CS412 - CS512
Machine Learning
Fall 2017

COURSE PROJECT

(Underlined text is added/modified)

For the course project, you have a choice of implementing a classification or regression engine for one of the given problems.

The aim is to get you handle a real problem, sometimes not knowing what kind of results can be achieved (in general or with these features etc).

Project groups can be 3-4 people per group for undergraduates and 2-3 people per group for graduates, to encourage better and collaborative work and ease grading/presentations (please, no exceptions!). Students in a group may be graded individually to some degree, based on their abilities to answer questions about the project. Undergraduate should be in an UG only group and grads should be in grads only group. If you have not found someone, please write to the Discussion group and then meet other such people.

Please decide on your project and group members by Nov. 23 and I will collect your answers in class on a sheet (preferred) or by November 24 on email.

The projects are due on SuCourse on Dec. 18 evening and selected projects (best 3 groups in each complete project, according to results and reports before Dec. 20) will be presented on Dec. 21, on the last day of class. Late submissions will be accepted **briefly** (few hours just to accommodate that "system and my time were different problems") and with penalty (5 pts off). Place and time of presentations will be arranged.

Please read the Details at the end of the document and use the discussion boards on SuCourse If you have additional questions (e.g. can I use…?).

PROBLEM 1–FRUIT IMAGE RECOGNITION (100PTS)

Task: Given features of a fruit image, you will be asked to classify it into one of the 755 classes.

Data: The data for this project consists of two separate CVS files as train and test sets, for subcategory of fruit pictures. Each row in the files corresponds to the features of one sample (one plant image). Pre-computed attributes are extracted corresponding to each image, using the last convolutional layer of our deep convolutional neural network. So, they are quite powerful features. These high-level features have a dimension of 1024, placed column wise (one image/sample in one row). In the labeled data (train set), a final column indicates the labels for the species; while this is missing in the test set.

You are expected to give the labels column only (column C below) for the test data (as a csv or xls file) without adding extra rows or changing the order rows, **so that we can paste your results as a new column into the existing test file with ground truth labels and calculate your error**.

Column	A	B	C
	s1	attributes	estimated label
	s2	attributes	estimated label

We have added some extra features to this project, so that in addition to the CNN features, you also have the image and other metadata (who took the picture and coordinates and dates…). These features are exactly what is given in PlantClef competition, so you will also have a chance/challenge of deciding what features to include or further extract). The date in particular may be useful for fruit classification, but some other attributes maybe irrelevant and potentially harmful if used. You are encouraged to use validation for measuring your model's performance. The data files will be updated tomorrow under the Project folder.

To sum up each image is associated with the following meta-data:
- MediaId = ImageID (to link to images in the folder)
- Species the species names (containing 3 parts: the Genus name, the Species name, the author(s) who discovered or revised the name of the species): (species determines the label but the classID is what we are interested in as the label.
- Genus: the name of the Genus, one level above the Species in the taxonomical hierarchy
- Family: the name of the Family, two levels above the Species in the taxonomical hierarchy

Species/genus/family will not be available for the test data, but they are given here just in case, lets say if you were to collect more images for the class: but **I DO NOT expect that for full credit.**

- Date: (if available) the date when the plant was observed,
- Locality: (if available) locality name, most of the time a town
- Latitude & Longitude: (if available) the GPS coordinates of the observation in the EXIF metadata, or, if no GPS information were found in the EXIF, the GPS coordinates of the locality where the plant was observed
- Author: name of the author of the picture,
- ClassId: the class number ID that must be used as ground-truth. It is a numerical taxonomical number used by Tela Botanica

Fig. 1. Three samples of the fruit images.

This problem is now areal life problem from a competition.

PROBLEM 2 – SENTIMENT ANALYSIS - TURKISH TWEETS (100+UPTO 10PTS BONUS)

Task: Given a tweet about banks written in Turkish, you will be asked to classify its polarity (how positive a review is it) as a score between -1 (very negative) to +1 (very positive); so this is a regression problem.

E.g. "alt tarafi ATM karti istiyoruz. 5 aydir ne gelen ne giden. Para cekmek icin subeye gitmek zorundamiyim???" -> -0.8 (quite negative comment)

"sorun çözüldü, tesekkürler!"-> +1 (fully positive comment)

Data: The data given to you consists of a single Excel file with two sheets for **train and validation sets**, with 21 pre-computed attributes about significant tweet properties in order to simplify the NLP (natural language processing) tasks for you. But you should consider extracting more features from the input tweet (e.g. occurrence of certain important words) as the precomputed attributes will not be sufficient to do a good job.

The train-validation set split is done for convenience, but you can use the available data as you wish. **In order to prevent manual labelling of the test data, the test data will be provided on the date of the presentations and you will be expected to give your output as given in the Details section.**

E.g.

F1	F2	F3	F21	SCORE
-0.0639	2	1	...	0.7
0.0	4	0	...	-0.5
0.02132	7	3	...	-0.3

The features are commonly used sentiment indicators features such as:

- # of positive smileys,
- # of negative smileys,
- # of swear words,
- # of exclamation marks,
- the occurrence of repeated characters (e.g. çoooook)
- the occurrence of all-uppercase words (e.g. BERBAT) etc.
- ...

