Gaussian Soft Decision Trees for Interpretable Feature-Based Classification

Jaemin Yoo[1] and **Lee Sael**[2]

[1] Seoul National University

[2] Ajou University

PAKDD 2021

Outline

- <u>**Introduction**</u>
- Previous Works
- Proposed Method
- Experiments
- Conclusion

Black Box

- Deep neural network is a **black box**
  - Its decision process is not interpretable
  - Difficult to trust decisions even with high accuracy

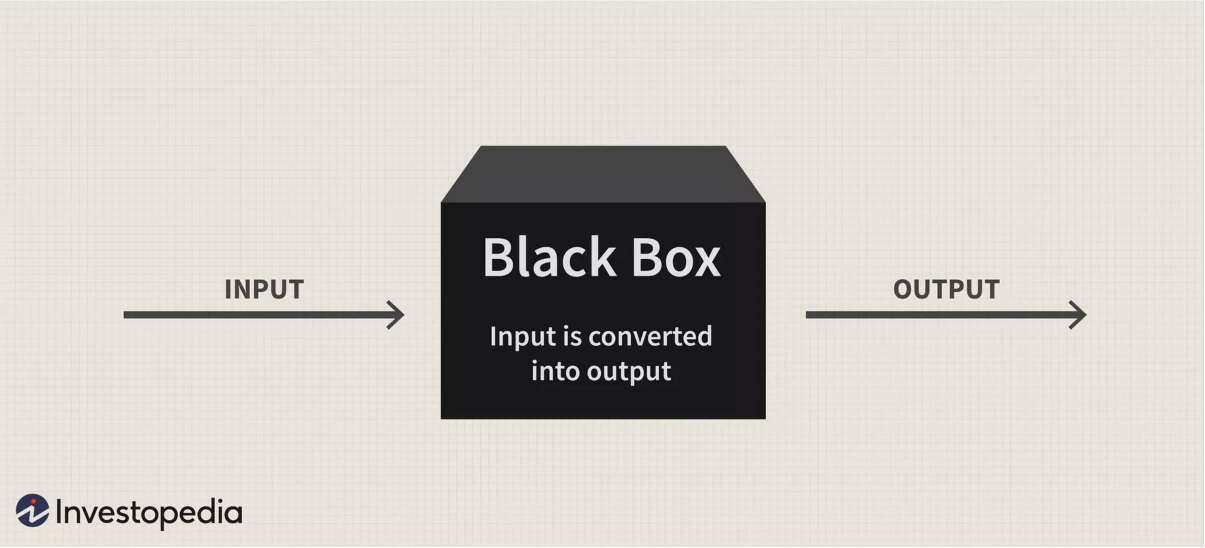


Image from https://www.investopedia.com/terms/b/blackbox.asp

Interpretable ML (1)

- Research to interpret a model's decisions
  - Important especially in bio or medical domains
- **Global interpretability**
  - A model's decision process is itself interpretable
    - Linear models or decision trees
- **Local interpretability**
  - To explain decisions made by black box models
    - Recent works for deep neural networks

Interpretable ML (2)

- Research to interpret a model's decisions
  - Important especially in bio or medical domains
- **Global interpretability**
  - A model's decision process is itself interpretable
  - Global interpretability makes **reliable decisions**
- **Feature-based classification**
  - Simple models can be better than neural networks
  - Generalizability is more important the capability

Tree Models

- Tree models provide global interpretability
  - Each decision is represented as a path in the tree, which has its own meaning

