Natural Language Processing with Mahout

Casey Stella

Hortonworks

June 26, 2013

Table of Contents

Preliminaries

Mahout → An Overview

NLP with Mahout → An Example

- Ingesting Data
- Topic Models

Questions

Introduction

- I'm a Systems Architect at Hortonworks
- Prior to this, I
  - Did data mining on medical data at Explorys using the Hadoop ecosystem
  - Did signal processing on seismic data at Ion Geophysical using MapReduce
  - Was a graduate student in the Math department at Texas A&M in algorithmic complexity theory
- I'm going to talk about Natural Language Processing in the Hadoop ecosystem.

Apache Mahout

- Apache Mahout is a
  - Library of stand-alone scalable and distributed machine learning algorithms
  - Library of high performance math and primitive collections useful in machine learning
  - Library of primitive distributed statistical and linear algebraic operations useful in machine learning
- The distributed algorithms are able to be run on Hadoop via a set of stand-alone helper utilities as well as providing an API.

Classes of Algorithms Included

- Mahout includes distributed algorithms for
  - Classification
  - Clustering
  - Pattern Matching/Frequent Itemset Mining
  - Recommendation Engines/Collaborative Filtering

Overview of Available Algorithms

Type	**Algorithm**
Linear Algebra	Stochastic Gradient Descent
Linear Algebra	Stochastic Singular Value Decomposition
Classification	Random Forests
Classification	Naïve Bayesian
Classification	Hidden Markov Models
Clustering	Normal and Fuzzy K-Means
Clustering	Expectation Maximization
Clustering	Dirichlet Process Clustering
Clustering	Latent Dirichlet Allocation
Clustering	Spectral Clustering
Clustering	MinHash Clustering
Pattern Mining	Parallel FP Growth

Ingesting a Corpus of Documents

- Mahout provides a number of utilities to allow one to ingest data into Hadoop in the format expected by the ML algorithms
- The basic pattern is
    - Convert the documents to SequenceFiles via the **seqdirectory** command and then create a set of sparse or dense vectors using **seq2sparse**
    - Create sparse vectors of word counts from the sequence files above with the **seq2sparse** command

Converting a Sequence File to a set of Vectors

- Create a sparse set of vectors using the mahout utility **seq2sparse**.
- The **seq2sparse** command allows you to specify:

`-wt`	The weighting method used: tf or tfidf
`--minSupport`	The minimum number of times a term has to occur to exist in the document
`--norm`	An integer $k > 0$ indicating the L_k metric to be used to normalize the vectors.

Ingestion → Demo

DEMO

Topic Models

- Topic modeling is intended to find a set of broad themes or “topics” from a corpus of documents.
- Documents contain multiple topics and, indeed, can be considered a “mixture” of topics.
- Probabalistic topic modeling algorithms attempt to determine the set of topics and mixture of topics per-document in an unsupervised way.
- Consider a collection of newspaper articles, topics may be "sports", "politics", etc.

High Level: Latent Dirichlet Allocation

- Topics are determined by looking at how often words appear together in the same document
- Each document is associated a probability distribution over the set of topics
- Latent Dirichlet Allocation (LDA) is a statistical topic model which learns
  - what the topics are
  - which documents employ said topics and at what distribution

Latent Dirichlet Allocation → Example

- Consider sentences:
  - I like basketball and football.
  - Tim drank gatorade after football practice.
  - John drank gatorade and thinks it tastes terrible.
- For the topics:
  - *Topic 1* → basketball, football
  - *Topic 2* → gatorade, drank
- And sentences:
  - Sentence 1 is 100% Topic 1
  - Sentence 3 is 100% Topic 2
  - Sentence 2 is 50% Topic 1 and 50% Topic 2

LDA in Mahout

- Original implementation followed the original implementation proposed by Blei, Ng and Jordan in 2003
  - The problem, in part, the amount of information sent out of the mappers scaled with the product of the number of terms in the vocabulary and number of topics.
  - On a 1 billion non-zero entry corpus, for 200 topics, original implementation sent 2.5 TB of data from the mappers *per iteration*.
  - Recently (as of 0.6 [MAHOUT-897]) moved to Collapsed Variational Bayes
  - About 15x faster than original implementation

LDA in Mahout

- The **cvb** tool will run the LDA algorithm
- Input: sequence file of SparseVectors of word counts weighted by term frequency
- Output: Topic model
- Parameters:

-dict	The term dictionary
-k	The number of topics
-nt	The number of unique features defined by the input document vectors
-maxIter	The maximum number of iterations.
-mipd	The maximum number of iterations per document
-a	Smoothing for the document topic distribution; should be about $\frac{50}{k}$, with k being the number of topics.
-e	Smoothing for the term topic distribution

LDA in Mahout → Demo

DEMO

Questions

Thanks for your attention! Questions?

- Find me at http://caseystella.com
- Twitter handle: @casey_stella
- Email address: cstella@hortonworks.com