EE 660 Tips for Finding Appropriate Real-World Datasets Fall 2020

Where to look for a dataset

(1) www.kaggle.com
A very good place to find interesting problems (and to compete!). Note that some of the featured competitions could require quite some time to execute - for example, time to understand the problem domain and perform preprocessing. So do some filtering to find one that is appropriate for a class project.
Note that for Kaggle competitions, problems that have extensive postings of approaches and code (typically in "kernels" section), are ineligible for class projects. This is because you could do a substantial amount of your project by copying other people's work and code, which subverts the intent of the class project. Currently open Kaggle competitions will typically have less posted online about approaches and code.

(2) www.topcoder.com/challenges/data/
Topcoder data science challenges can be even more difficult than Kaggle competitions.

(3) https://archive.ics.uci.edu/ml/index.html
The UCI repository is a well-known place to look for datasets. Many of the datasets, however, can be too easy for a Type-1 class project. One way to see if a dataset is too trivial would be to read the relevant papers listed under dataset description. On the other hand, for some Type-2 projects, a relatively easy dataset might be appropriate.

(4) https://catalog.data.gov/dataset
A wealth of real datasets, covering almost anything the government would collect (weather, demographics, financial/economics data, environmental data, etc.). However, they aren't typically packaged as machine learning problems, so you may need to devise your own machine learning problem from a given dataset. Also, consider how much preprocessing or data formatting might be required, and how many of you will be working on the project.

(5) Search on a problem that interests you
You can even start with something that haunts you in everyday life (e.g. how long can a pizza keep crispy after taking out of oven. Someone might have already created a dataset for that). There are many datasets available on the internet; you just need to be inquisitive and willing to look in order to find some. In this mode, do also consider its data format and amount of conversion that you would need for it to be useable.

Tips continue on subsequent pages…

Terminology and Issues to Consider

Note: This list of tips may seem a bit lengthy; but once you start finding candidate datasets of particular interest to you, this list of tips can help you down select from them. It can also help you start your project in a good direction, saving yourself some missteps.

(1) Number of data points and likely dimensionality (number of features)

You can vary the dimensionality (up or down), but in any case will need enough data points for at least a training and test set, and likely for validation sets also. You can check the generalization bound (at least for a test set with M=1), keeping in mind that it is an upper bound, and also try the rule of thumb of number of data points in a training set being at least 3-10 times the number of degrees of freedom.

(2) Label (Output) Types

Normally numerical output can be solved as a regression problem. Binary Categorical output can be solved as a binary (2-class) classification or logistic regression problem. Multi-Class Categorical output can be solved as a multiclass classification or logistic regression problem. In EE 660 for classification and logistic regression we have mostly considered binary problems. Techniques for multi-class classification were covered in EE 559; and techniques for multi-class logistic regression are described in Murphy 8.3.7. Any of these problem types can make a good EE 660 project.

(3) Unbalanced Data

If different classes have significantly different number of samples, the dataset is considered unbalanced. This common situation will take some extra steps in handling the data, and may indicate the need for alternate performance metrics (like f1 score, Area Under Curve, weighted MSE, etc.).

(4) Missing Data

Some datasets have missing data. Simple options for dealing with missing data include: (i) remove data points that have any missing feature values; or (ii) remove any features that have missing values. More complex options center around filling in missing data, using a method such as Expectation Maximization (EM), kNN, or weighted kNN. EM will be covered towards the end of the semester, and in a different context, so if you want to *fill in* missing data during your project, best to use a method you already know, or learn one on your own.

(5) Categorical (or symbolic) Features

Will need to be re-cast into another (numeric) form, which may increase the dimensionality significantly. This was covered in EE 559, and can be reviewed in discussion in EE 660.

(6) Feature extraction

Depending on your dataset, you might benefit from performing some feature extraction. In some problem domains, this can involve a lot of time and effort; in others, it is simpler. Doing some feature extraction is fine, even helpful, but the main emphasis in your project should be on machine learning techniques.

(7) Test set

It is strongly recommend to first separate out a final test set before trying anything on the data, in order to ensure you will have an unseen and unused set for assessing the final performance. The test set should be randomly sampled from your data; for a classification problem, it should be randomly sampled from each class, with the percent representation of each class in the test set essentially the same as the percent representation of each class in the training set (stratified sampling).

(8) Pre-training set

You might also consider separating out a "pre-training" set from the data, and use only this set for any preliminary trials on your data.

(9) Predictive power of inputs

Is the information needed in the input variables (features) likely sufficient to be able to make good predictions or classifications?
Are there features that are too predictive or would not be available for use of a final system (so that using them would amount to "cheating")? if so, consider removing these before using the dataset.

(10) Check the literature to see what others have done. This can help answer the question of predictive power in the previous point. Also, if their solutions are very complicated, it may be too much for a class project. Or, if most of the work of others is very domain specific (e.g., a lot of preprocessing and feature extraction that you would also have to do), then the required emphasis may not be appropriate.