EE 559 **Course Project Assignment** Posted: 4/18/2020
Jenkins v1.0 Due: Fri., 5/8/2020, 2:00 PM

Introduction

The goal of this project is to develop a pattern recognition system that operates on a given real-world dataset. In doing so, you will apply tools and techniques that we have covered in class. You may also confront and solve issues that occur in practice and that have not been covered in class. Discussion sessions and past homework problems will provide you with some tips and pointers for this; also internet searches and piazza discussions will likely be helpful.

Projects can be individual (one student working on the project) **or team** (2 students working together on one project).

You will have significant freedom in designing what you will do for your project. You must cover various topics that are listed below (as "Required Elements"); the methods you use, and degree of depth you go into each topic, are up to you. And, you are encouraged to do more than just the required elements.

Everyone will choose their project topics from the two topic datasets listed below. Collaboration and comparing notes on piazza may be helpful; however each student or team is required to do their own work, coding, and write up their own results.

Topic datasets:

There are 2 topic datasets to choose from:

(i) Wireless Indoor Localization

(ii) Hand Postures (from Motion Capture)

They are described in the separate handout, "Overview of Project Datasets".

Note on both datasets: you are required to use the training and testing datasets that we have posted on D2L. This is so that everyone's performance can be fairly compared. You also have some dataset flexibility or options that are in described in the "Overview of Project Datasets" document.

Which topic datasets can you use?

Individual projects may use either topic dataset. If the Wireless Indoor Localization dataset is chosen, then it is good to extend the stated problem in some way to make it a complete project (some sample suggestions are given in the dataset description).

Team projects may use only the Hand Postures dataset, or both datasets.

Teams are expected to complete more work than individual projects.

Computer languages and available code

You may use Matlab, Python, or C/C++. (To use another language, please ask the TA or instructor first.) Python (possibly supplemented with C/C++) is recommended. Matlab and Python are supported.

You may use any toolbox or libraries you prefer. For Matlab users, some common choices include PRTools and LIBSVM. For Python users, scikit-learn, LIBSVM, pandas, and matplotlib are common choices.

Be sure to state in your project report what languages and toolboxes/libraries you used; what you coded up yourself specifically for this class project; and any code from other sources.

Required elements

- **The items below give the minimal set of items you are required to include in your project, for each dataset you report on.** Note that you are welcome and encouraged to do more than the minimal required elements (for example, where you are required to use one method, you are welcome to try more than one method and compare the results). Doing more work will increase your workload score, might increase your interpretation score, and might improve your final system's performance.
- **Consider Preprocessing: use it if appropriate**
  - Tip: you might find it easier to let Matlab (using the "import" button) or Python (using pandas) handle csv parsing.
  - Normalization or standardization. It is generally good practice to consider standardization. It is often beneficial if different features have significantly different ranges of values. Standardization doesn't have to be all features or none; for example, binary variables are typically not standardized.
  - The datasets in this project don't really require any other preprocessing. (Feature extraction is considered separately.)
- **Feature extraction (Hand Postures dataset)**
  - Define and extract a set of features from the given raw data. You can (optionally) start from the suggested 13 features.
- **Feature-space dimensionality adjustment.**
  - Use a method to try reducing and/or expanding the dimensionality, and to choose a good dimensionality.
- **Cross validation**
  - Use for choosing parameter values, comparing different models or classifiers, and/or for dimensionality adjustment.
- **Training and classification.**
  - Try at least 4 different classification techniques that we have covered in class, including Naïve Bayes, SVM, and two others (including at least one more

statistical classifier). Beyond this, feel free to optionally try other methods (either from those we covered in class or other pattern recognition/machine learning methods).

