Apricot: Submodular selection for data summarization in Python

Jacob Schreiber,Jeffrey Bilmes,William Stafford Noble

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

The Task

- Large amounts of data available for multiple tasks
- Select good representative subsets. Why?

The Task

- Large amounts of data available for multiple tasks
- Select good representative subsets. Why?
    - place higher demands on computational resources: requre change in hardware etc
    - performance flattens after some point with increase in data
    - additional data may be noisy/redundant/unrelated to the task

The Task: Selecting good subsets

- Given the ground set Y (the original training set)
- Find a subset $X \subseteq Y$ that is representative of the original Y
- find $X^* = \max_{X \subseteq Y, |X| \leq k} f(X)$
- Use submodular optimization because can be optimized efficiently by extremely simple algorithms: greedy and linear lazy greedy

Task Formulation

$$X^* = \max_{X \subseteq Y, |X| \leq k} f(X) \tag{1}$$

- submodular functions are those that, for any two sets X, Y satisfying $X \subseteq Y$ and any example $v \notin Y$, have the "diminishing returns" property $f(X \bigcup v) - f(X) \leq f(Y \bigcup v) - (Y)$.

Choices: Submodular function

multiple choices available for choice of submodular function f. The two considered here:

- Facility Location: diversity in the original space by measuring the distance from all points to their nearest representative.
- Feature Based : force a diversity of feature values by modeling the saturation of each feature in the growing subset

Optimization for Submodular functions

- A greedy algorithm can find a subset whose objective value is guaranteed to be within a constant factor ($1 - e^{-1} \sim 0.63$) the optimal subset
- Note it is independent of n

Submodular Function: Facility Location

$$f(X) = \sum_{Y} \max_{x \in X} \phi(X, Y) \tag{2}$$

- Requires $\phi(X, Y)$ for all pairs of examples : $O(n^2)$
- memory cost high, but more generally applicable.
- For graph structure learning?

Submodular Function: Feature Based

$$f(X) = \sum_{d=1}^{D} w_d \phi(\sum_{x \in X} x_d) \tag{3}$$

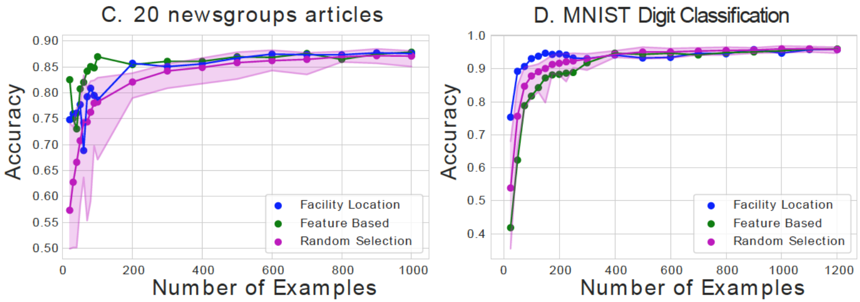
Taken from:

$$f(X) = \sum_{d=1}^{D} w_d \phi_d(\sum_{x \in X} m_d(x_d)) \tag{4}$$

where m_d indicates relevance of feature d for sentence x.

- ϕ is a non decreasing concave (saturating) function
- the degree of diminishing returns and ultimately the measure of redundancy of the information provided by the feature, is controlled by the concave function
- ϕ ca be either log or square root
- do not require the construction of a pairwise graph
- Cost of only $O(nD)$
- But hard to always have features that satisify this property

Experiments



Apricot the package

- Implementation of the two above cases
- Uses Numba : Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.
- pip install apricot-select
- uses the 'lazy' greedy algorithm to avoid doing $O(n^2)$

Future Directions in the paper

- subset selection on feature attributions rather than feature values to identify a model-based summary of the data
- subset selection on the internal representations of samples in a neural network
- discriminative subset selection when given a data set and associated labels
- model-guided subset selection for accelerating transfer learning