MACHINE LEARNING & HADOOP: 3 WAYS TO MAKE YOUR DATA WORK HARDER

By: Albert Hui, Associate Director, EPAM
INTRODUCTION

When I was doing my Master thesis in fuzzy logic, I struggled with writing my owned statistical algorithms as well as processing a large set of data to train my AI (Artificial intelligence) model. These tools I created are widely available. Thanks for the advancement in the software world, the forward-thinking computer scientists and engineers to make machine learning a reality.
WHAT IS MACHINE LEARNING?

SUPERVISED MACHINE LEARNING

Supervised learning is tasked with learning a function from labeled training data in order to predict the value of any valid input. Common examples of supervised learning include classifying e-mail messages as spam, labeling Web pages according to their genre, and recognizing handwriting. Many algorithms are used to create supervised learners, the most common being neural networks, Support Vector Machines (SVMs), and Naive Bayes classifiers.

UNSUPERVISED MACHINE LEARNING

Unsupervised learning is tasked with making sense of data without any examples of what is correct or incorrect. It is most commonly used for clustering similar input into logical groups. It can be used to reduce the number of dimensions in a data set in order to focus on only the most useful attributes, or to detect trends. Common approaches to unsupervised learning include k-Means, hierarchical clustering, and self-organizing maps.

LIST A
Two main types of machine learning

This paper is an introduction to Machine Learning, and focuses on three objectives.
1. Shows you the trends and latest technologies in machine learning.
2. Helps you to start thinking how to better use your data via the machine learning techniques.
3. Demonstrates how to get started with some sample codes and algorithms.

This paper touches on a spectrum of technologies, mostly from the Apache Software Foundation projects, e.g. Apache Hadoop, Mahout, Lucene and Solr.

What is machine learning? First, machine learning is not a novel topic. In fact, it was the original promise from the computer scientist when they first invented computer software programming. The long history shows scientists have not given up the idea of using machines to help improving our lives.

Machine Learning is a field of Artificial Intelligence (AI), and AI textbooks define this field as “the study and design of intelligent agents” where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1956, defines it as “the science and engineering of making intelligent machines, and is to program computers to optimize a performance criterion using example data or past experience”. It is a field of combining efforts of information technology, statistics, biology, linear algebra and psychology etc. The modern scientists divide machine learning into two major types, Supervised and Unsupervised. This paper focuses mainly on supervised machine learning. Machine learning subtly differs from data mining, which focuses on extracting patterns – the unknown properties on the data. Machine learning focuses on prediction, based on known properties learned from the training data. Proudly, machine learning has finally become closer to reality. IBM invented Watson the computer to compete...
human brains in a series of Jeopardy TV shows. The project is called IBM Watson, it is the ultimate machine learning example using natural language processing method to demonstrate the power of machine learning, see link http://www-03.ibm.com/innovation/us/watson/what-is-watson/index.html.

**Structured DATA:** Walmart logs 1M transactions per hour; Boeing logs 640 terabytes from a 4-engine jumbo jet on one Atlantic crossing.

**Unstructured DATA:** Twitter products 90M tweets per day; Facebook creates 30Billions web links, news, blogs, photo; YouTube has 10M+ views, uploads and download per day.

**Social MOVEMENT:** as people are getting richer, they become more literate that fuels the information growth

---

**LIST D**

Big Data Variety and Variability

What are the use cases of machine learning? Believe or not, machine learning has been integrated into our lives for years. Facebook is able to generate tailored advertisement based on how we are using it to connect with friends and status/comments that you left there. Amazon uses machine learning to recommend books and items that you mostly like. This accounts for 30% of sales out of its recommendation engine. Google’s new piracy policy needs you to agree on allowing them to integrate your preference among its offered products, Gmail, Google+, Google Map aiming to learn more about you and to sell more to you. Yahoo’s spam mail detection and intelligent filtering is also based on the clustering mechanism from machine learning. An ordinary traditional grocery British retailer Tesco collects 1.5 billion piece of information every week in order to use machine learning to adjust prices and promotions in real time. Not to mention it is mandatory for all their executives to tweet about their company in a daily basis. There are no shortages of use cases of machine learning; List B provides a few more. The pioneers in the industry have adopted so well using machine learning, now it is our turn as corporate to respond this “big” trend to further grow our businesses.

