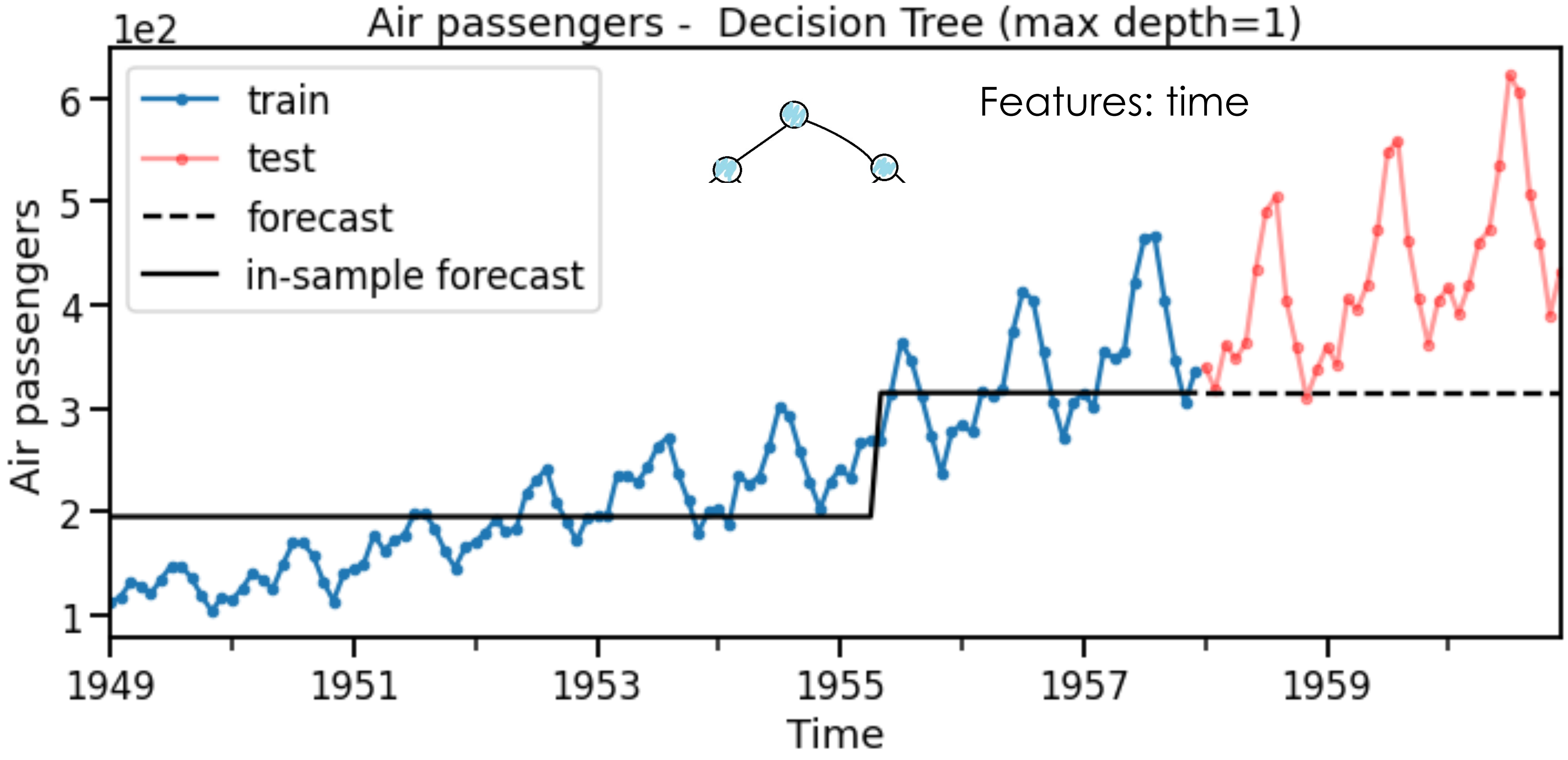
Linear trees to handle trend

Trend features

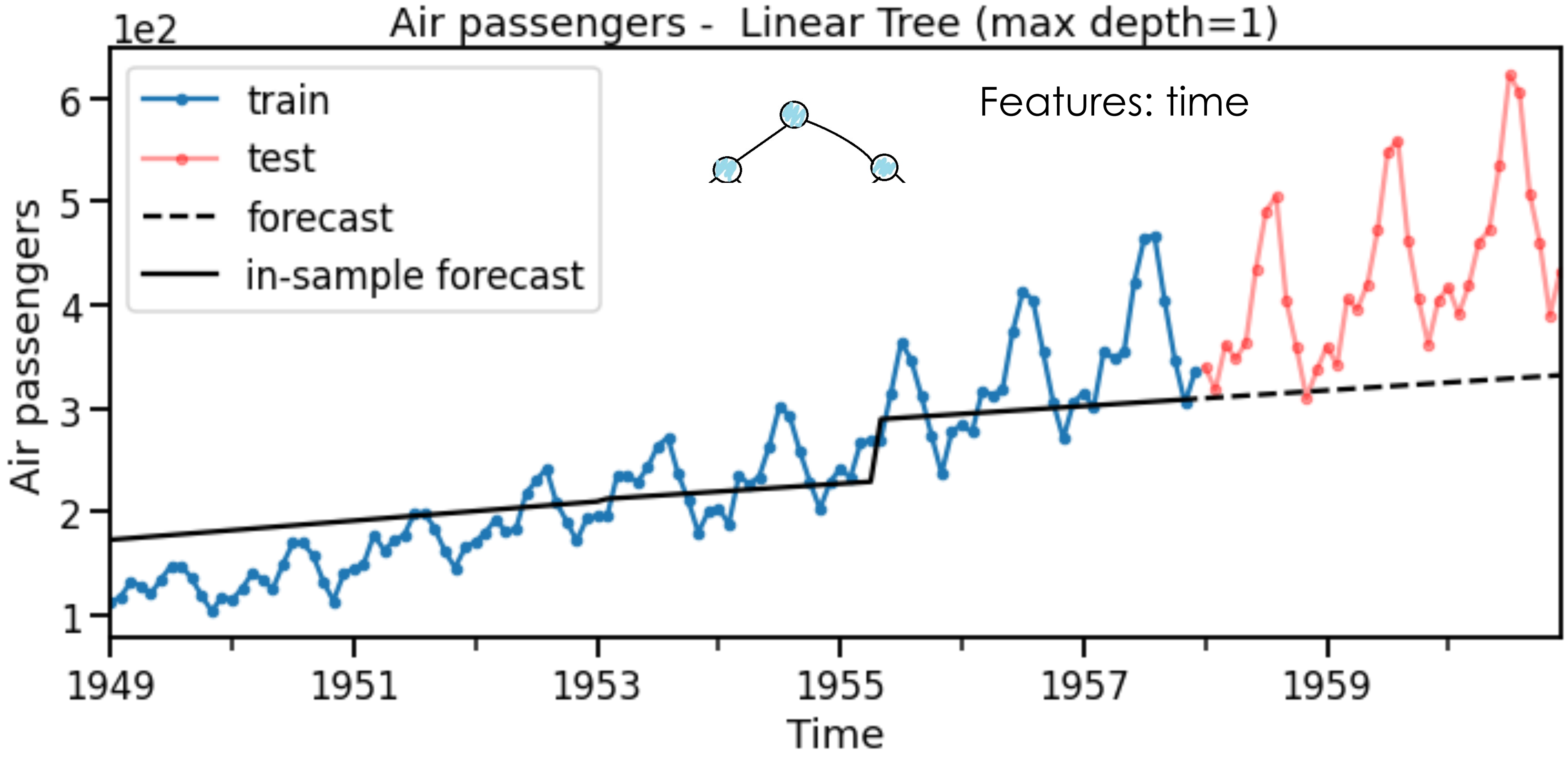
Standard tree-based models

Standard tree-based models use the **average of the target in the training data** in the leaf node to make predictions.



Linear trees

Linear trees use the **fit a separate linear model to the training data in each leaf node** and use the **linear models to make predictions.**



Paper that inspired the linear tree implementation in LightGBM

Gradient Boosting with Piece-Wise Linear Regression Trees

Yu Shi, **Jian Li** and **Zhize Li**
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
shiyu17@mails.tsinghua.edu.cn, lijian83@mail.tsinghua.edu.cn, zz-li14@mails.tsinghua.edu.cn

25 Jun 2019

Abstract

Gradient Boosted Decision Trees (GBDT) is a very successful ensemble learning algorithm widely used across a variety of applications. Recently, several variants of GBDT training algorithms and implementations have been designed and heavily optimized in some very popular open sourced toolkits including XGBoost, LightGBM and CatBoost.

replacing the constant values at the leaves by linear functions, so called *piecewise linear regression trees* (PL Trees). This idea has been explored in [Wang and Hastie, 2014; Hall *et al.*, 2009; Kuhn *et al.*, 2012]. However, due to its heavy computation cost, so far there's no fast and scalable implementation of gradient boosting with PL Trees.

