SSID: zenguest
Password: password

WOMEN WHO CODE MANILA

Artificial Intelligence Study Group

Twitter: @wwcodemanila
FB: fb.com/wwcodemanila

#WWCodeManila
#AI
#StudyGroup

Issa Tingzon

Research Fellow

PCARI

New Member's Introduction

I am <name>

<your current profession>

<why did you join this study group?>

OUR MISSION

Inspiring women to excel in technology careers.

WOMEN WHO CODE MANILA

OUR VISION

A world where women are representative as technical executives, founders, VCs, board members and software engineers.

WOMEN WHO CODE MANILA

STUDY GROUP

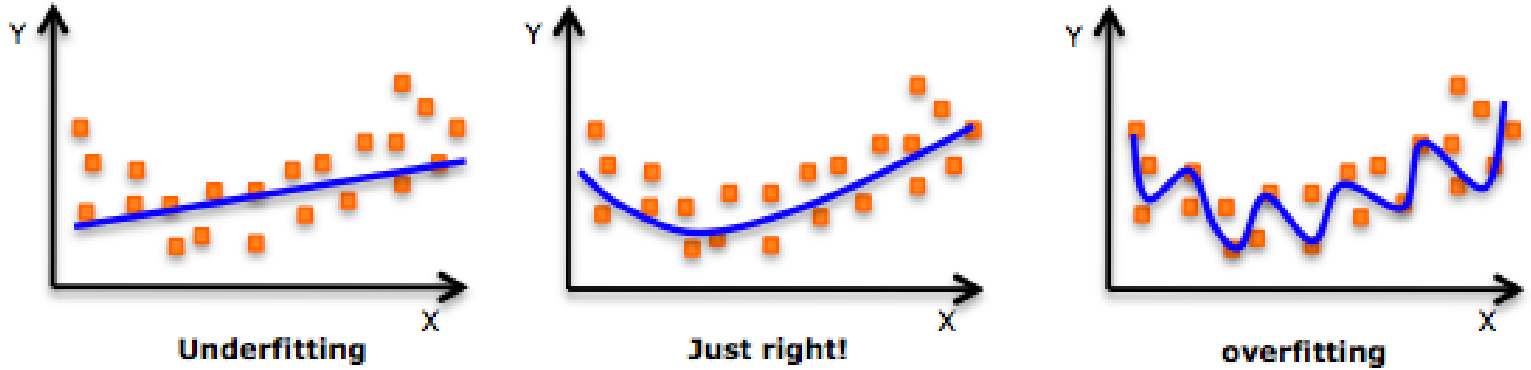
Study groups are events where women can come together and help each other learn and understand a specific programming language, technology, or anything related to coding or engineering.

GUIDELINES

- If you have a question, just **ask**
- If you have an idea, **share it**
- **Make friends** and learn from your study groupmates
- **Do not** recruit or promote your business

REVIEW

- Training set, testing set, validation set
- Overfitting, underfitting

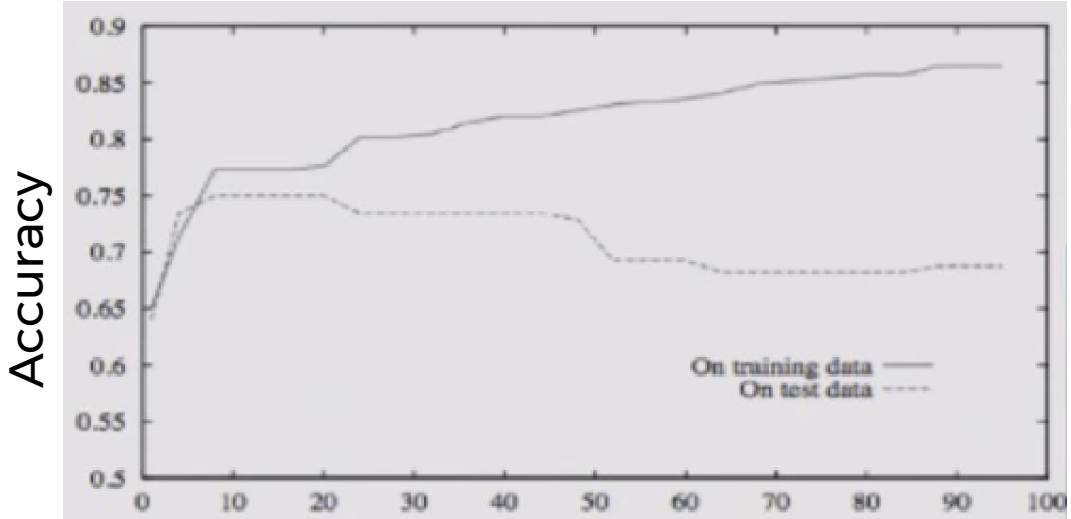


REVIEW

DECISION TREES

Overfitting in Decision Trees

- Recursive algorithm
  - keeps splitting until all subsets are pure
- Can always classify training examples perfectly
- Doesn't work on new data



Avoid Overfitting

- Don’t grow a tree that is too large or has singleton subsets
  - Set maximum tree depth, max no. of features, etc.