---

**E-Tailing:** product recommendation engines, cross channel analytics.

**Retailing/consumer:** Merchandizing and market basket analysis, campaign management and optimization, market and consumer segmentation.

**Financial Services:** Next generation of Fraud detection and risk management, credit risk scoring decision making.

**Government:** Fraud detection, catching Bin laden, cybersecurity.

**Web Scale:** clickstream, social graph analysis, ad targeting/forecasting.

**Telecommunication:** customer churn management, Mobile behavior analysis.

---

**LIST B**

More examples of Use cases in machine learning
HOW DO WE USE MACHINE LEARNING TO IMPROVE OUR BUSINESS?

Before answering this question, let us take a look of fundamental question first. Machine learning needs a lot of data, and actually it is a process to turn a lot of data to make them smarter. Like in my master thesis, my research was to optimize machining processes using machine learning. The fuzzy logic model that I developed was designed to process and learn a lot of machining data. The goal was to help people to make better decisions. There are four steps in turning data into smart data:

1. Collect your data
2. Define the problem you are trying to solve
3. Build your model to train using the data
4. Use the model to predict and analyze the results, and modify your model if necessary

Advancement of TECHNOLOGY: Digital devices soar as prices plummet, 4.6 billion mobile phones, GPS Geospatial

Internet: 2 billion people has access to Internet, Social networks, Digital clickstream

Social MOVEMENT: as people getting richer, they become more literate that fuels the information growth

THE FOLLOWING DISCUSSES MORE DETAILS ON EACH STEP.

STEP 1: DATA COLLECTION

In this digital age, data collection is not easy. We simply have too much data to collect and process. This phenomenon is easy to explain. The storage cost has fallen from $14M per TB in 1980 to $70 in 2010. CPU drops to $59 processor in 2009 for an AMD Radeon. Bandwidth skyrockets from 200KB/s in 1980s’ to 4GB/s in 2010. All these factors create the “best” environment for generating a lot of data. Digital universe estimates there will be 1.8 zetabytes (1 zetabyte = 1 billion terabyte) of data in 2015. And by 2020, the world will have 35.2 zetabytes growing by a factor of 44. List C contains a few reasons for data explosion. Data today is not just BIG. It is also very complex. There is 1.8 trillion gigabytes of data created in 2011 and more than 90% is unstructured. It is approximately 500 quadrillion of files stored outside the databases, and the quantity doubles every 2 years. The industry has defined it as “Era of Big Data”, which is defined as datasets that grow so large that they become awkward to work with using relational databases.
Think big
Think outside the box
Think new revenue possibility
Think how to cut cost and reduce risk

LIST E
Think your machine learning problem

The Era of Big Data has posed some challenges to machine learning. Many old technologies are simply not scalable to read and process such big data volume. However, with the advent of Apache Hadoop, this problem has found a solution. Apache Hadoop, inspired by Google’s white paper on Map/Reduced and Google File System (GFS) and Big Table, is an Apache top level project. It is open source and it is designed for large scale of data processing and to deal with structured and unstructured data. It is able to run on commodity hardware and since it is based on Java, it is portable across heterogeneous hardware and OS platform. Why is it best for machine learning? It is because Hadoop is known for embarrassingly scalable as your data grows. It is reportedly claimed that Hadoop Map/Reduce is able to read and aggregate petabyte of data 500X faster than other commercial software. When you have more data to learn from, you can simply add more hardware to the Hadoop cluster to scale. It is merely a fraction of the cost comparing with other commercialized software. Corporations can simply collect their data first, and have an extremely scalable way to analyze them later. With Apache Hadoop, it gives a new realm of possibility for corporations to solve their machine learning problems.

“The sexy job in the next ten years will be statisticians... The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it.”

