Natural Language Processing with Mahout

Casey Stella

June 26, 2013

Table of Contents

Introduction

- I'm a Systems Architect at Hortonworks
- Prior to this, I've spent my time and had a lot of fun
  - Doing data mining on medical data at Explorys using the Hadoop ecosystem
  - Doing signal processing on seismic data at Ion Geophysical using MapReduce
  - Being 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.
- I'm going to go over Apache Mahout in general and then focus on Topic Models.

Natural Language Processing

Peter Norvig, the Director of Research at Google, said in the Amazon book review for the book "Statistical Natural Language Processing"[1] by Manning and Schuetze

> *If someone told me I had to make a million bucks in one year, and I could only refer to one book to do it, I'd grab a copy of this book and start a web text-processing company.*

[1] http://www.amazon.com/review/R3GSYXSKRU8V17

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.

Selection 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.

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: A Parable

Tim is the owner of an independent record shop and is interested in finding the natural genres of music by considering natural groupings based on what people buy. So, Tim logs the records people buy and who buys them. Tim does not know the genres and he doesn't know the different genres each customer likes.
Tim chooses that he wants to learn K genres and let a set of records define a given genre. He can then assign a label to the genre by eyeballing the records in the genres.

- Words correspond to records
- Documents correspond to people
- Topics correspond to genres and are represented by $\{records\}$.

Latent Dirichlet Allocation: A Parable

Tim starts by making a guess as to why records are bought by certain people. For example, he assumes that customers who buy record A have interest in the same genre and therefore record A must be a representative of that genre. Of course, this assumption is very likely to be incorrect, so he needs to improve in the face of better data.

Latent Dirichlet Allocation: A Parable

He comes up with the following scheme:

- Pick a record and a customer who bought that record.
- Guess why the record was bought by the customer.
- Other records that the customer bought are likely of the same genre. In other words, the more records that are bought by the same customer, the more likely that those records are part of the same genre.
- Make a new guess as to why the customer bought that record, choosing a genre with some probability according to how likely Tim thinks it is.

Latent Dirichlet Allocation: A Parable

Tim goes through each customer purchase over and over again. His guesses keep getting better because he starts to notice patterns (i.e. people who buy the same records are likely interested in the same genres). Eventually he feels like he's refined his model enough and is ready to draw conclusions:

- For each genre, you can count the records assigned to that genre to figure out what records are associated with the genre.
- By looking at the records in the genre, you can give the genre a label.
- For each customer D and genre T, you can compute the proportions of records who were bought by D because they liked genre T. These give you a representation of customer D. For example, you might learn that records bought by Jim consist of 10% "Easy Listening", "20% Rap", and 70% "Country & Western".

LDA in Mahout

- Original implementation followed the original implementation proposed by Blei et al. [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 by Asuncion et al. [2009].
  - About 15x faster than original implementation

LDA in Mahout

- Input data is expected to be a sparse vector
- Ingestion Pipeline from a document directory
  - **seqdirectory** $\rightarrow$ Transforms docs to a sequence file containing a document per entry
  - **seq2sparse** $\rightarrow$ Transforms to a sequence file of sparse vectors as well as a dictionary of terms
  - **rowid** $\rightarrow$ Transforms the set of vectors into a matrix
- LDA expects the matrix as input as well as the dictionary

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:

-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 → Topics

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
bush	sharon	sharon	sharon	sharon
election	administration	disengagement	bush	disengagement
palestinians	bush	administration	administration	us
hand	us	election	settlement	bush
year	endorse	state	plan	west
fence	leader	bush	palestinians	election
roadmap	iraq	hand	disengagement	solution
arafat	year	conflict	solution	hand
have	election	year	election	day
conflict	transfer	us	fence	year

Questions & Bibliography

Thanks for your attention! Questions?

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

BIBLIOGRAPHY

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press. ISBN 978-0-9749039-5-8.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003. ISSN 1532-4435.