Map-Reduce for Machine Learning on Multicore

Chu, et al.
Problem

- The world is going multicore
- New computers - dual core to 12+-core
- Shift to more concurrent programming paradigms and languages
- Erlang, Scala, Occam, Go, among others

Tuesday, August 30, 11
- Paper discusses other languages, but
  1) I dont know any of them
  2) These languages are more up to date and useable.
• Machine Learning frameworks on multicore
• How to parallelize algorithms?
• Highly specialized, non-obvious solutions
• Often applicable to very few algorithms

Example 1: Cargea et al – Restricted to decision trees
Example 2: Jin, Agrawal – Shared Memory Machines only
Goal

• Develop general techniques for parallelizing many machine learning algorithms
• Goal of “throw more cores at it” optimization
• Algorithms must fit the Statistical Query Model

Rather than traditional approach of algorithm specific tweaks
Statistical Query Model

- (Kearns, 1999)

- Permit learning algorithm to access learning problem only through a Statistical Oracle

- Oracle returns estimated expectation of \( f(x,y) \), given over all data instances

- The expectation averaged over the training/test distribution
• All algorithms that computes sufficient statistics can be expressed as a summation over all data points

• This trait lends itself to batch summations over all data points

• Why do we care?

Sufficient statistics –
A statistic is sufficient for a family of probability distributions if the sample from which it is calculated gives no additional information than does the statistic, as to which of those probability distributions is that of the population from which the sample was taken.

\[ \Pr(X=x \mid T(x) = t, \Theta) = \Pr(X=x \mid T(X) = t) \]
conditional probability not reliant on underlying parameter \( \Theta \).
(Source wikipedia article)
Multicoring Algorithms

- Algorithms that sum over data points allows for partitioning
- Chunk data into subgroups
- Process and aggregate results
Example: Linear Least Squares

Fits model of $y = \theta^T x$

Solve $\theta^* = \min_\theta \sum_{i=1}^{m} (\theta^T x_i - y_i)^2$

Letting $X \in \mathbb{R}^{m\times n}$ be the instances

$\vec{y} = [y_1, \ldots, y_m]$ be the target values

Solve $\theta^* = (X^T X)^{-1} X^T \vec{y}$
- To Summation form:
  - Two-phase algorithm
    - Compute sufficient statistics
    - Aggregate statistics and solve
  - Solve to get $\theta^* = A^{-1}b$
Divide and conquer using the multiple cores

What programming architecture can help us do this task?

\[
A = X^T X \\
A = \sum_{i=1}^{m} (\vec{x}_i \vec{x}_i^T) \\
b = X^T \vec{y} \\
b = \sum_{i=1}^{m} (\vec{x}_i \vec{y}_i)
\]
Map-Reduce

- Google inspired software framework
- Specialized for big data on clusters
- Based on functional programming

- **Map**: Give sub-problems to worker nodes
- **Reduce**: Combine all sub-problem results to obtain answer

Tuesday, August 30, 11
- Map and Reduce functions not the same as in the functional programming
- Nodes are cores in a single-machine implementation, and both machines and cores in a cluster setup
If mapper/reducer requires scalar information, they can access from the algorithm (1.1.1.1 and 1.1.3.2)

You can have multiple reducers as well (Load balancing and fault tolerance) ideal in cluster setting but you have increased communication overhead
Example: k-means

- Given a set of data points
  - Compute initial k centroids
  - Cluster data points to these centroids
  - Calculate the new centroid
  - Repeat process till convergence

We will use Euclidean (easy to think about)
Left image: The dataset we want to run k-means on
Right Image: Clustering and calculating the new centroid
Old Centroids – Square
New Centroids – Triangle
Example: k-means

- Mappers
  - Cluster: Compute distance between points and centroids
  - Centroid: Split up data, compute partial sums of vectors
- Reduce:
  - Cluster: Choose the closest centroid and apply it to that data point
  - Centroid: Aggregate partial sums and calculate centroids

Iterative Clustering process...
Eventually settle and have our clusters
Each image here represents a different map job
The map job is responsible for calculating distance between each data point and the centroid specified for the mapper
Each point will have \( k \) different distances for each centroid

Can anyone think of a different way to run this mapping job?
Reduce Step

Reduce step can take the values for each point, and select the min value
Different map jobs for calculating the new centroids
- Calculate the partial sum for each centroid
Reduce jobs aggregate and average these partial sums to form new centroid points
Example: PCA

• Computing covariance matrix $\Sigma$
  \[ \Sigma = \frac{1}{m} \left( \sum_{i=1}^{m} x_i x_i^T \right) - \mu \mu^T \]
  
  • With $\mu = \frac{1}{m} \sum_{i=1}^{m} x_i$

• How would we map this? Reduce?

Ans: Map each sum to one of the cores. How to map it further??
Reduce: perform final calculation (subtract
Results

• Theoretical Time Complexity speedup of $P'$ on $P$ cores
• Experiment using two versions of algorithm
• For dual-core processors at least 1.9 times speedup for parallelized algorithms
• Linear speedup with number of cores
• Viable method for parallelizing machine learning algorithms

P' appx. equals P
Many different datasets used
Original algorithms don't utilize all cpu cycles
Sub 1 slope for speed increase – increasing communication overhead
Frameworks

• Influenced a variety of researchers and developers
  • Drost, Ingersoll - Apache Mahout
  • Ghoting et al. - IBM SystemML
  • Zaharia et al. - Berkeley Spark
Frameworks: Mahout

- Build scalable machine learning libraries
- Four use cases
  - Recommendation
  - Clustering
  - Classification
  - Frequent item set mining

- Developed with Hadoop in mind though
  Recommendation – Given a set of features for a user, find items that a user may like
  Clustering – Given set of data, group them based on some set of features
  Classification – Assign unlabeled data based on learned patterns from trained data
  Frequent item set mining – Given items in a set, find items that are most frequently associated with elements in that item set
• Still new
  • current release 0.5/0.6
  • Many algorithms proposed in paper yet to be implemented
    • Ones implemented scale well
• Compatibility
  • Works with/without Apache Hadoop
  • Works single or multi node setup

Many clustering techniques available, some classification techniques, SVD, Similarity, and Recommenders available, with others either open or awaiting integration.
• (Relatively) Easy to use with Apache Lucene

• Text analysis

• Benefits

• Algorithms available

• Plug in data and play
• Compatibility

• Available to use right now and open source

• Detractions

• Setting up Mahout not always easy

• What you want might not be implemented

• Very young, and code/usability shows it
Discussion

• What are some of the drawbacks to Map-Reduce?

•
http://en.wikipedia.org/wiki/MapReduce
http://en.wikipedia.org/wiki/Naive_Bayes_classifier
http://en.wikipedia.org/wiki/Sufficient_statistic

http://hadoop.apache.org/
http://mahout.apache.org/
http://www.spark-project.org/

K-means images generated in Matlab