Hal Varian, Google Chief Economist

LIST F
Mahout provides statistics algorithm capability for machine learning
STEP 2: DEFINE YOUR PROBLEM

With Apache Hadoop, you can contextualize your data by decomposing them in key-value pair. For example, you can quickly learn how many people are talking about apples, oranges regarding fruit or windows and doors regarding renovation. Or people are talking apple and windows regarding Apple’s Ipad. And you can analyze terabyte of clickstream data to form relationship between pages and user's behaviours and buying decision. It can also marry both structured and unstructured data to gain broader insight of customers' buying behaviours, which was before nearly impossible.

STEP 3: BUILD YOUR MODEL TO TRAIN USING THE DATA YOU COLLECTED

Now, it is time to introduce Apache Mahout (means elephant driver in Hindi) to help develop your machine learning models. Apache Mahout is an open source project by the Apache Software Foundation (ASF) with the primary goal of creating scalable machine learning algorithms that are free to use under the Apache license. It uses the Apache Hadoop library to enable Mahout to scale effectively in a Hadoop cluster cloud. Mahout contains four major implementations, they are:
1. Clustering
2. Classification
3. Collaborative filtering (recommendations)
4. Frequent-pattern mining.

The power of Mahout is that it contains a list of well established statistical models. If you type bin/mahout at the command line, you will see the following list:
> canopy: : Canopy clustering
> cleansvd: : Cleanup and verification of SVD output
> clusterdump: : Dump cluster output to text
> dirichlet: : Dirichlet Clustering
> fkmeans: : Fuzzy K-means clustering
> fpg: : Frequent Pattern Growth
> itemsimilarity: : Compute the item-item-similarities for item-based collaborative filtering
> kmeans: : K-means clustering
> lda: : Latent Dirchlet Allocation
> ldatopics: : LDA Print Topics
> lucene.vector: : Generate Vectors from a Lucene index
> matrixmult: : Take the product of two matrices
> meanshift: : Mean Shift clustering
> recommenditembased: : Compute recommendations using item-based collaborative filtering
CLUSTERING

Let’s go a little deeper into each of four implementations to help you get started.

Given large data sets, whether they are text or numeric, it is often useful to group together, or cluster, similar items automatically. For instance, given all of the news for the day from all of the newspapers in the United States, you might want to group all of the articles about the same story together automatically; you can then choose to focus on specific clusters and stories without needing to wade through a lot of unrelated ones. Another example: Given the output from sensors on a machine over time, you could cluster the outputs to determine normal versus problematic operation, because normal operations would all cluster together and abnormal operations would be in outlying clusters. Clustering calculates the similarity between items in the collection, but its only job is to group together similar items. In many implementations of clustering, items in the collection are represented as vectors in an n-dimensional space. Given the vectors in Figure 1, one can calculate the distance between three items using measures such as the Manhattan Distance, Euclidean distance, or cosine similarity. Then, the actual clusters can be calculated by grouping together the items that are close in distance.

Popular approaches include k-Means and hierarchical clustering. Mahout comes with several different clustering approaches.
**Step a:** Convert dataset into a Hadoop Sequence File:

```bash
$ bin/mahout seqdirectory -I mahout-work/reuters-out -o mahout-work/reuters-out-seqdir
    -c UTF-8 -chunk 5
```

In the sequence of files, each record is a `<key, value>` pair

- **Key:** record name or file name or unique identifier
- **Value:** content as UTF-8 encoded string

**Step b:** Writing to the sequence files

**Step c:** Generate Vectors using the sequence files by running run:

```bash
$ bin/mahout seq2sparse
```

**Step d:** Start k-mean Clustering

```bash
$ bin/mahout kmeans -I mahout-work/reuters-out-seqdir-sparse-kmeans/tfidf-vectors/ -c
    mahout-work/reuters-kmeans-clusters -o mahout-work/reuters-kmeans -dm org.apache.
    mahout.distance.CosineDistanceMeasure -cd 0.1 -x 10 -k 20 -ow
```

There are many clustering algorithm available in Mahout. They are Canopy, fuzzy k-Means, Mean shift, Dirichlet process clustering and spectral clustering.