In this paper, we provide a fast and scalable implementation of gradient boosting with PL Trees. Our algorithm has training cost comparable to carefully optimized GBDT toolk-

https://arxiv.org/pdf/1802.05640.pdf
https://github.com/microsoft/LightGBM/pull/3299

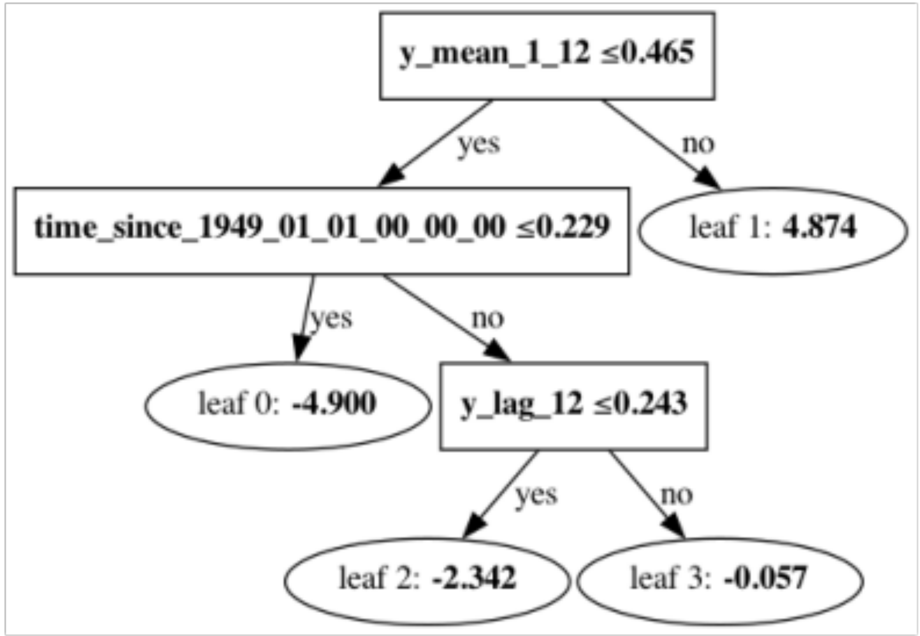
LightGBM implementation: what does it do?

- **Numeric features and categorical features are used when splitting the nodes** using the standard splitting criteria for gradient boosted trees (i.e., an impurity measure).

- A **linear model is fit** at the **leaf node <u>after</u>** the tree structure has been determined.

- **Only numeric features** are used **in the linear model** at each leaf. Each linear model **only includes** the **numerical features in that leaf's branch**.

- The **first tree** is **not a linear tree**. Instead a standard tree is used (i.e., average of target in training data) so we start with predictions close to the training set.



Source: [source 1](), [source 2](), [source 3]()

LightGBM implementation

```python
from lightgbm import LGBMRegressor # sklearn API
```