Avoid Overfitting

- Don’t grow a tree that is too large or has singleton subsets
    - Set maximum tree depth, max no. of features, etc.
- Remove nodes that are too specific to training data
- **Grow tree, then prune** (based on validation set)
    - For each node:
        - “pretend” remove node + all its children
        - Measure performance on validation set
    - Remove nodes that give you the biggest improvement
    - Repeat until further pruning is harmful

Decision Trees

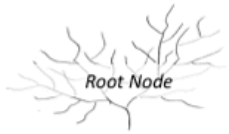
Pros and Cons

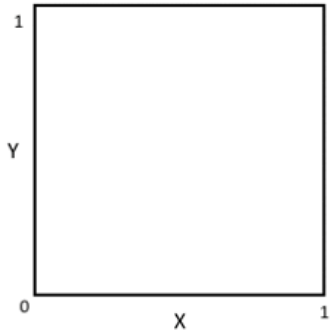
Pros

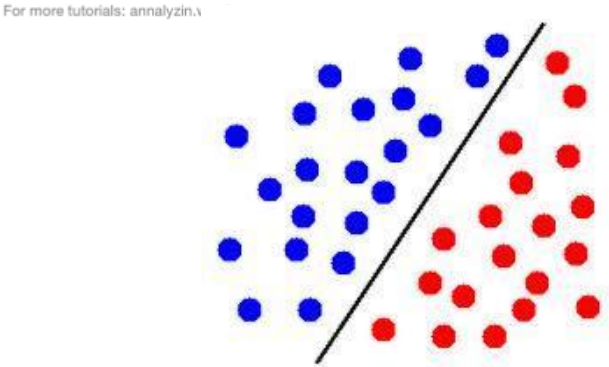
- Interpretable; humans can easily understand
- Easily handles irrelevant attributes
- Easily handles missing data
- Compact after pruning
- Very fast at testing time (O(depth))
- Can handle categorical and continuous data with ease
  - C4.5 is an extension of ID3 that handles continuous data and implements pruning

Cons

- Only Axis Aligned Splits
- Greedy selection of best attribute
  - Does NOT always result in the optimal tree!
- Finding the optimal tree would be too time consuming
  - exponentially many trees to compare
- Prone to Overfitting







TOPIC FOR TODAY

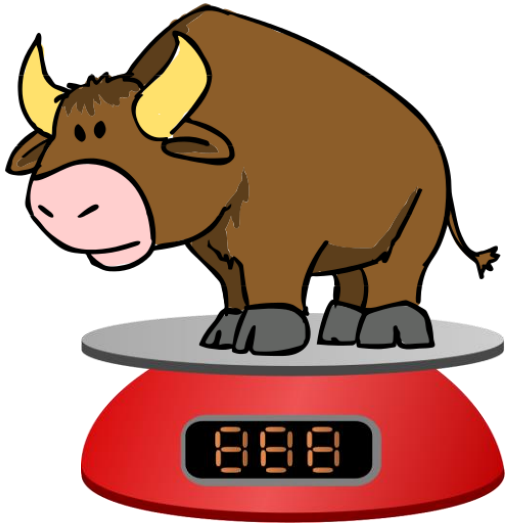
RANDOM FORESTS

Ensemble Methods

- *Wisdom of the Crowd?*
- While visiting a livestock fair, Francis Galton (19th century) was intrigued by a simple weight-guessing contest.

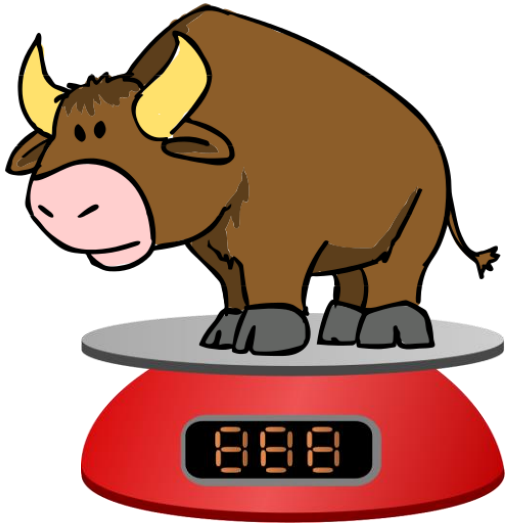
Ensemble Methods

- *Wisdom of the Crowd?*
- While visiting a livestock fair, Francis Galton (19th century) was intrigued by a simple weight-guessing contest.
- The visitors were invited to guess the weight of an ox.



