https://goo.gl/kfxwEg

KYOTO UNIVERSITY

Statistical Machine Learning Theory

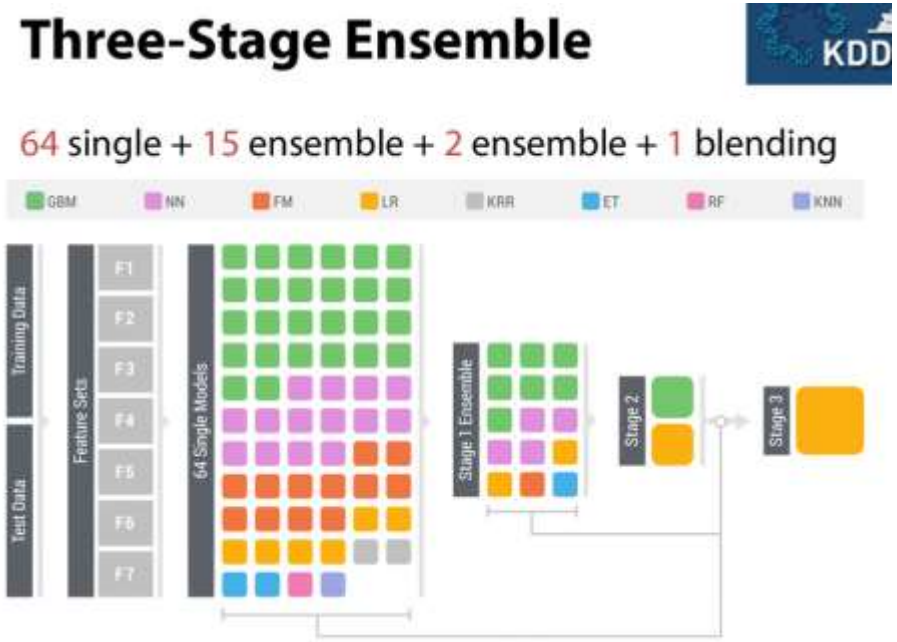
Model Ensemble

Hisashi Kashima
kashima@i.Kyoto-u.ac.jp

DEPARTMENT OF INTELLIGENCE SCIENCE AND TECHNOLOGY

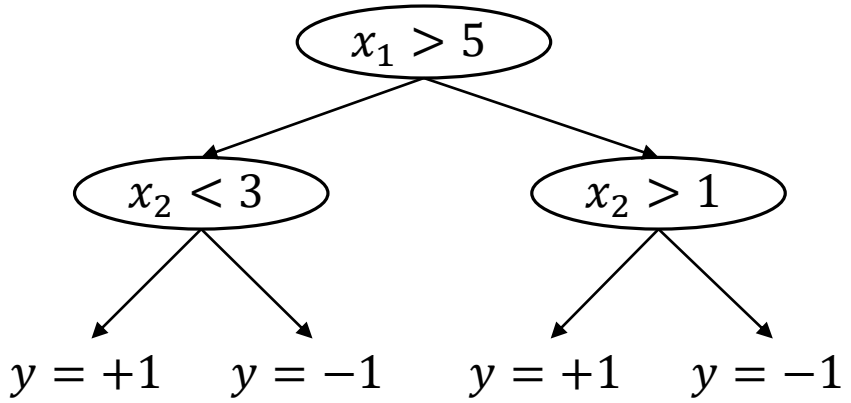
Model ensemble:
Combining different models to improve performance

- One model cannot fit all
- Combine different predictors to improve performance
- Commonly used technique in predictive modeling competitions (e.g. Kaggle)



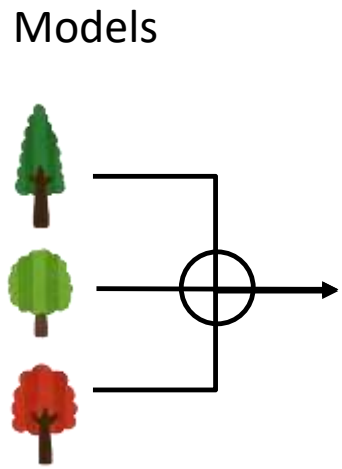
Decision trees:
An off-the-shelf non-linear predictor

- Decision tree:
  - Hierarchically divides input space
- Pros:
  - Relatively fast
  - High interpretability
- Cons:
  - Instable (like other non-linear models)
    - Sensitive to noise and hyper-parameters
- Variants: regression trees, piecewise linear prediction, …