You can find a list of most commonly occurring 500 words in these tweets (frequent_words.xlsx), however as there are shorthand notations and Turkish character issues (u used instead of ü etc), the list may not be very useful.

If you want to take an approach based on the frequencies of word occurrences (e.g. Naive Bayes, tf-idf), you can build your own dictionary (or use this set) and **compute frequencies or occurrence of those words in the corpus**.

Submission: You should give the estimated scores for the test data in the order as in the given data (in a CSV or Xls file). That way, we can compute your unbiased generalization performance at the end by simply copying and pasting your results column onto our labelled test Excel file.

This project is also about an important problem, but the available attributes are not very informative and finding informative features may be difficult or take more NLP steps. **For that reason, there will be unto 10pts bonus for extra work like collecting more data or doing significant feature extraction steps.**

PROBLEM 3 – KAGGLE SPEECH RECOGNITION COMPETITION (100+UPTO10PTS)

TensorFlow recently released the Speech Commands Datasets. It includes 65,000 one-second long utterances of **30 short words**, by thousands of different people.

In this competition, you're challenged to use the Speech Commands Dataset to **build an algorithm that understands simple spoken commands**. By improving the recognition accuracy of open-sourced voice interface tools, we can improve product effectiveness and their accessibility.

….

The details are given in https://www.kaggle.com/c/tensorflow-speech-recognition-challenge

Excellent problem that you can relate to and the competition is well-run.

PROBLEM 4 – KAGGLE ICEBER OR SHIP RECOGNITION COMPETITION (100+UPTO10PTS)

In this competition, you're challenged to build an algorithm that **automatically identifies if a remotely sensed target is a ship or iceberg**.

NOTE: The images are not visual data. All the images are 75x75 images with two bands. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle.
…

The details are given in https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/leaderboard

Well-run competition, but you may have a hard time in case you cant relate to the attributes.

PROBLEM 5 – KAGGLE HOUSE PRICE ESTIMATION COMPETITION (100PTS)

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to **predict the final price of each home**.

The details are given in **HTTPS://WWW.KAGGLE.COM/C/HOUSE-PRICES-ADVANCED-REGRESSION-TECHNIQUES**

I gave this problem as a very tangible/relatable problem, but it is a problem where you can overfit. That's why they keep a private leaderboard tested with sequestered (kept away) 50% of the test data. Consequently, your scores can also be overly optimistic. Therefore, for this particular problem, I will pay less attention on the actual scores obtained (public leaderboard scores).

GENERAL KAGGLE RULES

- **You must abide by competition rules.**
- **You must also prefix your team name with SU- followed by any name you pick and forward your team acknowledgement email for us to verify your team.**
- **You will instantly see your scores on submissions sent to Kaggle (some limitations like number of submissions per day). In order to prevent cheating, organizers take some precautions like testing on only a percentage of the test data or adding many artificial data among test samples to prevent hand labelling etc.**
- **Bonus is given in problems where features are not provided and are not readily available; depending on the goodness and effective use of such features.**

DETAILS

Software: You can use any available software (Weka, R, Matlab…), but note rules about using outside data/pretrained systems.

Classifier: You can try a base approach with a few variations (data normalization; dimensionality reduction, feature selection, classifier types or combination…).

Output: Report + Excel/CSV file with test labels without changing anything else (row order, adding rows etc). **Each group should submit their 2 best system's output (two per group) as they decide**. One system has to be trained with only supplied data and clearly marked as such, while a second system can use additional data or pretrained but publicly available systems.

Data: Training and validation data sets can be used as you wish; they are provided separately for convenience in case you want to split. Validation is typically used for parameter optimization or deciding when to stop training or how much to prune a tree, or hyper-parameters of the algorithms (how big the network shd. be etc.) But if you want you could add validation to train and use cross-validation for tuning and model selection. Your system's performance on the unlabelled test set is what matters.

All the data files will be put under Project/ in course website, but more test data may be used at the end for actual testing (for the first two projects).

Grading: Based on how you approached the problem (30%), your report (%30) and your accuracy on the test set (40%) - your approach and results will be correlated of course. The accuracy grade will be given based on your accuracy compared to best known results (other participants in this class, competition or state-of-the-art). Except for Kaggle problems, you will not know the expected accuracies.

While we will be interested in seeing your cross-validation or validation set results, your accuracy will be measured on test set results.

Report: You should write a report explaining in **detail** all aspects of your classifier: you should talk about all your choices (features, preprocessing if any, classifier type and model architecture/ parameters in detail, error measure, training algorithm, 10-fold cross-validation and/or validation set accuracies… also any comments about the working condition, assumptions...) You are expected to write a **3-5 page report** (written by you, w/o counting classifier outputs etc).

Submission: One zipped file named with firstnames_of_groupmembers.zip including

1. **your report;**
2. **all codes;**
3. **test data labels/score file (csv,Excel). Your test labels should be given without changing the order of the entries or adding rows (we will copy/paste your label column into our own excel).**

Presentation: The best 3 groups on each project will be selected on Dec 19, to present their work in a 5 min/presentation per group on the last day of classes (8-10 slides).

Note: You should spend about **2-3 homeworks' time** per person for the project. So, in particular, do not concentrate on finding better features from an image processing or text analysis perspective; rather concentrate on machine learning aspects.