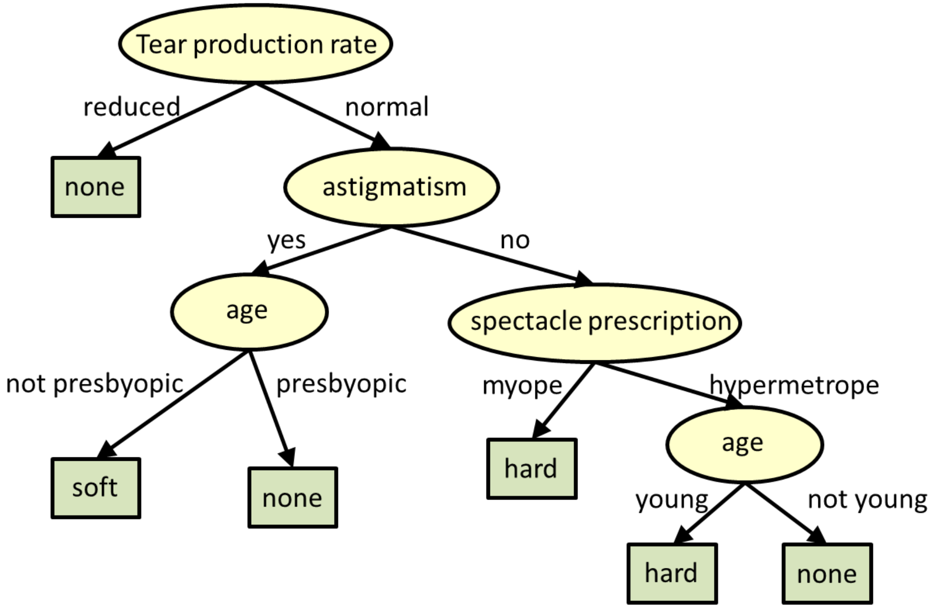


Image from https://www.cs.cmu.edu/~bhiksha/courses/10-601/decisiontrees/

Limitations of Tree Models

- **Linear decisions**
  - Restrict the overall representation power
  - Make it difficult to learn complex decision rules
- **Large tree depth**
  - Limits the interpretability of models
  - Tree depth means the complexity of interpretation
  - *Is a tree still interpretable with large depth* $d > 10$?

Problem Definition

- **Given** a feature-based dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_i$
  - No structural information exists in $\mathbf{x}$
  - Each element in $\mathbf{x}$ is itself meaningful
- **Train** an interpretable tree classifier f
- **Maximizing** its accuracy and interpretability
  - Addressing the limitations of previous models

Outline

- Introduction
- **Previous Works**
- Proposed Method
- Experiments
- Conclusion

Decision Trees

- One of the most popular tree models
  - Has been used for decades
  - Learns an explicit decision rule at each branch
    - For instance, to pass $\mathbf{x}$ to the left child if $x_3 > 3$
- **Strength**
  - Its decision process is clear and interpretable
- **Weakness**
  - It easily overfits, making limited performance

Soft Decision Trees (1)

- Improve the representation power of DTs
  - Perform a soft decision with all features
  - Learn a soft target distribution at each leaf
- **Strength**
  - Larger capability to learn complex decision rules
- **Weakness**
  - Less interpretability due to the soft decisions

Soft Decision Trees (2)

- SDTs are characterized by **soft decisions**
- The probability f_i at node i to pass $\mathbf{x}$ to the right child is

$$f_i = \sigma\left(\mathbf{w}_i^\top \mathbf{x} + b_i\right)$$

  - $\mathbf{w}_i$ and b_i are learnable parameters at node i
  - σ is the sigmoid function for the split
  - The probability to the left is $1 - f_i$ thanks to $\sigma(\cdot)$

Soft Decision Trees (3)

- The interpretability is worse than that of DTs
  - Because all features are used for every decision
  - Each decision path involves $O(dm)$ parameters
    - d is the depth, and m is the number of features
- EDiT (ICDM 2019) focused on decreasing m
  - It learns a sparse weight vector at each branch
  - However, the large depth d remains the same

Outline

- Introduction
- Previous Works
- **Proposed Method**
- Experiments
- Conclusion

Overview (1)

- **Gaussian Soft Decision Trees (GSDT)**
  - Tree model having a multi-branched structure
  - Decisions are modeled as Gaussian mixtures
  - Address the limitations of previous tree models
- **Main ideas**
  - Gaussian decisions
  - Low-rank perturbation
  - Path regularization
  - Post-optimization

Overview (2)

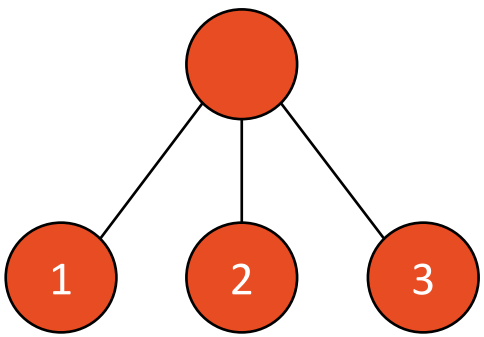
- GSDT first computes the arrival probability $\mathbf{r}(\mathbf{x})$
- Then, the prediction is done by a single leaf:

$$\hat{y}(\mathbf{x}) = \mathbf{p}_i \quad \text{where} \quad i = \mathrm{argmax}_k\, r_k(\mathbf{x})$$

