E6893 Big Data Analytics Lecture 5:

Streaming Big Data Analytics

Ching-Yung Lin, Ph.D.
Adjunct Professor, Dept. of Electrical Engineering and Computer Science

October 8th, 2021
Spark Streaming
Spark Streaming

- Input data stream
- Spark Streaming
- Batches of input data
- Spark Engine
- Batches of processed data
Spark Streaming

Figure: Overview Of Spark Streaming

https://www.edureka.co/blog/spark-streaming/
Spark Streaming

- **Basic Concepts**
  - Linking
  - Initializing StreamingContext
  - Discretized Streams (DStreams)
  - Input DStreams and Receivers
  - Transformations on DStreams
  - Output Operations on DStreams
  - DataFrame and SQL Operations
  - MLlib Operations
  - Caching / Persistence
  - Checkpointing
  - Accumulators, Broadcast Variables, and Checkpoints
  - Deploying Applications
  - Monitoring Applications

- **Performance Tuning**
  - Reducing the Batch Processing Times
  - Setting the Right Batch Interval
  - Memory Tuning

- **Fault-tolerance Semantics**
Spark Streaming Example

First, we import `StreamingContext`, which is the main entry point for all streaming functionality. We create a local `StreamingContext` with two execution threads, and batch interval of 1 second.

```python
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

# Create a local StreamingContext with two working thread and batch interval of 1 second
sc = SparkContext("local[2]", "NetworkWordCount")
ssc = StreamingContext(sc, 1)
```

Using this context, we can create a DStream that represents streaming data from a TCP source, specified as hostname (e.g. `localhost`) and port (e.g. 9999).

```python
# Create a DStream that will connect to hostname:port, like localhost:9999
lines = ssc.socketTextStream("localhost", 9999)
```

This lines DStream represents the stream of data that will be received from the data server. Each record in this DStream is a line of text. Next, we want to split the lines by space into words.

```python
# Split each line into words
words = lines.flatMap(lambda line: line.split(" "))
```

```python
# Count each word in each batch
pairs = words.map(lambda word: (word, 1))
wordCounts = pairs.reduceByKey(lambda x, y: x + y)
```

```python
# Print the first ten elements of each RDD generated in this DStream to the console
wordCounts.pprint()
```
Spark Streaming Example

```bash
$ ./bin/spark-submit examples/src/main/python/streaming/network_wordcount.py localhost 9999
```

Then, any lines typed in the terminal running the netcat server will be counted and printed on screen every second. It will look something like the following.

```
# TERMINAL 1:
# Running Netcat
$ nc -lk 9999
hello world
...
```

```
# TERMINAL 2: RUNNING network_wordcount.py

$ ./bin/spark-submit examples/src/main/python/streaming/network_wordcount.py localhost 9999
...


(\nhello,1\nworld,1\n...
```
Discretized Streams

DStream → RDD @ time 1 → RDD @ time 2 → RDD @ time 3 → RDD @ time 4

- Data from time 0 to 1
- Data from time 1 to 2
- Data from time 2 to 3
- Data from time 3 to 4
Discretized Streams

The diagram illustrates the discretization of data streams over time. It shows a DStream named `lines` with discretized time intervals from time 0 to 1, 1 to 2, 2 to 3, and 3 to 4. Similarly, another DStream named `words` follows the same discretization scheme. The `flatMap` operation is applied to the `lines` DStream, transforming it into the `words` DStream at each time interval.
Discretized Streams

https://www.edureka.co/blog/spark-streaming/
DStream Transforms

https://www.edureka.co/blog/spark-streaming/
Output DStreams

Transformed DStream → Output Operations → Output DStream

External Systems
- Database
- File System

https://www.edureka.co/blog/spark-streaming/
DStreams Caching

https://www.edureka.co/blog/spark-streaming/
DStreams Example — Twitter Sentiment Analysis

```scala
// Import the necessary packages into the Spark Program
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.SparkContext._

import java.io.File

object twitterSentiment {
  def main(args: Array[String]) {
    if (args.length < 4) {
      System.err.println("Usage: TwitterPopularTags <consumer key> <consumer secret> " + ":<access token> <access token secret>
      System.exit(1)
    }

    StreamingExamples.setStreamingLogLevels()

    // Passing our Twitter keys and tokens as arguments for authorization
    val Array(consumerKey, consumerSecret, accessToken, accessTokenSecret) = args.take(4)
    val filters = args.takeRight(args.length - 4)

    // Set the system properties so that Twitter4j library used by twitter stream
    // Use them to generate OAuth credentials
    System.setProperty("twitter4j.oauth.consumerKey", consumerKey)
    System.setProperty("twitter4j.oauth.consumerSecret", accessTokenSecret)

    val sparkConf = new SparkConf().setAppName("twitterSentiment").setMaster("local[2]")
    val ssc = new StreamingContext(sparkConf, None, filters)

    // Streaming code
    ssc.checkpoint("checkpointDirectory")
    val twitterStream = ssc.socketTextStream("localhost", 9999)

    // Process the incoming streaming data
    twitterStream.map { line =>
      val parts = line.split(" ")
      val timestamp = parts(0)
      val tweet = parts.drop(1).mkString(" 
      
      // Process the tweet
      
      // Save the sentiment to HDFS
      
    }

    ssc.start()
    ssc.awaitTermination
  }
}
```

https://www.edureka.co/blog/spark-streaming/
DStreams Example — Twitter Sentiment Analysis