Two ways of ensemble: Horizontal ensemble and vertical ensemble

- Horizontal ensemble
  - Construct a set of models in parallel
  - Integrate their outputs to make final predictions
  - E.g. bagging, boosting, …



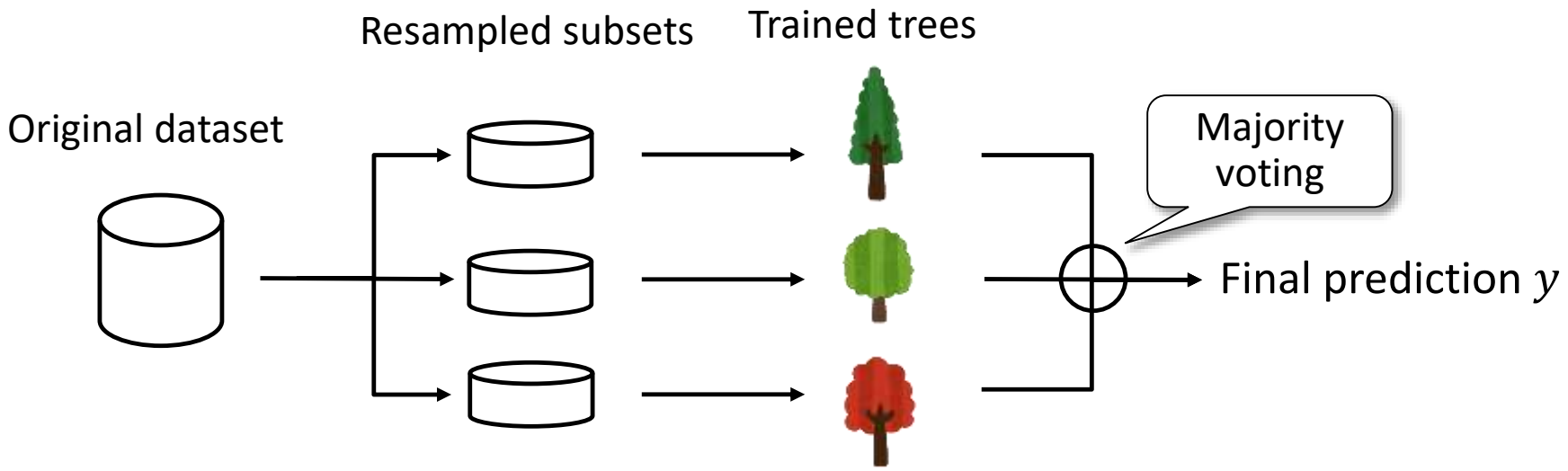
- Vertical ensemble
  - Makes a cascade of models
  - Outputs of ℓ-th level models are use as inputs to $\ell + 1$-th level models
  - E.g. stacking, “deep” neural networks, …

Parallel ensemble methods:
Bagging and boosting

1. Bagging:
   - Simple ensemble method based on data resampling
   - Random forest is often “the first choice”
2. Boosting:
   - Adaptive version of bagging
   - XGBoost is the regular winner in competitions

Bagging:
Majority voting with different predictors

- Decision trees are sensitive to data change
- Bootstrapping: Train a classifier using randomly resampled subset of the original dataset
- Majority voting to aggregate predictions
- Stable, but lost interpretability



Random forest:
Also uses feature resampling for diversity

- Resample not only the data but also features
- Example: 5 samples {1, 2, 3, 4, 5}, 4 features {A, B, C, D}


