Project Part 1: Dataset
CSE 446: Machine Learning

University of Washington

Deadline: October 17, 2017

September 28, 2017: updates in red.

For each part of the project, your team will be evaluated as a whole; you will share a grade. The first task for your team to complete is to create a dataset; this task will account for 10% of the final grade of each team member.

There is considerable flexibility in creating your dataset, and we encourage you to be creative. Note that your dataset must instantiate either binary classification (output values of $\{0, 1\}$) or regression (finite, floating point values). You should aim for a dataset where the machine learning problem is challenging, but also arguably possible. That is, you should be able to articulate why you believe y is predictable from x, and why a machine learning algorithm should be able to learn to generalize from a sample of (x, y) pairs to make such predictions for unseen x examples. Further, you should try to create a dataset where the task is worthwhile, or where solving this task would be a step toward solving a larger, harder problem that someone in the world really faces. Finally, you should think through the ethical implications of your dataset, and make defensible choices as best you can. For example, do not collect data that violates anyone's privacy. Ethics in machine learning is a very hot and widely debated topic.[1] We ask that you keep ethical considerations in mind now, and critically evaluate your classmates' datasets later.

Although it is not part of your grade for the project, we will hold a competition among datasets. A small number of datasets will be selected for use in future assignments and the final evaluation of all teams' projects. The more seriously you take this part of the project, the more likely your dataset is to be one of the selected ones.

Here are a few ideas that might help get you started thinking about an interesting dataset.

- If you're interested in text data, Wikipedia offers a tremendous range of possibilites. If x is a Wikipedia document, y might be the number of revisions it has received, or the number of people who have revised it, or some page view statistics, or the number of external links to the page, or whether the page is about a person, or whether the page is in a particular language.
- If you're interested in images, you could construct a collection of images x where the labels y indicate whether the image came from a news source, or the year the image was created, or whether a particular artist created the image. Here is a set of open image collections that might inspire you: https://blogs.ntu.edu.sg/openimagecollections/browse/#collections

[1] Here's a short article that's worth a read, to get your wheels turning: https://medium.com/@mrtz/how-big-data-is-unfair-9aa544d739de

- If you're interested in public policy, take a look at https://www.data.gov/ for datasets produced by the US government and https://data.seattle.gov/ for datasets produced by the Seattle government. There are likely many more of these.
- If you're interested in health, take a look at https://www.cdc.gov/nchs/data_access/ftp_data.htm for datasets produced by the Centers for Disease Control and Prevention. As above, you may be able to find other sources.
- It's perfectly reasonable to create output labels yourself. If you do this, you should take steps to ensure quality. E.g., if two of your team members independently annotate in input, what is their rate of agreement, and how does this compare to agreement-by-chance? (You are encouraged to look up the "Cohen's kappa" and "Krippendorff's alpha" statistics to understand this matter more deeply, and use them to check the quality of your data.)

In completing this part of the project, we want you to find a happy medium between extremes.

Too easy. We realize that there are many benchmark datasets available for machine learning exploration.[2] We would prefer that you not derive your dataset from such datasets, for several reasons. First, many publications describe solutions on these datasets that perform quite well, benefitting from many years of trial and error by the research community. It will be very difficult for anyone in the class to outperform those published models, and so the incentive to merely replicate known solutions will be quite high. This goes against the exploratory spirit of the class; we would rather have you work on datasets where no one knows yet what the best solutions will be. Second, part of understanding machine learning is understanding what problems it can help solve; designing a dataset is a way to get you thinking about this before you get excited about solutions. That said, if you believe you have an idea that meets the criteria above and involves a non-trivial twist on an existing benchmark dataset, please discuss it with the instructors.

Too hard. At the other extreme, you may have a creative idea for a machine learning task for which the data are just too difficult to collect. As disappointing as that is, we only have a ten-week quarter to teach you the basics of machine learning. We don't want all of your energy to go into a complicated data cleanup effort. Save this idea for a future project, once you have mastered machine learning!

Too small. In general, the more complicated a function is, the more examples are required to learn it. If there aren't enough examples, then your dataset may be quite challenging. Another way to be too small is for the complexity of your examples to be fairly low; if each individual example is simple, then it may be too easy for your classmates to reach ceiling performance on the dataset. Rather than putting a minimum on the size of your dataset or the complexity of the examples, we encourage you to aim for at least a few hundred training examples and test out some really simple baselines on easily extracted features to make sure the task is somewhat challenging.

Too big. If your training dataset contains a huge number of examples, or if each example is extremely large and unwieldly, then your classmates won't want to work with the data. We only have ten weeks in the quarter, and everyone will need to train models in minutes or hours, not days.

[2] For example, the UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/.

If loading up the training data in a Python program takes more than a few seconds, consider how you might simplify things. (Feel free to ask for advice from the course staff.)

Deliverables. You will upload the following as a single gzipped tarball named `NAME-dataset.tgz`, where `NAME` is the name of your dataset (see constraints below).

- `README.pdf`: a **one-page** pdf report describing your dataset. The report must include:
  1. all team members' names and netids
  2. a name for the dataset (an alphabetical string with no whitespace; use this in the filename of the submission)
  3. a short, readable description of what the dataset contains, why it instantiates an interesting problem appropriate for machine learning (why do you believe it's not too hard, and not too easy?), and the steps you took to ensure that the dataset is of high quality
  4. an explanation of the format of an input instance and, if necessary, pointers to full specifications (e.g., for image formats)
  5. an explanation of how you decided to divide your data into training and test, and why
  6. an explanation of steps you took to ensure that using your dataset is *ethical* (it may be helpful to read section 8.3 in the book to frame your thoughts)

  Your pdf may include a second page, but it may only contain tables and/or figures with their captions; please only use this option for information best conveyed in tables and figures.
- `train-x`: a list of filenames, each included under the `train` directory, that constitute the **training** dataset. Do *not* include the prefix "`train/`" before the filenames in the list.
- `train-y`: a list of the output values for each training instance; each line in this file corresponds to a line in `train-x`. The values will either be all $\{0, 1\}$ (if this is a binary classification dataset) or they will be floating point values (if this is a regression dataset).
- `test-x`: a list of filenames, each included under the `test` directory, that constitute the **test** dataset. Do *not* include the prefix "`test/`" before the filenames in the list.
- `test-y`: a list of the output values for each test instance, aligned to `test-x` in the same way as `train-y` is aligned to `train-x`.
- Subdirectory `train/`, which contains all training data inputs, one file per instance. These are the files listed in `train-x`.
- Subdirectory `test/`, which contains all test data, one file per instance. These are the files listed in `test-x`.