  - $\mathbf{p}_i$ is the class distribution learned by leaf i
- The training is done by a gradient-based way
  - All parameters are updated at the same time
  - We minimize the hinge loss for classification

Gaussian Decisions (1)

- We make all decisions as Gaussian mixtures
- This enables us to preserve the interpretability even with the nonlinearity of decisions



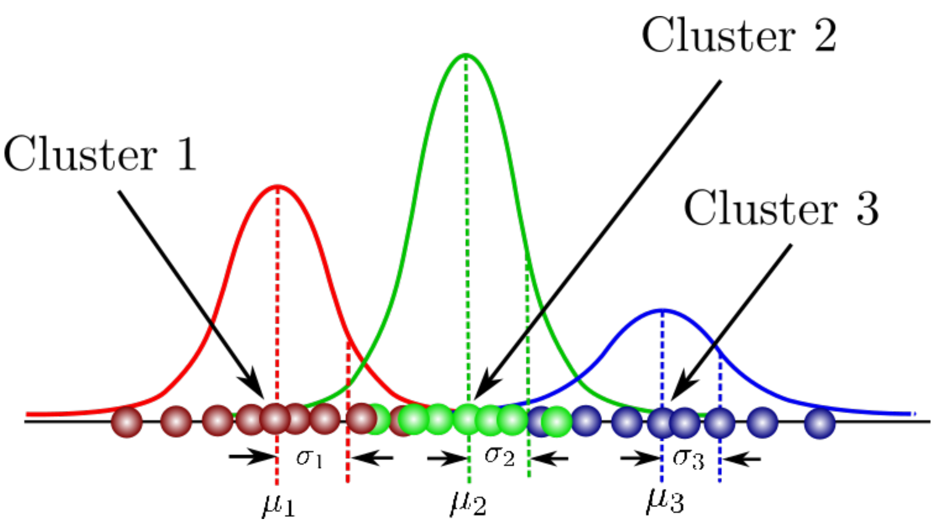


Image from https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

Gaussian Decisions (2)

- The probability $f_{ij}(\mathbf{x})$ of $\mathbf{x}$ from node i to j is

$$f_{ij}(\mathbf{x}) = \frac{\exp(\mathcal{L}(\theta_j \mid \mathbf{x}))}{\sum_k \exp(\mathcal{L}(\theta_k \mid \mathbf{x}))},$$

  - $\mathcal{L}$ is the log likelihood of $\mathbf{x}$, which is defined as

$$\mathcal{L}(\theta_j \mid \mathbf{x}) = -\frac{1}{2}\left((\mathbf{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j) + \log\det(\boldsymbol{\Sigma}_j) + d\log(2\pi)\right).$$

  - μ_j and Σ_j are learned through backpropagation

Gaussian Decisions (3)

- Gaussian decisions make several advantages
  - **Nonlinearity**
    - Each branch can learn a complex decision function
  - **Interpretability of decisions**
    - $f_{ij}(\mathbf{x})$ is itself interpretable as a probability
  - **Interpretability of parameters**
    - μ_i summarizes the examples arriving at node i
    - Σ_i gives insights about the given features
      - E.g., which feature is more important than others?

Gaussian Decisions (4)

- What if we apply multiple branches directly to soft decision trees?
  - It makes multiple children at each branch as

$$\mathbf{p}(\mathbf{x}) = \mathrm{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

  - However, $\mathbf{p}$ becomes no longer interpretable
    - $w_{ij} \neq$ the correlation between x_j and p_i

Low-Rank Perturbation (1)

- It is burdensome to learn a full matrix Σ_i
  - Because of the $\log\det(\Sigma_i)$ and Σ_i^{-1} operations
- Diagonal covariance is a simple choice
  - But it ignores the correlations between features
- We propose **low-rank perturbation**
  - Strengthen the diagonal Σ_i with correlations
  - Involve only $O(m)$ additional parameters

Low-Rank Perturbation (2)

- Our covariance matrix at each node i is

$$\boldsymbol{\Sigma}_i = \mathrm{diag}(\log(1 + \exp(\boldsymbol{\sigma}_i))) + \mathbf{U}\mathbf{U}^\top$$