//Input DStream transformation using flatMap
val tags = stream.flatMap { status => Get Text From The Hashtags }

//RDD transformation using sortBy and then map function
tags.countByValue().
  foreachRDD { rdd =>
    val now = Get current time of each Tweet
    rdd
      .sortBy(_.2)
      .map(x => (x, now))

    //Saving our output at ~/twitter/ directory
    .saveAsTextFile(s"~/twitter/$now")
  }

//DStream transformation using filter and map functions
val tweets = stream.filter { t =>
  val tags = t . Split On Spaces . filter(_.startsWith("#")) . Convert To Lower Case
tags.exists { x => true }

  val data = tweets.map { status =>
    val sentiment = SentimentAnalysisUtils.detectSentiment(status.getText)
    val tagss = status.getHashtagEntities . map(_.getText . toLowerCase) .
      (status.getText, sentiment.toString, tagss.toString())
  }

  data.print()
  //Saving our output at ~/ with filenames starting like twitters
data.saveAsTextFiles("~/tweets","20000")

  ssc.start()
  ssc.awaitTermination()
}

https://www.edureka.co/blog/spark-streaming/
DStreams Example — Twitter Sentiment Analysis

Results:

The following are the results that are displayed in the Eclipse IDE while running the Twitter Sentiment Streaming program.

All the tweets are categorized into Positive, Neutral and Negative according to the sentiment of the contents of the tweets

https://www.edureka.co/blog/spark-streaming/
Stream Analyses Technical Challenges
Example IP Packet Stream Instantiation

Inputs

Dataflow Graph

By IBM Dense Information Gliding Team
Semantic MM Filtering

Per PE rates:
- 200-500MB/s
- ~100MB/s
- 10 MB/s
Configurable Parameters of Processing Elements to maximize relevant information:

\[ Y''(X | q, R) > Y'(X | q, R), \]

with resource constraint.

Required resource-efficient algorithms for:
- **Classification**, **routing** and **filtering** of signal-oriented data: (audio, video and, possibly, sensor data)

**Input data X – Queries q – Resource R**
- \( Y(X | q) \): Relevant information
- \( Y'(X | q, R) \) \( \cdot \) \( Y(X | q) \): Achievable subset given \( R \)
Example: Distributed Video Signal Understanding (Lin et al.)

(Distributed Smart Sensors) Block diagram of the smart sensors

- **Encodings**
  - MPEG-1/2
  - 1.5 Mbps

- **GOP Extraction**
  - 320 Kbps

- **Feature Extraction**
  - 22.4 Kbps

- **Event Extraction**
  - 2.8 Kbps

- **CDS Features**

- **Control Modules**
  - User Interests
  - Resource Constraints

- **Resource Constraints**

- **User Interests**

- **Display and Information Aggregation Modules**

- **Sensor 1**

- **Sensor 2**

- **Sensor 3**

- **Sensor N**

- **TV broadcast, VCR, DVD discs, Video File Database, Webcam**

- **Concept Detection Processing Elements**
  - PE1: 9.2.63.66: 1220
  - PE2: 9.2.63.66: 1240
  - PE3: 9.2.63.66
  - PE4: 9.2.63.66:1235
  - PE5: 9.2.63.66: 1240
  - PE6: 9.2.63.66
  - PE7: 9.2.63.66
  - PE100: 9.2.63.66

- **Meta-data**
  - 600 bps

- **Control Modules**

- **Meta-data**

- **Display and Information Aggregation Modules**

- **Control Modules**

- **User Interests**

- **Resource Constraints**

- **Concept Detection Processing Elements**
  - Face
  - Female
  - Outdoors
  - Male
  - Indoors
  - Airplane
  - Chair
  - Clock
Semantic Concept Filters

E.g.:
Complexity Reduction Introduction

• Objective: Real-time classification of instances using Support Vector Machines (SVMs)
• Computationally efficient and reasonably accurate solutions
• Techniques capable of adjusting tradeoff between accuracy and speed based on available computational resources
SVM formulation

- **Given:**
  - Training instances $\{x_i\}$ with labels $y_i$

- **Objective:**
  - Find maximum margin hyperplane separating positive and negative training instances
Decision

- Score of unseen instance $u_j : w \cdot \phi(u_j)$
- In terms of Lagrangian multipliers
  $$\sum_i \alpha_i y_i k(x_i, u_j)$$
- Computational Cost: $O(n_{sv}d)$
  - $n_{sv}$: Number of support vectors
  - $d$: Dimensionality of each data instance
Problems

