Special Topics lecture: Machine Learning using Python

May 11, 2022

Administrative Notes

Project 3 stuff

Grades -

- There were only 11 labs. The plan is still to drop 3. I'll figure out how to get 100 points out of your top 8

Expect to see the sample final Thursday evening or Friday

“Machine Learning”

- Many times, just a buzzphrase for “I thought of a really clever algorithm”
- A core part of Artificial Intelligence
  - Which is all too often a buzzphrase itself
- The idea behind machine learning is that we can write a program that will process a large amount of data; “learn” something about the general world based on that data; and then be better when processing the next batch of data

Note: almost everything we talk about today will be from https://scikit-learn.org/stable/

Scikit-learn, or sklearn, is a large lobby of Python code that’s useful in running “machine learning” algorithms

Two major types of machine learning:

- Supervised: the programmer/operator feeds data to the program to “train” it.
  - The program is provided with the data and the “right” answer; e.g., this is a “positive review” or this is a “negative review”
  - The program uses various techniques to “learn” what features make something a “positive review” and what features make something a “negative review” (for example)
  - Then the program is “tested” on additional data, but the program is not told the “right answer.” The program is evaluated on how many right answers it gets vs. how many and what type of wrong answers
- Unsupervised: the programmer/operator feeds data to the program; no “right answers” are provided (or maybe even known)
  - The program uses various techniques to group data in various ways, trying to find associations that might not previously been known

Supervised learning algorithms

- Support-vector machines
- Linear regression
- Logistic regression
- Naive Bayes
- Linear discriminant analysis
- Decision trees
- K-nearest neighbor algorithm
- Neural networks (Multilayer perceptron)
- Similarity learning

Supervised learning techniques

- Support-vector machines(SVM) - build a model that maximizes the “distance” between two training groups, to make it clear which group a new data point belongs in
  - Example: recognizing digits https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digits-classification-py
- K-nearest neighbors:
  - decide that a data set belongs in some number, k, of groups.
  - One by one, put each data item on the graph and figure out where it’s “nearest neighbor” - the nearest center of a group - is.
  - Repeat until all data points are placed
  - Commonly used to decide what type, species or genus something belongs to

Supervised learning techniques

- Naive Bayes
  - Used to identify spam e-mails with impressively high accuracy
  - Relies on applying Bayes' rule:
    - $p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$
    - The "C's" are the classes into which you want to divide the data; e.g., "this is spam" and "this is a legitimate email" and maybe "this message is malware."
    - The x's are various "features" associated with the data - e.g, "it comes from a previously-unknown sender" and "it contains executable macros" and "it contains words known to be associated with spam"
      - p(x) is the frequency in the training set of that feature. E.g., we use a training set consisting of 50% spam e-mails or a training set consisting of 1% spam e-mails
    - You can apply as many features as you want to improve accuracy; you multiply the various probabilities together in that case

Unsupervised machine learning techniques

- K-means clustering: break the data set into some pre-determined number, k, of groups.
  - Identify the centroid (center, but multidimensional so it might not be a point) of each cluster
  - Put each data item into the cluster who centroid is closest to the data item
  - You might have to continually shift items from one cluster into another
- Example: recognizing handwritten digits
  - https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html#sphx-glr-auto-examples-cluster-plot-kmeans-digits-py
  -