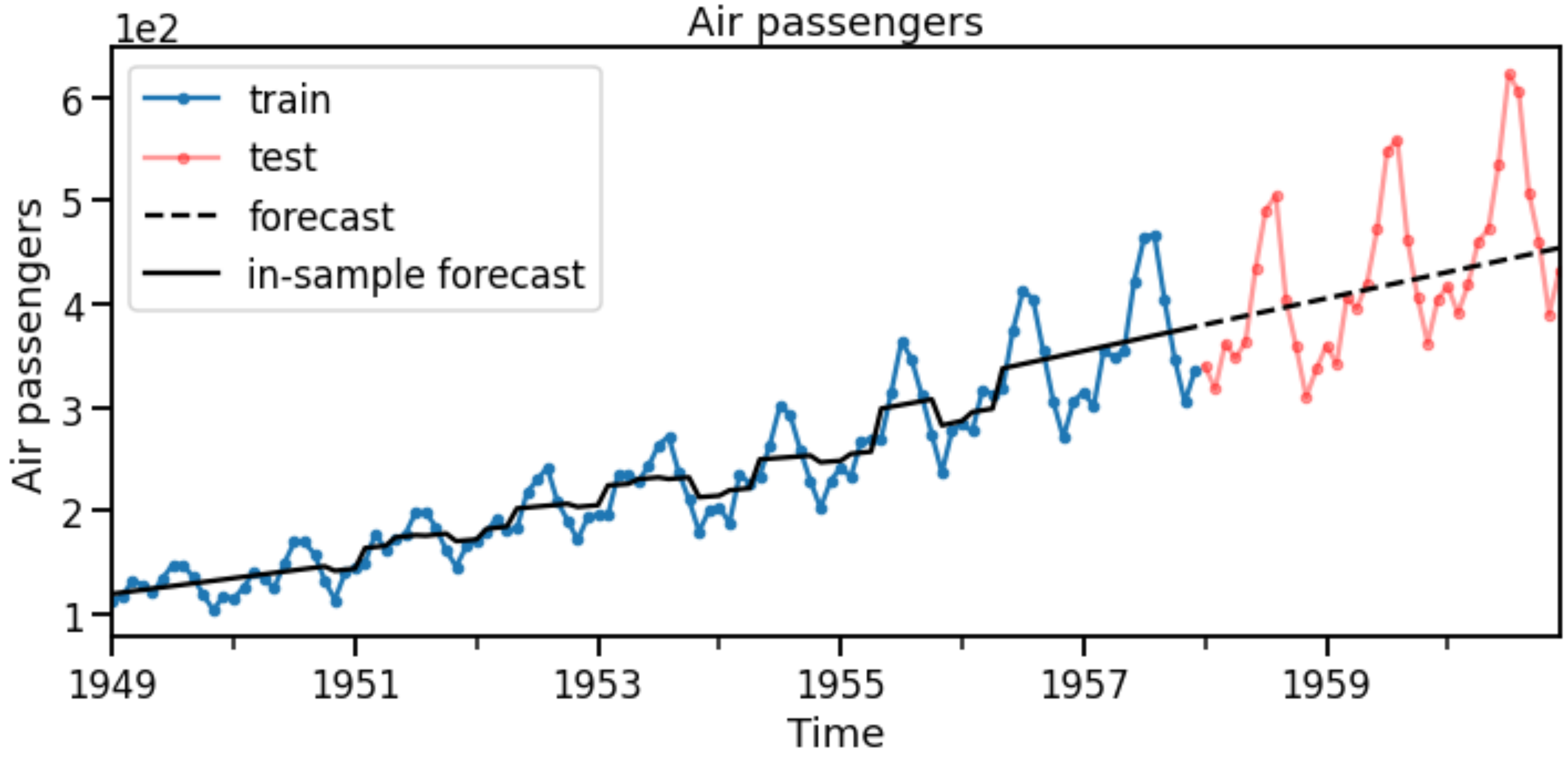
LightGBM implementation

```python
# Define the model.
model = LGBMRegressor(
                      boosting = "gbdt", # Gradient Boosted
                                         # Decision Trees.
                                         # Can also set `boosting`
                                         # to `rf` to us a
                                         # Random Forest model.
                      linear_tree=True, # Use linear trees.
                      linear_lambda=0 , # L2 regularisation
                                        # for linear models
                                        # in the leaves (i.e., Ridge).
                     )
```

LightGBM implementation – Random Forest mode

```
# Define the model.
model = LGBMRegressor(
                      boosting = "rf", # Gradient Boosted
                                        # Decision Trees.
                                        # Can also set `boosting`
                                        # to `rf` to us a
                                        # Random Forest model.
                      linear_tree=True, # Use linear trees.
                      linear_lambda=0 , # L2 regularisation
                                        # for linear models
                                        # in the leaves (i.e., Ridge).
                      n_estimators=100, # Number of boosting iterations
                                        # and therefore the number of trees.
                      bagging_freq = 1, # Required when using `rf` mode.
                                        # Ensures training samples are
                                        # bagged (i.e, sampled without replacement)
                                        # for every tree.
                      bagging_fraction = 0.8, # Required when using `rf` mode.
                                              # Determines size of sample
                                              # when bagging the rows.
                      feature_fraction = 0.8, # Required when using `rf` mode.
                                              # Determins size of column subset
                                              # to use for each tree.
                     )
```