  - $\boldsymbol{\sigma_i} \in \mathbb{R}^m$ is a learnable vector
  - $\mathbf{U} \in \mathbb{R}^{m \times k}$ is a learnable matrix
  - k is the target rank
    - We set k to 1 or 2 in experiments

Path Regularization

- How to encourage GSDT to utilize all leaves?
  - GSDT is prone to use only a few leaf nodes
- We add the **path regularizer** to the objective

$$l_{\mathrm{lr}}(\mathcal{B}) = \sum_{j \in \mathcal{N}_d} r_j(\mathcal{B}) \log r_j(\mathcal{B}) \quad \text{where} \quad \mathbf{r}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \mathbf{r}(\mathbf{x}),$$

  - $l_{\mathrm{lr}}(\mathcal{B})$ calculates the negative entropy of $\mathbf{r}(\mathcal{B})$
  - $\mathbf{r}(\mathcal{B})$ is the mean arrival probability for batch $\mathcal{B}$

Post-Optimization (1)

- Each leaf i corresponds to a set of examples
  - That arrive at the leaf node i at the inference
  - $\mathcal{X}_i = \{x \in \mathcal{D} |\, \mathrm{argmax}_k\, r_k(\mathbf{x}) = i\}$
- **Post-optimization**
  - Our technique to maximize the correspondence
  - We make the dist. $\mathcal{N}_i$ represent the examples $\mathcal{X}_i$
  - The interpretability of leaves further improves

Post-Optimization (2)

- The algorithm is given as follows:
  - μ_j and Σ_j are updated to represent the set $\mathcal{X}_j$

Algorithm 1: Post-optimization of the leaf Gaussians of GSDT.

Input: A trained GSDT M, a set $\mathcal{D}$ of training features, a learning rate α for the covariances, and the number n of iterations

```
for leaf node j in M do
    X_j ← {x ∈ D | arg max_k r_k(x) = j}
    μ_j ← Σ_{x ∈ X_j} x
    for i ∈ [1, n] do
        l ← sum((Σ_j − cov(X_j))^2)
        Σ_j ← Σ_j − α · ∂l/∂Σ_j
    end for
end for
Fine-tune the whole parameters of M for a fixed number of epochs
```

Outline

- Introduction
- Previous Works
- Proposed Method
- <u>**Experiments**</u>
- Conclusion

Experimental Setup

- **Datasets**
  - We use six public feature-based datasets
  - Taken from UCI Repository or Kaggle
  - All of them are bio and medical domains
    - Interpretability is a crucial factor
- **Baselines**
  - Interpretable models: LR, SVM, DT, SDT, EDiT
  - Black box models: RF, MLP

