kokchun giang

improving the performance of decision tree by combining many trees using **random forest** and **XGBoost**

bagging to sample multiple datasets for multiple trees

Decision trees

+ interpretable
+ can visualize
+ no need to scale
+ handle qualitative variable

- ↓ predictive performance

improve by combining many decision trees

Bagging

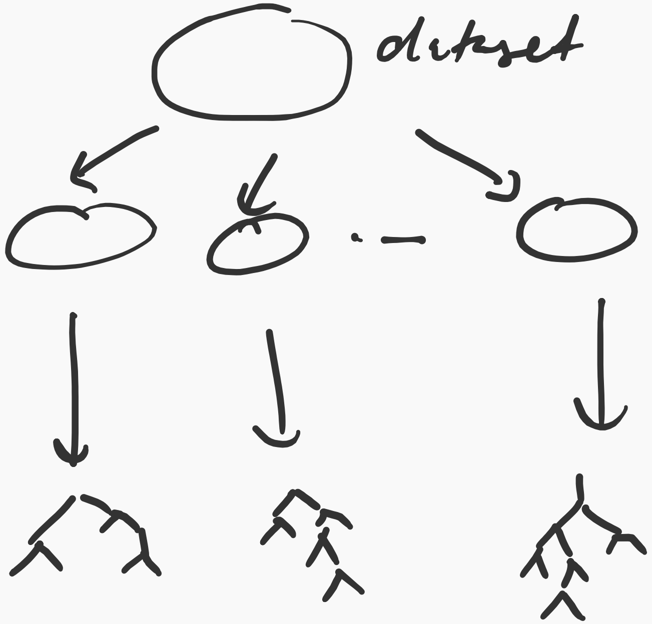
we sample from same dataset multiple times

=> bootstrapping

=> predict on each bootstrapped dataset and average

=> bagging

· majority vote in classification



Random forest

- build DTs on bootstrapped trng sets
- every split in tree is based on random choice of predictors

boosting performance by growing trees based on info from previously grown trees

Boosting

trees grown sequentially

each new tree attacks the residuals (error)

improving where it previously performed badly

→ learns slowly

Hyperparams

1. # trees B overfit if too large
2. shrinkage param λ controls learning rate
3. # splits d in each tree

→ control complexity

$d = 1$ usually works well

XGBoost

eXtreme Gradient Boosting

gradient boosting w.

- regularization ℓ_1, ℓ_2
- ↑ efficiency
- ↑ performance
- handle missing data

short note on **trees**

decision trees are interpretable but low performance

random forest & XGBoost combines many decision trees to improve performance

random forest & XGBoost are among the state of art algorithms for tabular data. Note however that in cases you might combine several algorithms