Team Crux
DSP P1 Lightning Talk

Zach Jones
Omid Setayeshfar
Vinay Kumar Bingi

Methodology

Methodology - Feature Extraction

- For Naive Bayes, word counts are features
- Approach:
  - combine all training documents of the same class into a mega-document
  - pre-process document contents to get RDD containing (word, class) tuples
    - tokenize
    - remove HTML character refs
    - strip punctuation
    - remove stopwords
  - Combine (word, class) tuples into (word, class_count_vector) tuples
    - e.g. (‘airplane’, ‘GCAT’) becomes (‘airplane’, [0, 0, 1, 0])
  - ReduceByKey to sum up the word counts
    - e.g. (‘airplane’, [3, 7, 53, 1])

Methodology - Classifiers

- Cosine Similarity
  - Basic idea:
    - Compute word frequency vector c_i for each class i
    - For new documents, build frequency vector v
    - Class = argmax_i [CosineSimilarity(c_i , v)]
  - Achieved around 80% accuracy
- Naive Bayes
  - Basic version performed well on the small dataset but extended poorly to large dataset
  - Enhancements:
    - Feature selection based on class-count variance
    - Weight words by term-frequency inverse-class-frequency (TF-ICF)
    - Include monograms and bigrams

Methodology - Software Engineering

- ‘main’ script - driver program
  - collected command-line args and kicked off learner
- ‘Classifier’ interface
  - standardized OO interface for classifiers
  - allowed new classifiers to be introduced easily
- Reusable functions in separate Python modules
  - well organized
  - easy to test
  - easy-ish packaging

Project Management

Project Management

- The Good
  - Extensive use of Github issues, projects, and milestones
  - Ad-hoc team structure worked well for us
  - Project wiki - helpful in providing extensive explanations of methods employed
  - Constant Slack chatter
- The Bad
  - All team members worked out of the same “core” repository
    - Difficult to effectively conduct code reviews
    - No guards against harmful changes or careless mistakes
- The Okay-ish
  - Branching model and commit messages could have been better standardized
  - Code style and documentation is *mostly* consistent

Thanks!