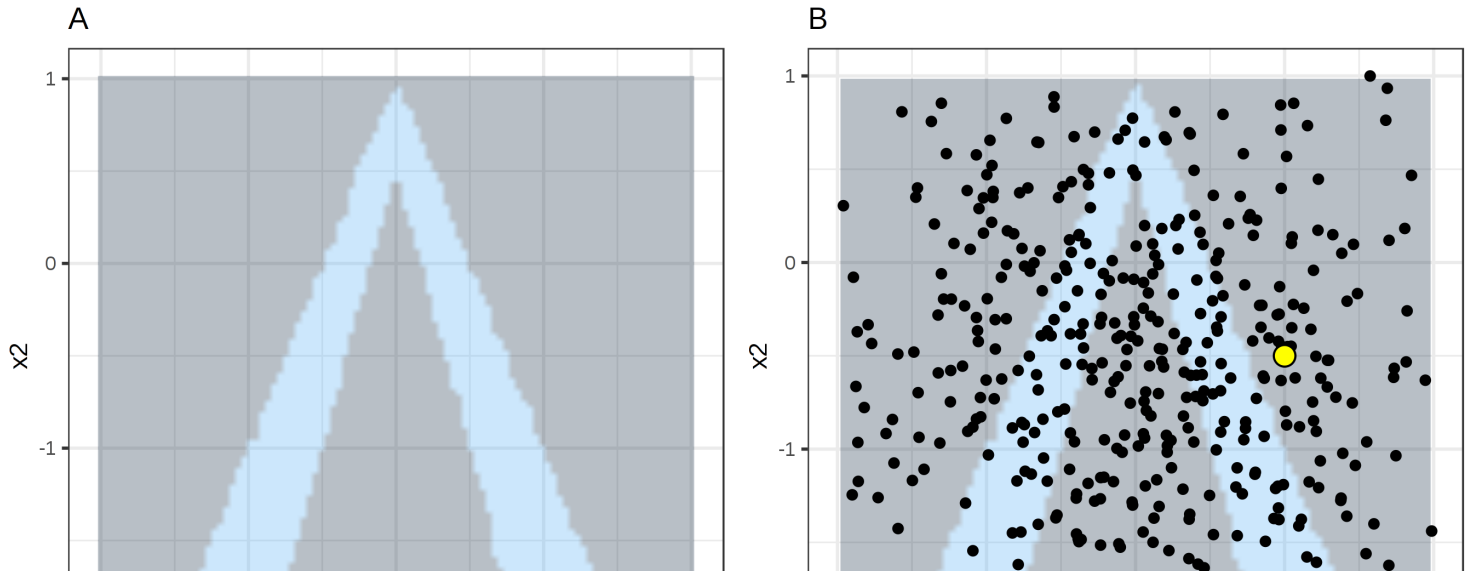
- The partial dependence plot (short PDP or PD plot) shows how the **average** predicted outcome of a machine learning model changes when the i-th feature is changed by marginalizing the predicted outcome over the distribution of all the other features.
  - A partial dependence plot can show whether the relationship between the target and a feature is *linear*, *monotonic* or *more complex*.
  - for regression: the PDP displays the change of y-hat conditionally on x-ith on average
  - for classification: the PDP displays the probability for a certain class given different values for feature(s) in i-th
- example

model-agnostic + global (average out the y-hat): PDP (cont.)