---

**FIGURE 2.**
Clustering: understanding data as vectors

The vector deoted by point (5, 3) is simply Array ([5, 3]) or HashMap ([0 = > 5], [1 = > 3])
The goal of classification (often also called categorization) is to label unseen documents, thus grouping them together. Many classification approaches in machine learning calculate a variety of statistics that associate the features of a document with the specified label, thus creating a model that can be used later to classify unseen documents. For example, a simple approach to classification might keep track of the words associated with a label, as well as the number of times those words are seen for a given label. Then, when a new document is classified, the words in the document are looked up in the model, probabilities are calculated, and the best result is output, usually along with a score indicating the confidence the result is correct. Features for classification might include words, weights for those words (based on frequency, for instance), parts of speech, and so on. Of course, features really can be anything that helps associate a document with a label and can be incorporated into the algorithm.

Mahout has several implementations, Naïve Bayes, Complementary Naïve Bayes, Decision Forests and Logistic Regression. The most popular classification is Naïve Bayes which is based on condition probability of the class given an instance. And:

\[
P(E_1, E_2, \cdots, E_n | H) = \\
= P(E_1 | E_2, \cdots, E_n, C)P(E_2, \cdots, E_n | H) = \\
= P(E_1 | H)P(E_2, \cdots, E_n | H) = \\
= P(E_1 | H)P(E_2 | H) \cdots P(E_n | H)
\]

where Evidence E is the instance and Event H value for the instance.

For example: The following is to play tennis Event H and the weather conditions E are the evidences for the instance.

<table>
<thead>
<tr>
<th>DAY</th>
<th>OUTLOOK</th>
<th>TEMPERATURE</th>
<th>HUMIDITY</th>
<th>WIND</th>
<th>PLAY TENNIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcas:</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Based on the above dataset, we can first train the classifier:

```
$MAHOUT_HOME/bin/mahout tennnisclassifier -i 20news-input/bayes-train-input -o playtennismodel -type bayes -ng 3 -source hdfs
```

After the classifier is built and trained, we can test them using Mahout by entering a new day with its evidences. Use a new day looks like: Outlook = Sunny, Temp = Cool, Humidity = High, Windy = true. Hence, use the model to determine the probability of yes playing tennis vs probability of No playing tennis.

```
$MAHOUT_HOME/bin/mahout tennnisclassifier -m playtennismodel -d 20-input -type bayes -ng 3 -source hdfs -method mapreduce
```

```
[java] 09/07/23 17:06:38 INFO bayes. tennnisclassifier: --------------
[java] 09/07/23 17:06:38 INFO bayes. tennnisclassifier: playingTennis      70.0
[21/30.0]
[java] 09/07/23 17:06:38 INFO bayes. tennnisclassifier: --------------
[java] 09/07/23 17:06:38 INFO bayes. tennnisclassifier: notPlayingTennis
81.3953488372093
[35/43.0]
[java] 09/07/23 17:06:38 INFO bayes. tennnisclassifier: Summary
[java] -------------------------------------------------------
[java] Correctly Classified Instances : 9    79.5%
[java] Incorrectly Classified Instances : 5    20.5%
[java] Total Classified Instances : 14
[java] -------------------------------------------------------
```