Ensemble Methods

- *Wisdom of the Crowd?*
- While visiting a livestock fair, Francis Galton (19th century) was intrigued by a simple weight-guessing contest.
- The visitors were invited to guess the weight of an ox.
- Hundreds of people participated in this contest but no one individual managed to guess the exact weight: **1,198 lbs**.



Ensemble Methods

- What surprised Galton was the **average of all the guesses came quite close to the exact weight → 1,197 lbs!**

Ensemble Methods

- What surprised Galton was the **average of all the guesses came quite close to the exact weight → 1,197 lbs!**
- **Rationale:** Combining different (poor) classifiers can exploit their individual advantages in order to achieve an overall performance that is better than by using each of them separately.

Ensemble Methods

- What surprised Galton was the **average of all the guesses came quite close to the exact weight → 1,197 lbs!**
- **Rationale:** Combining different (poor) classifiers can exploit their individual advantages in order to achieve an overall performance that is better than by using each of them separately.
- **Poor/Weak Learners** – models that are slightly better than random guessing (e.g. decision trees, perceptrons)

Random Forest

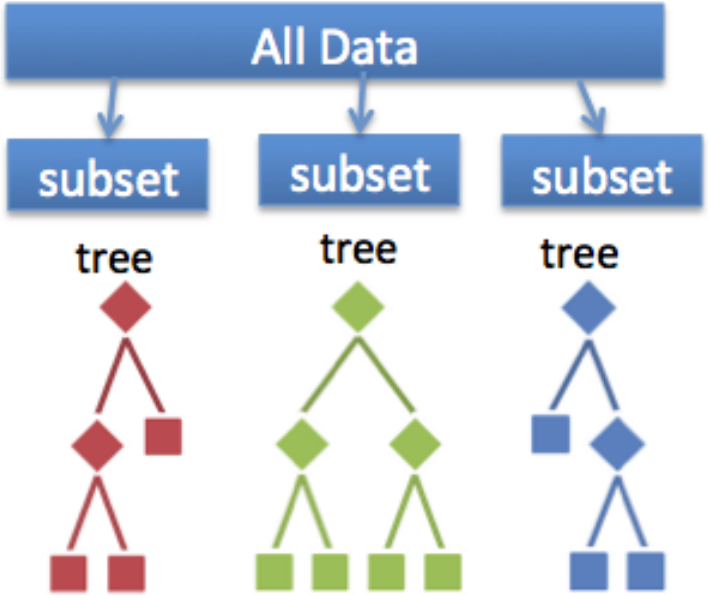
- **Grow K different decision trees**

  1. Pick a random subset S_k $(k \in \{1, \dots, K\})$ of training examples S
  2. Grow a full tree T_k (no pruning)
     - on a subset of d features from a total of D features ($d << D$)
     - compute gain based on S_k instead of S