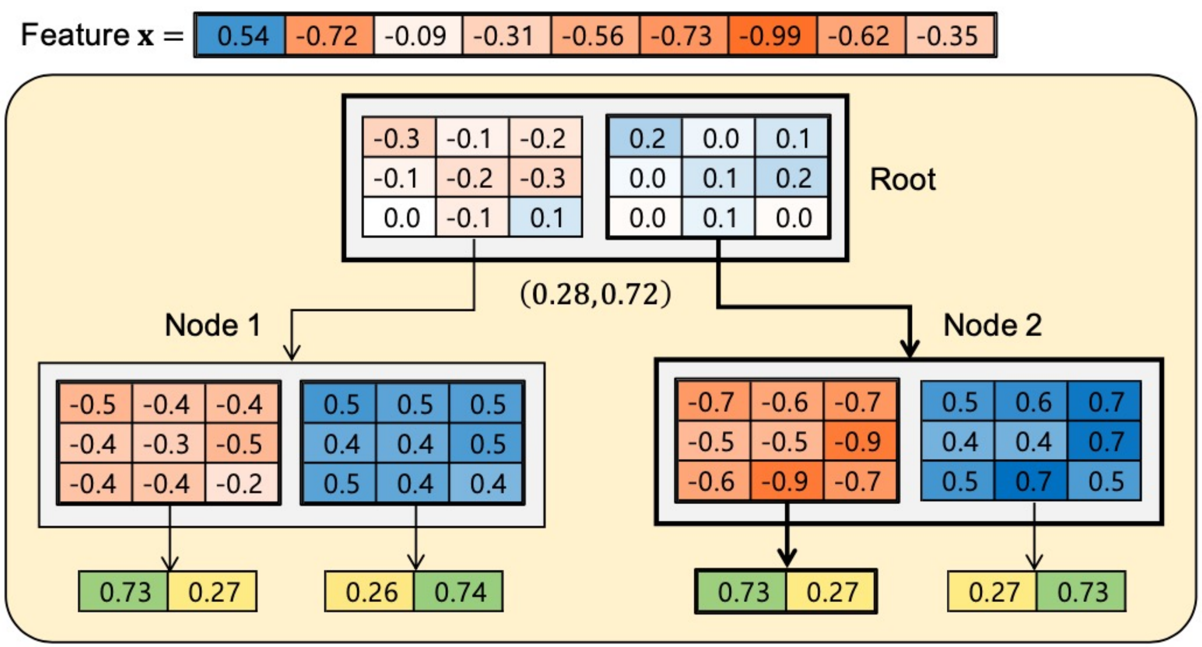
Classification Accuracy

- GSDT shows the best accuracy in five datasets
  - GSDT outperforms even strong black box models

Model	Brain	Breast	Breast-wis	Diabetes	Heart	Hepatitis
LR	63.4 ± 0.0	65.5 ± 0.0	97.1 ± 0.0	$\underline{76.0 \pm 0.0}$	$\mathbf{86.9 \pm 0.0}$	77.4 ± 0.0
SVM-lin	61.0 ± 0.0	62.1 ± 0.0	97.1 ± 0.0	$\mathbf{76.6 \pm 0.0}$	83.6 ± 0.0	77.4 ± 0.0
SVM-rbf	58.5 ± 0.0	70.7 ± 0.0	97.1 ± 0.0	$\underline{76.0 \pm 0.0}$	$\mathbf{86.9 \pm 0.0}$	77.4 ± 0.0
DT	70.5 ± 0.7	68.8 ± 1.6	96.0 ± 0.9	69.7 ± 1.6	67.2 ± 1.6	70.0 ± 6.9
SDT	66.8 ± 5.0	73.3 ± 5.2	97.9 ± 0.0	$\underline{76.0 \pm 0.7}$	80.7 ± 2.7	67.3 ± 4.7
EDiT	58.5 ± 0.0	75.0 ± 2.6	97.1 ± 0.2	74.6 ± 1.5	85.2 ± 2.3	$\underline{77.8 \pm 3.8}$
MLP	$\underline{73.4 \pm 1.7}$	73.3 ± 2.3	$\underline{98.6 \pm 0.2}$	75.0 ± 0.8	80.5 ± 1.5	64.2 ± 3.0
RF	68.0 ± 2.3	$\underline{76.6 \pm 0.8}$	98.1 ± 0.3	73.4 ± 0.7	84.8 ± 0.8	70.3 ± 2.4
GSDT	$\mathbf{73.5 \pm 1.5}$	$\mathbf{77.2 \pm 1.7}$	$\mathbf{98.8 \pm 0.6}$	$\underline{76.0 \pm 0.9}$	$\mathbf{86.9 \pm 1.2}$	$\mathbf{78.2 \pm 3.1}$

Structure Visualization

- GSDT provides a clear decision process
  - Each mean vector is a representative of the path



Breast-cancer-wisconsin

Features

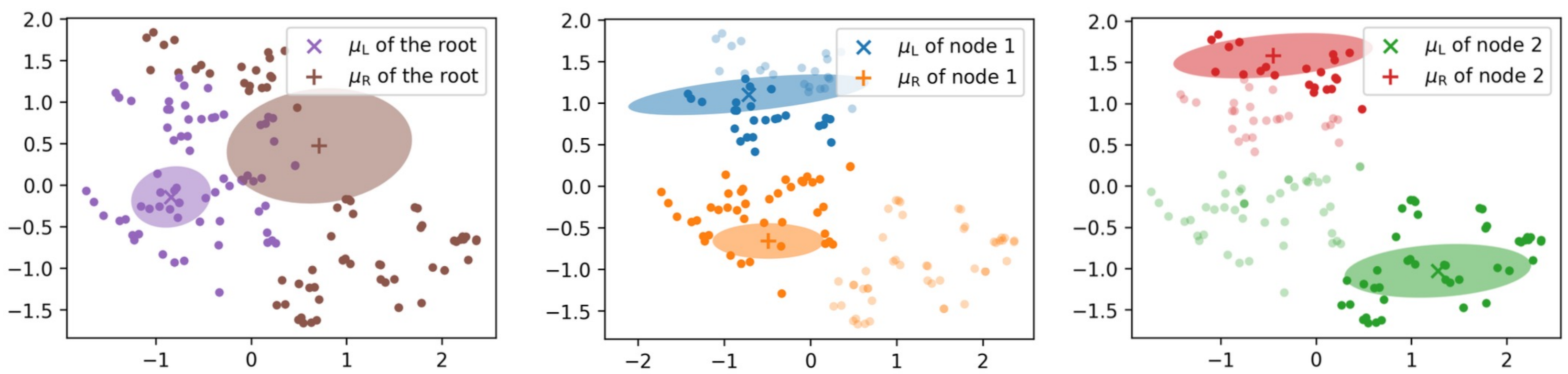
x_1: Clump thickness
x_2: Uniformity of cell size
x_3: Uniformity of cell shape
x_4: Marginal adhesion
x_5: Single epithelial cell size
x_6: Bare nuclei
x_7: Bland chromatin
x_8: Normal nucleoli
x_9: Mitoses