- **Proper dataset (and subset) usage.**
  - Final test set (as given), training set, validation sets, cross validation.
- **Interpretation and analysis.**
  - Explain the reasoning behind your approach. For example, how did you decide whether to do normalization and standardization? And if you did use it, what difference did it make in system performance? Can you explain why?
  - Analyze and interpret intermediate results and final results. Especially, if some results seem surprising or weren't what you expected. Can you explain (or hypothesize reasons for) what you observe?
  - (Optional) If you hypothesize a reason, what could you run to verify or refute the hypothesis? Try running it and see.
  - (Optional) Consider what would be helpful if one were to collect new data, or collect additional data, to make the prediction problem give better results. Suggest this in your report and justify why it could be helpful.
- **Performance evaluation and baseline comparison**
  - Use the measures given below.
  - Compare with a baseline system as given below.
- **Written final report and code**
  - Submit by the deadline.
  - More detail given below.

Evaluation of performance.

- Report on the cross-validation accuracy (mean and standard deviation)
  - For the Hand Postures dataset, the cross-validation accuracy should use the leave-one-user-out method.
- Report the test-set accuracy of your final system (measured on the given D2L test set).
  - For your final-system test-set accuracy, you may use the best parameters found from model selection, and re-train your final system using all the training data to get the final weight vector.
- Give the confusion matrix (at least on the test set).

General Tips

1. Be careful to keep your final test set uncorrupted, by using it only to evaluate performance of your final system(s).

2. If you find the computation time too long using the provided dataset(s), you could try the following: check that you are using the routines and code efficiently, consider using other classifiers, or down-sample the training dataset further to use a smaller N for your most repetitive work. In the latter case, once you have narrowed your options down, you can do some final choices or just your final training using the required training dataset(s) on D2L.
3. If possible, it can be helpful to consider the degrees of freedom, and number of constraints, as discussed in class. However, this is easier to evaluate for some classifiers than for others; and yet for others, such as SVM, it doesn't directly apply.
4. If using Python, consider using a util.py file for your import statements, then import util in each of your code files.

Baselines

1. For your baseline system, use Naïve Bayes.
2. Also, make a note (in your report) of what accuracy a random classifier would achieve (i.e., a classifier that outputs class labels without looking at the inputs; if percent representation of each class is equal, then the random classifier will randomly pick a class label (e.g., role a die if a 6-class problem) and output it). If your trained system doesn't do significantly better, then it hasn't learned anything.

Grading criteria

Please note that your project will not be graded like a homework assignment. There is no set of problems with pre-defined completion points, and for many aspects of your project there is no one correct answer. The project is open-ended, and the work you do is largely up to you. You will be graded on the following aspects:

- Workload (effort in comparison with required elements, additional work beyond the required minimum);
- Difficulty of the problem (Hand Postures is more difficult than Indoor Wireless Localization; but either can be made more difficult by adding to it).
- Approach (soundness and quality of work);
- Performance of final system (as measured by stated performance metrics);
- Analysis (understanding and interpretation);
- Final report write-up (clarity, completeness, conciseness).

In each category above, you will get a score, and the weighted sum of scores will be your total project score.

For team projects, both members of a team will typically get the same score, but in some cases the score can vary among team members (depending on quantity and quality of each team member's effort).

Final report

In your final report you will describe the work that you have done, including the problem statement, your approach, your results, and your interpretation and understanding.

Note that where you describe the work of others, or include information taken from elsewhere, it must be cited and referenced as such; **similarly, any code that is taken from elsewhere must be cited in comments in your code file.** Instructions for citing and referencing will be included with the final report instructions. Plagiarism (copying information from elsewhere without crediting the source) of text, figures, or code will cause substantial penalty.

For team projects, each team will submit one final report; the final report must also describe which tasks were done by which team member.

You will submit one pdf of your (or your team's) report, one pdf of all code, and a zip file with all your code as you ran it.

Please note that your report should include all relevant information that will allow us to understand what you did and evaluate your work. Information that is in your code but not in your report might be missed. The purpose of turning in your code is so that we can check various other things (e.g., random checks for proper dataset usage, checking for plagiarism, verifying what you coded yourself (as stated in the report), and running the code when needed to check your reported results).

More detailed guidelines for the written report will be posted later.