Handling Missing Data in Decision Trees: A Probabilistic Approach

Pasha Khosravi, Antonio Vergari, YooJung Choi, Yitao Liang, Guy Van den Broeck

UCLA

STAR AI

Overview

- Missing values common occurrence in machine Learning
  - Hinders performance of discriminative models
  - Generative models can handle missing values but not as good in discriminating (classification/regression).

- Decision trees are a popular family of models

- This paper: learning parameters of decision trees from missing data, using tractable density estimators.

Decision Trees/ Random Forests

Popular models:

- Scalability
- Interpretability
- Ability to handle mixed types of features (discrete vs real)

- Used for both regression and classification

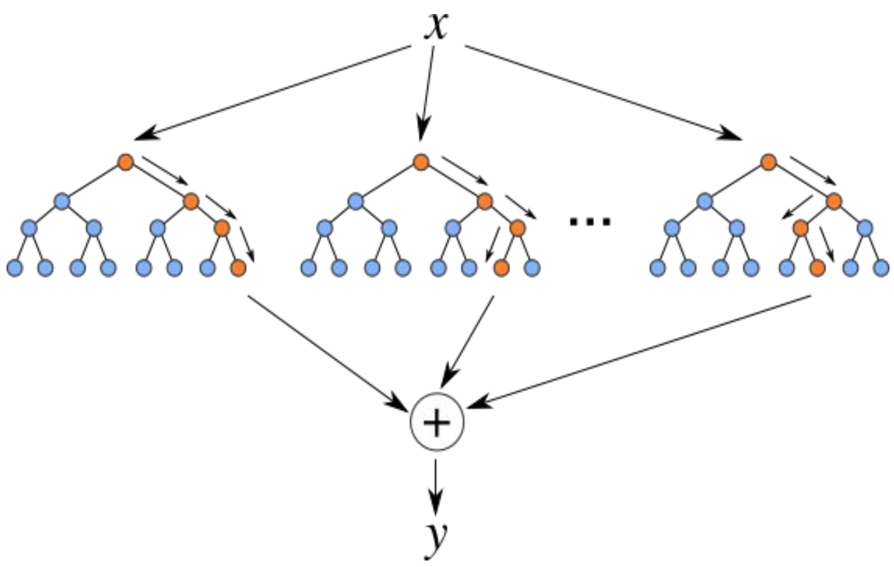
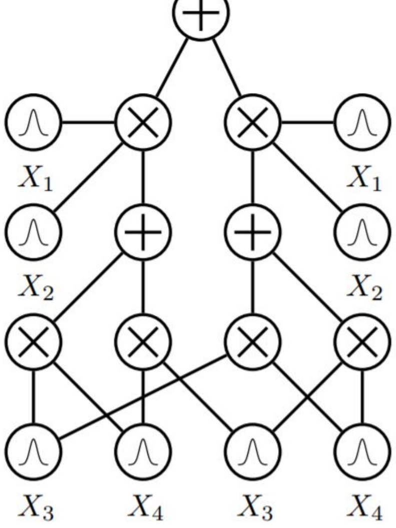


Figure Credit link

Probabilistic Circuits

- Can be thought of as "deep mixture models"
  - Expressive density estimators
  - Tractable probabilistic queries such as **exact** marginalization on ***any subset*** of features in **linear time**
  - Both structure and parameters can be learned from missing data
  - They are a computation graph, so can differentiate



Check out tutorial: Probabilistic Circuits: Inference, Representations, Learning and Theory

Expected loss minimization

$$\mathcal{L}(\Theta; \mathsf{D}_{\mathsf{train}}) = \frac{1}{|\mathsf{D}_{\mathsf{train}}|} \sum_{\mathbf{x}^o, y \in \mathsf{D}_{\mathsf{train}}} \mathbb{E}_{p_\Phi(\mathbf{X}^m | \mathbf{x}^o)} \left[l(y, f_\Theta(\mathbf{x})) \right]$$

For one tree and using loss = MSE, can be computed exactly:

$$\theta_\ell^* = \frac{\sum_{\mathbf{x}^o, y \in \mathsf{D}_{\mathsf{train}}} y \cdot p_\ell(\mathbf{x}^o)/p(\mathbf{x}^o)}{\sum_{\mathbf{x}^o, y \in \mathsf{D}_{\mathsf{train}}} p_\ell(\mathbf{x}^o)/p(\mathbf{x}^o)}$$

More scenarios such as bagging/boosting in the paper.

Preliminary Experiments