Default Category: unknown: 2

Confusion Matrix

```
a   b   <--Classified as
21  9   |  30       a   = playingTennis
8   35  |  43       b   = NotPlayingTennis
```

```
### FIGURE 4.
Summary Result of Playing Tennis after Naïve Bayes

<table>
<thead>
<tr>
<th>OUTLOOK</th>
<th>TEMPERATURE</th>
<th>HUMIDITY</th>
<th>WINDY</th>
<th>PLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>9</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>5</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A new day:

<table>
<thead>
<tr>
<th>OUTLOOK</th>
<th>TEMP.</th>
<th>HUMIDITY</th>
<th>WINDY</th>
<th>PLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

Likelihood of the two classes

For “yes” = 2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.0053

For “no” = 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0206

Conversation into a probability by normalization:

P(“yes") = 0.0026 / (0.0053 + 0.0206) = 0.205

P(“no") = 0.0206 / (0.0053 + 0.0206) = 0.795
COLLABORATIVE FILTERING

Collaborative filtering is a technique, popularized by Amazon and others, that uses user information such as ratings, clicks, and purchases to provide recommendations to other users of their site. Collaborative filtering is often used to recommend consumer items such as books, music, and movies, but it is also used in other applications where multiple actors need to collaborate to narrow down data. Chances are you’ve seen Collaborative filtering in action the last time you were on Amazon.

Given a set of users and items, Collaborative filtering applications provide recommendations to the current user of the system. Four ways of generating recommendations are typical:

- **User-based**: Recommend items by finding similar users. This is often harder to scale because of the dynamic nature of users.
- **Item-based**: Calculate similarity between items and make recommendations. Items usually don’t change much, so this often can be computed offline.
- **Model-based**: Provide recommendations based on developing a model of users and their ratings.
All Collaborative filtering approaches end up calculating a notion of similarity between users and their rated items. There are many ways to compute similarity, and most Collaborative filtering systems allow you to plug in different measures so that you can determine which one works best for your data.

Mahout currently provides tools for building a recommendation engine through the Taste library; it is a Java based flexible and a fast engine for recommendation. The library comes with classes that you can inherit the interfaces and properties. Taste supports both user-based and item-based recommendations and comes with many choices for making recommendations, as well as interfaces for you to define your own. Taste consists of five primary components that work with Users, Items and Preferences:

- **DataModel**: Storage for Users, Items, and Preferences.
- **UserSimilarity**: Interface defining the similarity between two users.
- **ItemSimilarity**: Interface defining the similarity between two items.
- **Recommender**: Interface for providing recommendations.
- **UserNeighborhood**: Interface for computing a neighborhood of similar users that can then be used by the Recommenders.

For example, to use Taste for book recommendation using a item-based recommendation, the first step is to load the data containing the recommendations and store them into the DataModel storing the users, Item book ID and their preferences.

```java
//create the data model for the user preferences
FileDataModel dataModel = new FileDataModel(new File(bookFile));
UserSimilarity userSimilarity = new PearsonCorrelationSimilarity(dataModel);
// Optional:
userSimilarity.setPreferenceInferrer(new AveragingPreferenceInferrer(dataModel));
```

The second step is to construct a LogLikelihoodSimilarity and a Recommender. The UserNeighborhood identifies users similar to the users and is handed off to the Recommender to create a ranked list of recommended books.

```java
//create the data model
FileDataModel dataModel = new FileDataModel(new File(recsFile));
//Create an ItemSimilarity
ItemSimilarity itemSimilarity = new LogLikelihoodSimilarity(dataModel);
//Create an Item Based Recommender
ItemBasedRecommender recommender =
    new GenericItemBasedRecommender(dataModel, itemSimilarity);
//Get the recommendations
List<RecommendedItem> recommendations =
    recommender.recommend(userId, 5);
TasteUtils.printRecs(recommendations, handler.map);
```
Running the main.java class will produce the following output.

```java
[echo] Getting similar items for user: Machine Learning with Mahout with a ItemSimilarity of 5
[java] 09/08/20 08:13:51 INFO file.FileDataModel: Reading file info...
[java] Data Model: Machine Learning with Mahout
[java] -----
[java] Title: Mahout 21 Rating: 3.630000066757202
[java] Title: Action in Mahout Rating: 2.703000068664551
[java] Title: Big Data Rating: 4.230000019073486
[java] Title: Correction Rating: 5.0
[java] Title: Abraham Lincoln Rating: 4.73999977118164
[java] Title: Data intelligence in bigdata: 3.430000066757202
[java] Title: Boston consulting in bigdata Rating: 2.00999990463257
[java] Title: Atlanta, Georgia Rating: 4.42999828338623
[java] Recommendations list below:
[java] Doc Id: 50575 Title: April 10 Score: 4.98
[java] Doc Id: 134101348 Title: April 26 Score: 4.860541
[java] Doc Id: 133445748 Title: Mahout In action Score: 4.430866
[java] Doc Id: 1193764 Title: Sam Doe Score: 4.404066
[java] Doc Id: 2417937 Title: Mike Olson Score: 4.24178
```

After processing the item of “Machine Learning with Mahout”, the system recommended several more books with various levels of confidence.
FREQUENT PATTERN MINING

Frequent patterns are itemsets and subsequence itemsets that appear in a data set with frequency no less than a user-specified threshold. For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set, is a frequent itemset. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern. Finding frequent patterns plays an essential role in machine learning associations, correlations, and many other interesting relationships among data.

Frequent pattern mining is very powerful to analyze customer buying habits by finding associations between the different items that customers place in their “shopping baskets”. For instance, if customers are buying milk, how likely are they going to also buy cereal (and what kind of cereal) on the same trip to the supermarket? Such information can lead to increased sales by helping retailers do selective marketing and arrange their shelf space.

FP-Growth Tree Algorithm is a high speed frequent item-set mining algorithm. It builds a prefix tree from an input data set which represents the data-set in a compressed form. Frequent item-sets can be pulled from the tree by a process pattern fragment growth involving projection of the FP-tree.

