Introduction to Machine Learning Applications

Spring 2021

Lecture-19

Lydia Manikonda

manikl@rpi.edu

Rensselaer

Agenda for today

- Announcements
- Ensemble Learning
- Bagging
- Boosting
- Random Forest Classifier
- Class exercises

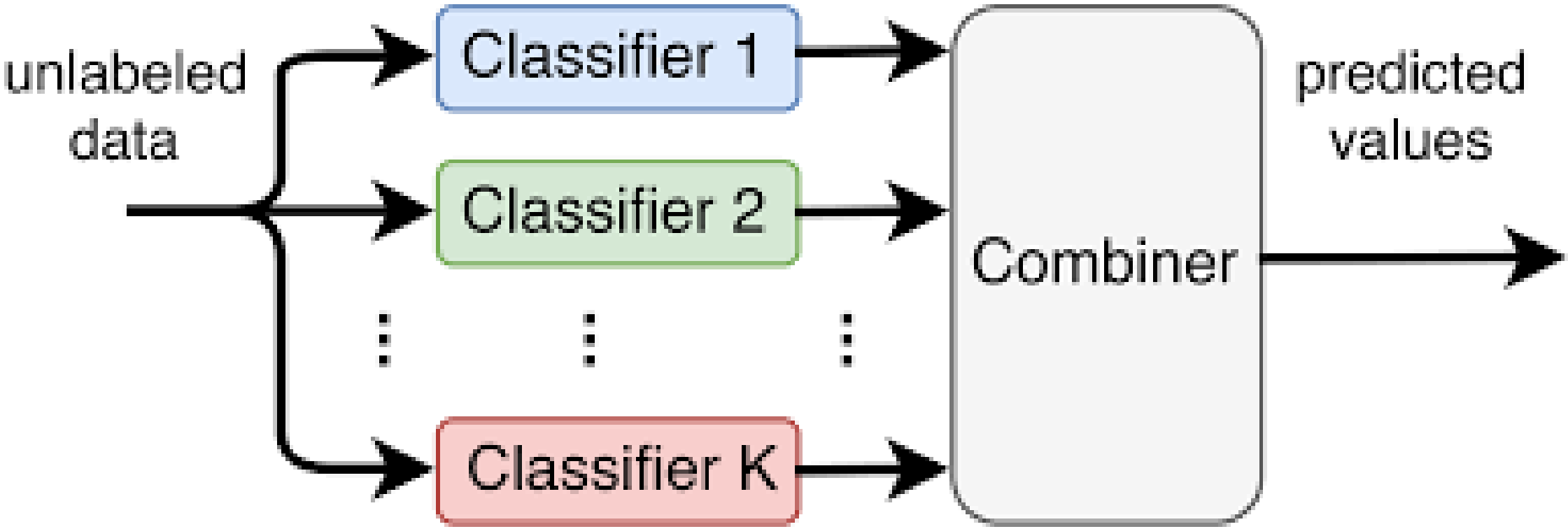
Announcements

- April 15th wellness day – no class
  - HW8 will be due on April 14th
- Project presentations
  - No in-class presentations.
  - Just submit your project report due on 04/08/2021 11:59 pm via LMS
  - Max 2-page pdf
  - which project did you consider?
    - Summarize the dataset;
    - Which 3 solutions did you consider to evaluate?
    - Summarize the features, modeling approach, and performance in a table about these 3 solutions;
    - Report basic statistics on the data and provide atleast 3 visualizations;
    - NO need to submit your code for this deadline.
- We will have only ONE final in-class project presentations

Ensemble Learning

- It is a process by which multiple models are generated and are strategically combined together to predict the final class label.
- One of the main reasons to create this learning is to improve the performance of classification.
- In other words, this would lead to a better predictive performance than a single model.
- One of the main challenge is to obtain these multiple models that make different kinds of errors.

Ensemble Learning



PC: https://www.mathworks.com/

Ensemble Learning

- Two popular ways of learning – bagging and boosting.

- **Bagging** – Also called as bootstrap aggregating – builds various models and learns from them independently in parallel and combines their output.

- **Boosting** – Considers various models that are trained sequentially to generate a robust final output.

Bagging

- Task: Given training data and a testing data point, generate the class label.

- **Step-1**: Create different training data subsets by randomly drawing data points – with replacement – from the entire training dataset.

- **Step-2**: Using each training subset, build a classifier of the same type.

Bagging

- By considering all the classifiers built in the previous step, perform step-3.

- **Step-3**: Obtain the class label for the testing data point and take a simple majority vote.

- The ensemble decision for any given instance is the class chosen by most number of classifiers.

Boosting

- Similar to bagging, boosting also creates an ensemble of classifiers by resampling the training data that are combined by majority voting.
- Unlike bagging, the classifiers are created and utilized sequentially.
- In other words, resampling is done strategically to provide informative data to the consecutive classifier.

- **Step-1**: First build a classifier (**C1**) using a random subset of the given training dataset.

Boosting

- **Step-2**: Given **C1**, now we will build another classifier **C2**, where its training data constitutes of – half of the dataset is correctly classified by **C1** and the other half is wrongly classified by **C1**.

- **Step-3**: Now the third classifier **C3** is trained on the training instances where **C1** and **C2** disagreed on their class predictions.

Random Forests

- Its base is the decision tree
  - which is built by selecting the best feature-value pair to create node and its branches
  - Uses metrics such as Information Gain or Gini value
- However, decision trees have a high risk of overfitting.
- The idea behind random forests is to create several weak models to build a strong classifier.

Random Forests

- Random Forest (strong learner) is built as an ensemble of Decision Trees (weak learners) to perform different tasks such as regression and classification.
- These are trained via the bagging approach where we randomly resample the training data and fit a model (which is a decision tree) to these multiple subsets of training data.
- And we aggregate the results of these multiple trees to make a decision.

Random forests

- This approach introduces more randomness and diversity thus reducing the variance
  - More variance means – fits the data too well, and learns the noise in addition to the inherent patterns in the data
- Regardless of how efficient the approach is, it is always a good idea to perform cross validation.

Python notebook exercises