Naïve Bayes (HW1): Tips and Tricks

Shannon Quinn
Git

• Basics
  – http://git-scm.com/docs/gittutorial

• [Advanced] Cheat sheet
  – https://github.com/tiimgreen/github-cheat-sheet
Some useful Hadoop-isms

- Testing and setup
  - You won’t need to worry about setup!
    - On GACRC, I’m setting up the VMs
    - On AWS, Amazon handles it
- But the testing tips near the bottom are excellent
  - Putting files into HDFS
  - Reading existing content/results in HDFS
  - Submitting Hadoop jobs on the command line
- If you use AWS GUI, you won’t need to worry about any of this
Some useful Hadoop-isms

• Counters
  – Initialize in `main() / run()`
  
```java
public static enum MATCH_COUNTER {
  INCOMING_GRAPHHS,
  PRUNING_BY_NCV,
  PRUNING_BY_COUNT,
  PRUNING_BY_ISO,
  ISOMORPHIC
);
```

  – Increment in mapper / reducer
    
```
context.getCounter(MATCH_COUNTER.INCOMING_GRAPHHS).increment(1);
```

  – Read in `main() / run()`
  – Example:
    http://diveintodata.org/2011/03/15/an-example-of-hadoop-mapreduce-counter/
Some useful Hadoop-isms

- **Joins**
  - Join values together that have the same key
  - Map-side
    - Faster and more efficient
    - Harder to implement — requires custom Partitioner and Comparator
    - [http://codingjunkie.net/mapside-joins/](http://codingjunkie.net/mapside-joins/)
  - Reduce-side
    - Easy to implement — shuffle step does the work for you!
    - Less efficient as data is pushed to the network
    - [http://codingjunkie.net/mapreduce-reduce-joins/](http://codingjunkie.net/mapreduce-reduce-joins/)

- **MultipleInputs**
  - Specify a specific mapper class for a specific input path
  - [https://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/MultipleInputs.html](https://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/MultipleInputs.html)
Some useful Hadoop-isms

- `setup()`
  - Optional method override in Mapper / Reducer subclass
  - Executed **before** `map()` / `reduce()`
  - Useful for initializing variables...
Some useful Hadoop-isms

• DistributedCache
  – Read-only cache of information accessible by each node in the cluster
  – Very useful for broadcasting small amounts of read-only information
  – Tricky to implement
WARNING!!!
Hadoop 1.2.1 vs Hadoop 2.2.0
A little MapReduce: NB
Variables

1. \( |V| \)
2. \( |L| \)
3. \( Y=\ast \)
4. \( Y=y \)
5. \( Y=y, W=\ast \)
6. \( Y=y, W=w \)

1. Size of vocabulary (unique words)
2. Size of label space (unique labels)
3. Number of documents
4. Number of documents with label \( y \)
5. Number of words in a document with label \( y \)
6. Number of times \( w \) appears in a document with label \( y \)