CECS-550 — Pattern Recognition — Project 1 — DTrees Ensemble

Project 1 — DTrees Ensemble

Introduction

This project is to build an automated Ensemble Decision Tree (DTree) Builder and build a binary pattern recognition Decision Tree Ensemble using the Builder. The project will include a Classified Set of feature vectors. The Training Set will be a percentage of those vectors. Two DTrees will be build for the Ensemble, the first with an initial selection of the vectors, and the second with the same number but of boosted feature vectors. The Ensemble will be a weighted vote of the two DTrees based on their accuracy. Python will be used to implement the project.

Build Details

The project will also include a Classified Set of feature vectors, and a description of the attributes and their possible values.

The Training Set for the first DTree will be a random selection of 60% the Classified Set, and half the remainder (20%) for its Holdout Set. The remaining 20% will be the Validation Set. The Builder will use the standard Shannon-based Entropy and Information Gain mechanisms with a Training Set. Then the constructed DTree will be tested for accuracy on its Holdout Set.

The Training Set for the second DTree will be a random selection with replacement, but the same size as the first. However, it will be selected from the union of the first Training Set and Holdout Set augmented (Boosted) with two extra, each, of the mis-classed Holdout vectors (so that the mis-classed vectors have a 3x probability of being selected) – or any equivalent pseudo-augmentation (eg, by associating explicit higher probabilities). NB, this will likely result in a larger size for the Holdout set. Note, that none of the Validation Set vectors will be involved.

The Builder will use the same mechanisms with a Training Set to construct the second DTree. Then the second DTree will be tested for accuracy on its Holdout Set.

The Builder will then determine the voting weight for each of the DTrees, to complete the Ensemble.

Validation

The Ensemble classifier will be tested for accuracy on the Validation Set. Then each of the two DTrees will also be tested for accuracy on the Validation Set.

Output

The project will output 1) the initial Training Set, Holdout Set, and Validation Set (their IDs) for each of the DTrees – #1 and #2.

Also 2) the first DTree's construction details (Attribute Value Entropies, Attribute Entropies, Attribute Information Gains, and selected Attribute) for each Q-Node, along with the parent (mom) for each child Node, and 3) the Class for each Leaf Node. NB, for the second DTree's construction details, skip the Entropies and just report the rest of the details.

Also 4) the mis-classed Holdout vectors and accuracy (error rate) for both DTrees.

And 5) the Ensemble voting weights (based on the accuracies) for DTrees #1 and #2.

And 6) the Validation Set accuracies for the Ensemble and for DTrees #1 and #2 tested separately.

Teams

We recommend working in a team. Your team may contain from one to four members. (We would prefer large teams.) Pick a name for your team (e.g., "`Groggy`"). You can also include digits after the first letter. [For the next project, team members, and names, can be changed.]

Project Development Reporting

Standup Status Report, twice weekly. The Standup Status Report is due **Monday's** and **Friday's** by noon-ish, until your project is turned in. One report per team, **CC'ing the other team members**. It should contain, team name and the member names. This documents should be delivered **as a PDF file**; and the **filename** should be in the following format: include your course and section number, project number, your team name, the document type (Standup), and the date as YYMMDD:. E.g., “`550-01-p1-Groggy-Standup-201031.pdf`”.

Standup Status Report contents should be, for each team member, a list of the 3 **Standup question short answers**: Q1: what have you **completed** (name sub-task(s)) by the time of this Standup report; Q2: what do you **plan to complete** (name sub-task(s)) by the time of the next Standup report; and Q3: what obstacles if any (1-line description) are **currently blocking you** (for which you've reasonably tried to find the answers by yourself, including asking your team about them – known as “due diligence”). Note, that you can ask questions during office hours to get answers, or email the professor.

Readme File

When your project is complete, you should provide a README.txt text file. Be clear in your instruction on how to build and use the project by providing instructions a novice programmer would understand. If there are any external dependencies for building, the README must also list them and how to find and incorporate them. Usage should include an example invocation. A README would cover the following:

- Class number
- Project number and name
- Team name and members
- Intro (including the algorithm used)
- Contents: Files in the .zip submission
- External Requirements (None?)
- Setup and Installation (if any)
- Sample invocation
- Features (extra)
- Issues (if any) [Bugs, missing stuff]

Academic Rules

Correctly and properly attribute all third party material and references, lest points be taken off.

Submission

All Necessary Files: Your submission must, at a minimum, include a plain ASCII text file called **`README.txt`**, all project documentation files (except those already delivered), all necessary source files to allow the submission to be built and run independently by the instructor. [For this project, no unusual files are expected.] Note, the instructor not use use your IDE or O.S.

Headers: All source code files must include a comment header identifying the author, author’s contact info (please, no phone numbers), and a brief description of the file.

No Binaries: Do not include any IDE-specific files, object files, binary **executables**, or other superfluous files.

Project Folder: Place your submission files in a **folder named** like your Standup report files: `550-01-p1-Groggy`.

Project Zip File: Then zip up this folder. Name the .zip file the **same as the folder name**, like `550-01-p1-Groggy.zip`. Turn in by 11pm on the due date (as specified in the bulletin-board post) by **submitted via emailed zip file(s) (preferred)**, or via accessible cloud (eg, Github, Gdrive, Dropbox) with emailing the accessible cloud link/URL. See the Syllabus for the correct email address. The email subject title should include **the folder name**, like `550-01-p1-Groggy`.

Email Body: Please include your team members' names at the end of the email. (Not the IDs.)

Project Problems: If there is a defect with how your project runs, don't put it in the email body – put it in the README.txt file, under an “ISSUES” section.

Grading

- 80% for 'compiling' and executing with no errors or warnings
- 10% for clean and well-documented code (Clean Rule)
- 10% for a clean and reasonable documentation files, as indicated