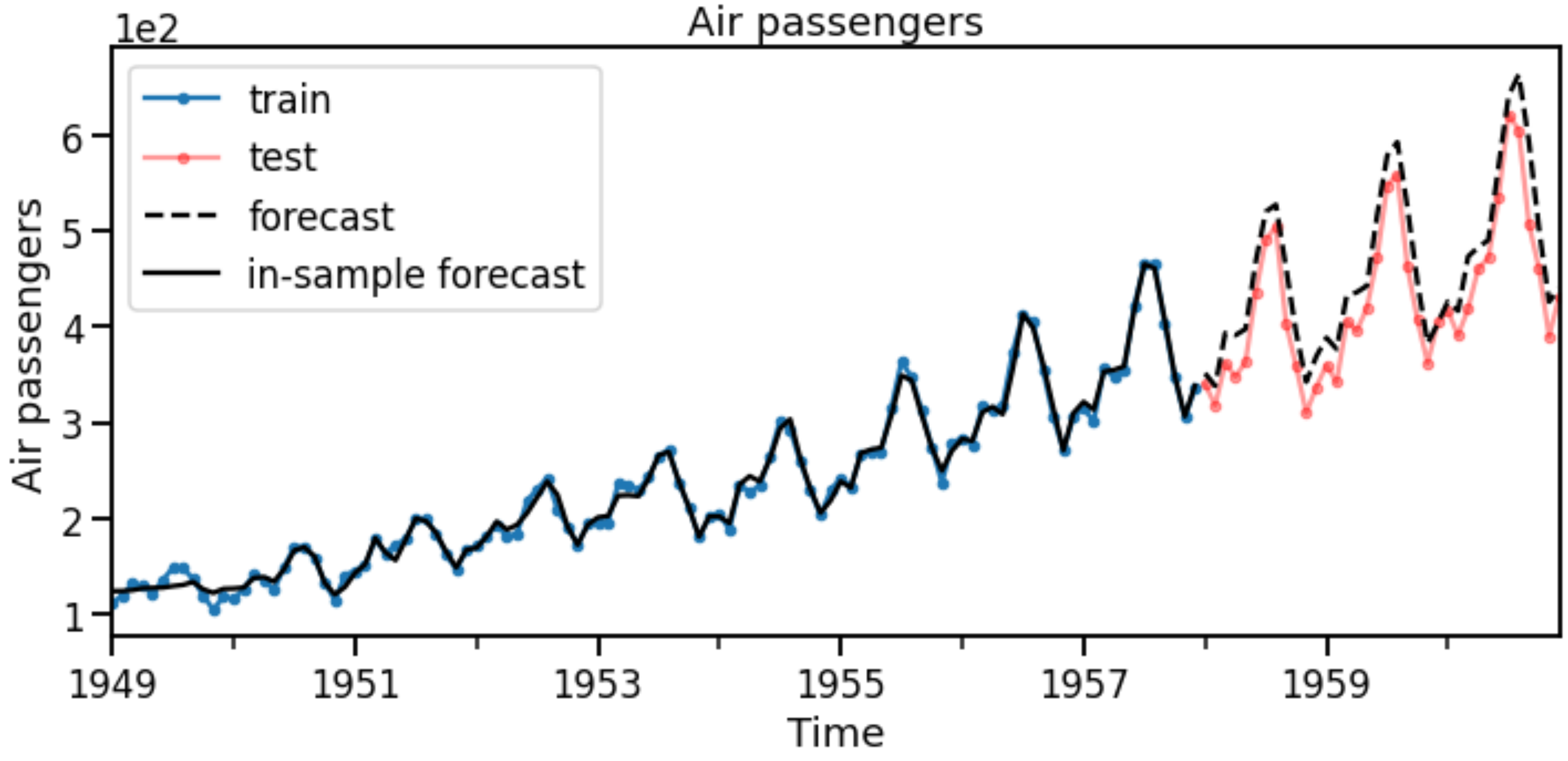
Example: linear trees on the air passengers dataset



Model:
- GBDT

Features:
- Time

Example: linear trees on the air passengers dataset



Model:
- GBDT

Features:
- Time
- lag 1, 2, 3, & 12
- Window mean of size 12

Summary

Standard tree-based models cannot extrapolate and so will struggle with trend.

Linear trees fit a linear model at the leaves and so can handle trend.

LightGBM implements gradient boosting and random forests using linear trees.