1st tree trained with 2 features {A, B} on subsamples {1, 3, 5}


2nd tree trained with 4 features {A, B, C, D} on subsamples {1, 2, 3}

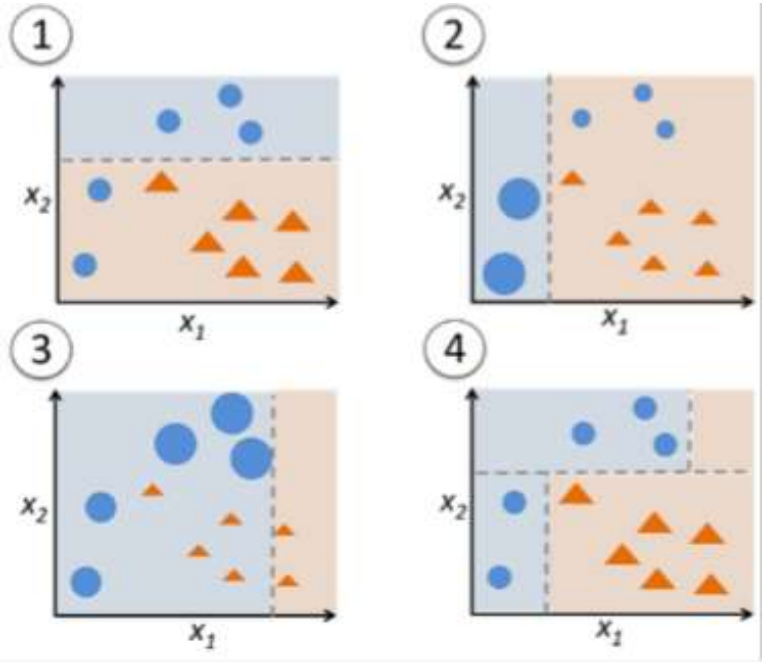

3rd tree trained with 3 features {A, C, D} on subsamples {2, 3, 4, 5}

…

- Off-the-shelf non-linear model: stable and high performance
  - "the first choice"

Boosting:
Adaptive resampling to difficult examples

- In bagging, all models are independently trained using uniformly-random resamples
- In boosting, the next model is trained focusing on the "difficult" data that the current model cannot correctly classify



Very nice introduction of bagging, booting, and random forest

https://sebastianraschka.com/faq/docs/bagging-boosting-rf.html

Gradient boosting:
Incremental addition of new model to current ensemble

- Loss function of the current model $L(F_{t-1}) = \sum_{i=1}^{N} \ell^{(i)}(F_{t-1})$
  - $F_{t-1} = \sum_{\tau=1}^{t-1} \alpha_\tau f_\tau$
    - f_τ: τ–th component model
    - α_τ: weight
- Loss function of the updated ensemble $L(F_{t-1} + \alpha_t f_t)$
- Find α_t and f_t that minimizes L
- Taylor expansion (first order approximation):

Weight of i–th instance

$$L(F_{t-1} + \alpha_t f_t) = \sum_{i=1}^{N} \ell^{(i)}(F_{t-1}) + \sum_{i=1}^{N} \frac{\partial \ell^{(i)}(F_{t-1})}{\partial F_{t-1}} \alpha_t f_t + \cdots$$

XGBoost:
A popular boosting implementation

- Implementation using gradient boosting + decision tree
  - Often appears in top ranked methods in competition
- Training with constraints on the number of decision tree leaves and the total weights
- Parallel implementation

Gradient boosted machines and deep neural nets have dominated recent Kaggle competitions

Competition	Type	Winning ML Library/Algorithm
Liberty Mutual	Regression	**XGBoost**
Caterpillar Tubes	Regression	**Keras** + **XGBoost** + Reg. Forest
Diabetic Retinopathy	Image	SparseConvNet + RF
Avito	CTR	**XGBoost**
Taxi Trajectory 2	Geostats	Classic neural net
Grasp and Lift	EEG	**Keras** + **XGBoost** + other CNN
Otto Group	Classification	Stacked ensemble of 35 models
Facebook IV	Classification	sklearn GBM

https://www.quora.com/What-machine-learning-approaches-have-won-most-Kaggle-competitions

Model stacking: Vertical ensemble method

- Stacking: vertical ensemble method
  - is similar to the multi-layer neural network
    - NN = Stacked linear classification models
  - but can have heterogeneous components
- Outputs of ℓ-th level models are use as inputs to $\ell + 1$-th level models
  - Outputs of 0-th level models $\mathbf{y}_0$ = Original feature vector $\mathbf{x}$
  - Outputs of ℓ-th level models $\mathbf{y}_\ell$
  - Inputs to $\ell + 1$-th level models $\mathbf{x}_{\ell+1} = \begin{pmatrix} \mathbf{x}_\ell \\ \mathbf{y}_\ell \end{pmatrix}$

Difficulty in model stacking: An easy solution is biased

- How can we train staked models?:
- An easy solution:
  1. train a classifier f using the training dataset L
  2. add the prediction values of f as a new feature

  …. seemingly works… but actually does NOT
- Remember the first principle: you cannot make a prediction for the data you used in the training
  - The prediction value to the training data are biased because your model has been trained to reproduce the labels

How to stack:
Use cross-validation to extend features

- Divide a given dataset into K non-overlapping sets
  1. Use $K-1$ of them for training a model
  2. Use the model to add a new feature to the remaining set
  - Doing steps 1&2 for K holdout sets gives the new feature for the whole dataset
- Train the level-2 predictor using the extended dataset
- Finally, the level-1 predictor is (re-)trained using the original whole dataset
  - Because the extended feature for training the level-2 predictor is produced by different level-1 predictors