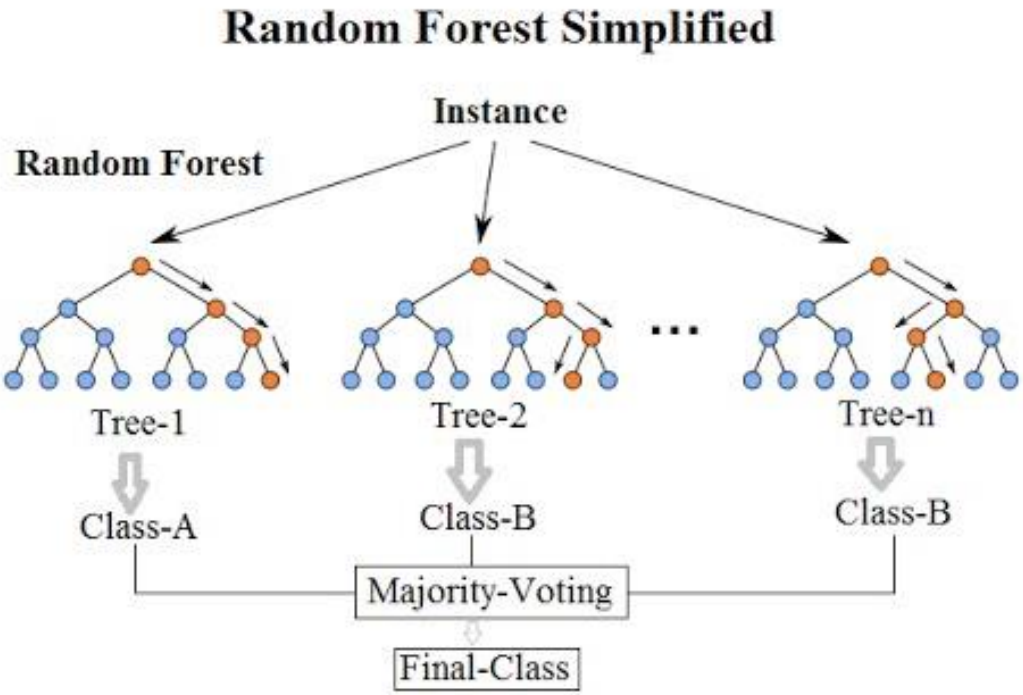
Random Forest

- **Grow K different decision trees**
  1. Pick a random subset S_k ($k \in \{1, \dots, K\}$) of training examples S
  2. Grow a full tree T_k (no pruning)
     - on a subset of d features from a total of D features ($d << D$)
     - compute gain based on S_k instead of S
- **Given a new data point x**
  1. Classify x using each of the trees $T_1, \dots, T_K$
  2. Use majority vote, i.e. which class was predicted most often

Random Forest



Random Forest



Random Forest

- *Simple, but state-of-the-art*
- Image Classification
- Object Detection
- Object Tracking
- Human/ Hand Pose Estimation
- etc.

Why use Random Forest?

- RFs require almost no input preparation
  - RFs can handle binary, categorical, numeric input without need for scaling
- RFs perform implicit feature selection
- RFs are quick to train
- Simplicity
- RFs can be grown in parallel
- Very little hyperparameters to tune
- Can handle multi-class classification seamlessly

Drawbacks of Random Forest

- Model size can be *huge*, and might be slow to evaluate
- More trees → better, typically, but slower!
- Black box? (Not to us anymore, at least!)

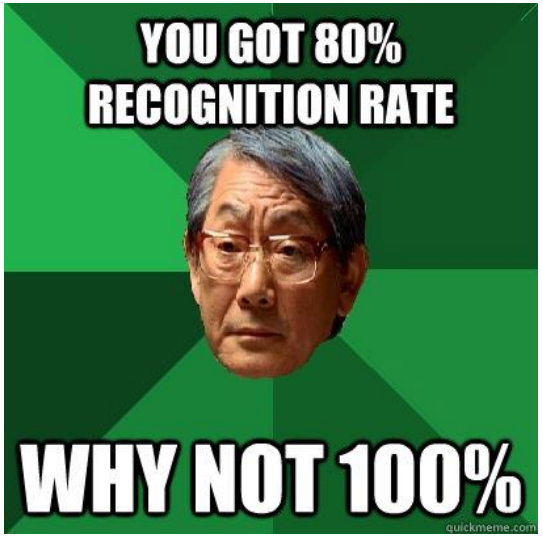
Partner/Group/Individual Presentation

Random Forest for MNIST Dataset

Kaggle Digit Recognizer

- https://www.kaggle.com/c/digit-recognizer

- Join Kaggle's digit recognizer competition
- Submit your predictions using Random Forest Classifier
- Pataasan ng score/rank!



Random Forest in Python

- Use scikit-learn's `RandomForestClassifier()` method
- Focus on optimizing parameters using cross validation
  - Try different values for parameters `n_estimator`, `criterion`, etc.
  - For cross validation, use `cross_val_score` and get the mean of the resulting cross validation scores
- As you increase the number of trees, evaluation gets slower.
  - Parallelization might help to speed things up (hint: tweak `n_job`)
- Feature binarization might also help.

References

Decision Tree Lecture by Victor Lavrenko (Youtube)

Random Forests Video by Siraj Raval

T.I.L.

SHARE IT!
In front!

On Twitter: @wwcodemanila
Or FB: fb.com/wwcodemanila

Don’t forget to tag WWCodeManila so we can retweet or share it.

Feedback Form

https://goo.gl/YzSqcS

Please don't rate the event on meetup.
Not helpful. It is best to just tell your concerns via the feedback form. We are a building a community not a Yelp restaurant.

THANK YOU :)