- pros & cons
  - pros: intuitive and introduce casual interpretation: If the feature for which you computed the PDP is *not correlated* with the other features, then the PDPs perfectly represent how the feature influences the prediction on average. In addition, the relationship shown in the PDP is **causal** for the model because we explicitly model the outcome as a function of the features (but not necessarily for the real world!)
  - cons:
    - The realistic maximum number of features in a partial dependence function is two (not the drawback of the method, but the visualization)
    - Some PD plots do not show the feature distribution.
    - The assumption of **independence** is the biggest issue with PD plots. It is assumed that the feature(s) for which the partial dependence is computed are not correlated with other features. (Accumulated Local Effect plots or short ALE plots that work with the conditional instead of the marginal distribution.)
    - Heterogeneous effects might be hidden because PD plots only show the average marginal effects. (individual conditional expectation curves could uneval heterogeneous effects)

model-agnostic + local (local surrogate model): LIME

- Surrogate models are trained to approximate the predictions of the underlying black box model. LIME tests what happens to the predictions when you give variations of your data into the machine learning model. LIME generates a new dataset consisting of permuted samples and the corresponding predictions of the black box model. On this new dataset LIME then trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest. The interpretable model can be anything from the interpretable models chapter, for example Lasso or a decision tree.
  - Select your instance of interest for which you want to have an explanation of its black box prediction.
  - Perturb your dataset and get the black box predictions for these new points, drawing from a normal distribution with mean and standard deviation taken from the feature.
  - Weight the new samples according to their proximity to the instance of interest.
  - Train a weighted, interpretable model on the dataset with the variations. (local fidelity: The learned model should be a good approximation of the machine learning model predictions locally, but it does not have to be a good global approximation.)
  - Explain the prediction by interpreting the local model.



model-agnostic + local: LIME (cont.)

- pros & cons
  - pros
    - Even if you replace the underlying machine learning model, you can still use the same local, interpretable model for explanation.
    - LIME is one of the few methods that works for tabular data, text and images.
    - A regression model can rely on a non-interpretable transformation of some attributes, but the explanations can be created with the original attributes. For instance, A text classifier can rely on abstract word embeddings as features, but the explanation can be based on the presence or absence of words in a sentence.
  - cons
    - The correct definition of the neighborhood is a very big, unsolved problem when using LIME with tabular data. (LIME currently uses an exponential smoothing kernel to define the neighborhood. The kernel width determines how large the neighborhood is: A small kernel width means that an instance must be very close to influence the local model, a larger kernel width means that instances that are farther away also influence the model.)
    - Data points are sampled from a Gaussian distribution, ignoring the correlation between features. This can lead to unlikely data points which can then be used to learn local explanation models.
    - The instability of the explanations. (vs Shapley Value)

model-agnostic + local (average out the x i-th): Shapley Value

- Shapley Value: the average marginal contribution of a feature value across all possible coalitions
  - The interpretation of the Shapley value for feature value j is: The value of the j-th feature contributed j to the prediction of this particular instance compared to the average prediction for the dataset.
  - `Note that: The Shapley value is *the average contribution* of a feature value t`
  - for regression
  - for classification
- example

model-agnostic + local (average out the x i-th): Shapley Value (cont.)

- pros & cons
  - pros (vs LIME)
    - the average prediction is *fairly distributed* among the feature values of the instance
    - carry over efficiency, symmetry, dummy and additivity axioms
  - cons
    - requires a lot of computing time: there are 2k possible coalitions of the feature values and the "absence" of a feature has to be simulated by drawing random instances, which increases the variance for the estimate of the Shapley values estimation. The exponential number of the coalitions is dealt with by sampling coalitions and limiting the number of iterations M. Decreasing M reduces computation time, but increases the variance of the Shapley value.
    - Shapley value is the wrong explanation method if you seek sparse explanations (explanations that contain few features). SHAP could be the alternative solution, LIME could select features as well.
    - The Shapley value returns a simple value per feature, but no prediction model like LIME. This means it cannot be used to make statements about changes in prediction for changes in the input, such as: "If I were to earn €300 more a year, my credit score would increase by 5 points."
    - need to access to the data
    - the Shapley value method suffers from inclusion of unrealistic data instances when features are correlated. To simulate that a feature value is missing from a coalition, we marginalize the feature. This is achieved by sampling values from the feature's marginal distribution. This is fine as long as the features are independent. When features are dependent, then we might sample feature values that do not make sense for this instance.