```
$ hadoop jar fp-growth-bell_mobility -0.4-job.jar
org.apache.mahout.fpm.pfpgrowth.FPGrowthDriver -i downloads-input -o reco-patterns-output -k 50 - method mapreduce -g 500 -regex '\[\ ]' -s 5
```

Apache Mahout has been rapidly adapted by the industry in the last two years. And it has continued to move forward in a number of ways. The current release has included many algorithms that can help solve real world problems. The community’s primary focus at the moment is on pushing toward a 1.0 release by doing performance testing, documentation, API improvement, and the addition of new algorithms. And the long term plan is to start looking at distributed, in-memory approaches to solving machine learning problems. It will make Mahout powerful and attractive to the machine learning practitioners. Mahout is well positioned to help solve today’s most pressing big-data problems by focusing in on scalability and making it easier to consume complicated machine learning algorithms.

**STEP 4: PUT THE MODEL TO USE AND FINE-TUNE IT**

It is fair to say machine learning is a trial and error process. There is no cookie cutting solution to a problem. When solving the basket analysis problem for a Canadian’s biggest Drug store chain, three different algorithms, K-means, Latent Dirichlet Allocation and FP-Growth tree were put into a trial and error process for a few months to land on a right algorithm. It is not an easy problem and requires a lot of efforts and patience. Thankfully, technology has helped the data churning process and made the process less painful.
SUMMARY OF APACHE MAHOUT

Apache Mahout has gained popularity in the last two years. Like the name of mahout suggests, a real mahout (elephant driver) leverages the strength and capabilities of the elephant, so too can Apache Mahout help you leverage the strength and capabilities of the yellow elephant of Apache Hadoop. Just within 2011, many user groups, forums, blogs, webinars and even start-up companies have sprout up around this new technology. And its community is becoming larger and stronger everyday to incorporate new algorithms with increasing capabilities for solving clustering, categorization, and collaborative filtering, frequent item mining or other machine learning problems. Observing the industry involvements and responses, in my opinion, Mahout likely will become the standard technology of machine learning on Big Data. Big technology firms like IBM, Oracle will incorporate Mahout in their product offerings to just acquire the user base.

REMARKS

Machine learning is definitely an exciting application that helps you to tap on the power of big data. As for corporate data continues to grow bigger and more complex, machine learning will become even more attractive. The industry has come up elegant solutions to help corporations to solve this problem. Let’s get ready; it is just a matter time this problem arrives at your desk.
Established in 1993, EPAM Systems, Inc. (NYSE: EPAM) is a software product development company. Headquartered in the United States, EPAM serves clients worldwide utilizing its award-winning global delivery platform and locations in 19 countries across North America, Europe, Asia and Australia. EPAM drives business value through four main service lines: digital engagement, intelligent enterprise, business transformation and advanced engineering. EPAM is recognized as a leader in the Product Development Services space, a primary agile services player and ranked among Top 10 Global Commerce Service Providers by leading independent research firms. According to the Forbes’ list of America’s Best 100 publicly held companies with annual revenue under $1 billion, EPAM ranks as the #1 technology company and #3 overall.

FOR MORE INFORMATION, PLEASE VISIT EPAM.COM