Diagnosis (as labels)

y_1: Benign
y_2: Malignant

Learned Distributions

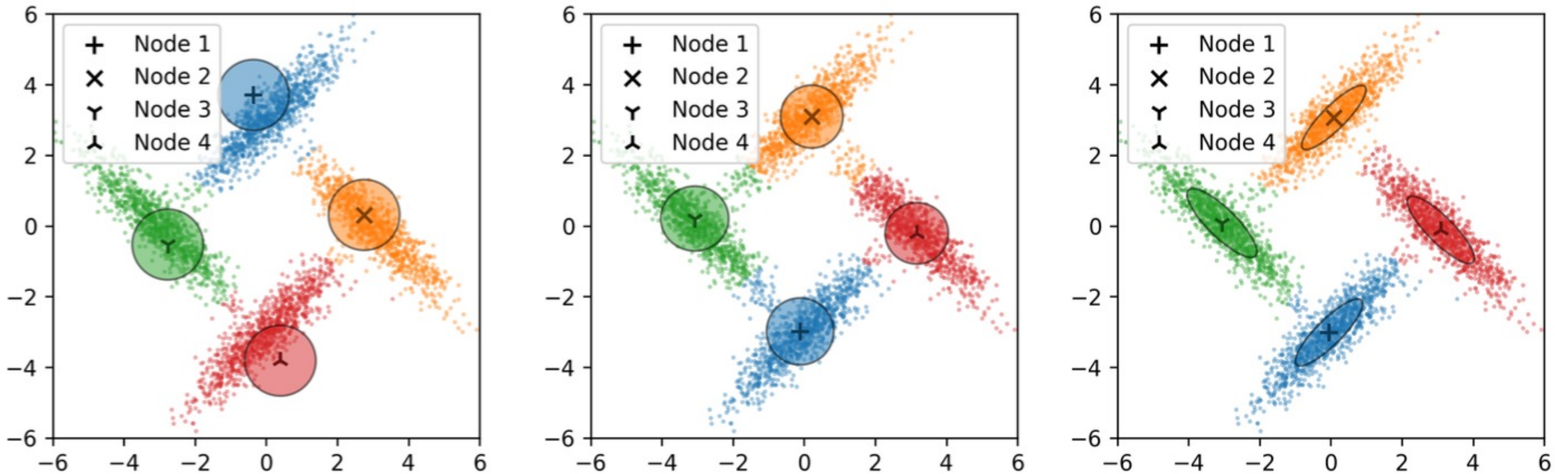
- GSDT learns meaningful node distributions
  - The root node splits examples horizontally
  - The internal nodes split examples vertically
    - Nodes 1 and 2 take different sets of examples



(b) Decision by the root. (c) Decision by node 1. (d) Decision by node 2.

Covariance Matrix

- Our low-rank perturbation makes the best fit
  - The identity and diagonal covariances are simple but fail to model the given distributions



(a) Identity. (b) Only diagonal. (c) Low-Rank Perturbed.

Outline

- Introduction
- Previous Works
- Proposed Method
- Experiments
- **<u>Conclusion</u>**

Conclusion

- **Gaussian Soft Decision Trees (GSDT)**
  - Our novel tree model for interpretable learning
  - Multi-branched structure with nonlinear decisions
- Main ideas
  - Gaussian decisions with low-rank perturbation
  - Path regularization
  - Post-optimization
- Experiments
  - GSDT outperforms baselines with interpretability

Thank you!

Code and datasets:

https://github.com/leesael/GSDT