- Number of support vectors grows quasi-linearly with size of training set [Tipping 2000]

- Inner product with each support vector of dimensionality $d$ expensive
  - Example TREC2003
    - Human : 19745 support vectors
    - Face : 18090

- High data rates (10Gbits/sec) means large number of abandoned data
Example

- Processing Power 1 Ghz
- 10000 support vectors
- 1000 / 2 features per instance
- Order of at least $10^7$ operations required per stream per sec
- Translates to less than 100 instances evaluated per sec with only one classifier
Naïve Approach I – Feature Dimension Reduction

- Experimental Results for Weather_News Detector
- Model Selection based on the Model Validation Set
- E.g., for Feature Dimension Ratio 0.22, (the best selection of features are: 3 slices, 1 color, 2 texture selections), the accuracy is decreased by 17%.
Naïve Approach II – Reduction on the Number of Support

- Proposed Novel Reduction Methods:
  - Ranked Weighting
  - P/N Cost Reduction
  - Random Selection
  - Support Vector Clustering and Centralization

- Experimental Results on Weather_News Detectors show that complexity can be at 50% for the cost of 14% decrease on accuracy
Weighted Clustering Approach

- **Basic steps**
  - Cluster support vectors
  - Use cluster center as representative for all support vectors in cluster
  - Determine scalar weight associated with each cluster center
  - Use only cluster centers to score new instances
Cluster center weight (contd.)

- Choose $\gamma_i$ minimizing square of difference in scores over all $\pm_i$ and $d$

- Sub-cases:

\[
d \geq \Delta_i \quad \text{and} \quad d < \Delta_i
\]
Using the weights

- For every support vector in cluster
  - Distance $\Delta_i$ known
  - Two weights computed
- Cumulative effect of all support vectors in clusters additive
  - $\Delta_i$ because of various support vectors added up at center to simulate effect of all support vectors
- $\Delta_i$ sorted, weight arrays rearranged
Experiments

• Datasets
  • TREC video datasets (2003 and 2005)
    • 576 features per instance
    • > 20000 test instances overall
  • MNist handwritten digit dataset (RBF kernel)
    • 576 features
    • 60000 training instances, 10000 test instances

• Performance metrics
  • Speedup achieved over evaluation with all support vectors
  • Average precision achieved
Results (Mnist 0-4)

Average Precision

Speedup Ratio

0 1 2 3 4

0 0.95 0.96 0.9625 0.975 0.9875 1

© 2021 CY Lin, Columbia University
Results (Mnist 5-9)

Average Precision vs Speedup Ratio

- Average Precision values range from 0.7 to 1.0.
- Speedup Ratio values range from 1 to 1000.

Legend:
- 5
- 6
- 7
- 8
- 9
Results (TREC 2003)

- Average Precision

- Concept: Human, Outdoors, Sport-Event, Crowd, People-Event
  - AP_fast
  - AP_original

- Speedup

- Concept: Human, Studio-Setting, Crowd
Summary of Complexity Reduction

- Techniques presented demonstrate reasonable performance in terms of both speedup and average precision over multiple concepts in datasets.

- Speedups
  - MNist: All concepts at least 50 times faster with AP within 0.04 of original.
  - TREC 2003: Eight out of nine concepts speedup greater than 80 times with AP within 0.05 of original.
  - TREC 2005: APs in some cases along with speedup respectable.

- APs of most concepts close to original APs.
Acceleration of Neural Network for Streams
Methods for Running CNNs on Mobile Devices

- Sending CNN jobs to cloud
- Acceleration CNN on Local Device
- Apple A9X SoC, 12-core GPU

- How to trade off between algorithmic complexity and performance?
- How to utilize hardware effectively
Summary of Acceleration of Neural Network on Mobile Devices

- Porting Deep Convolution Neural Network on iOS Device with near real-time computation
  - Reduce algorithmic complexity with ignorable performance degradation.
  - Utilize computational hardware to achieve better performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>iPhone 7</th>
<th>iPhone 7+</th>
<th>iPhone 6s</th>
<th>iPad Pro 12.9&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex Net</td>
<td>70 ms</td>
<td>70 ms</td>
<td>130 ms</td>
<td>69 ms</td>
</tr>
<tr>
<td>p-Alex Net</td>
<td>N/A</td>
<td>35 ms</td>
<td>N/A</td>
<td>28 ms</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>130 ms</td>
<td>128 ms</td>
<td>195 ms</td>
<td>110 ms</td>
</tr>
<tr>
<td>p-GoogLeNet</td>
<td>N/A</td>
<td>80 ms</td>
<td>N/A</td>
<td>70 ms</td>
</tr>
<tr>
<td>VGG16 Net</td>
<td>880 ms</td>
<td>883 ms</td>
<td>1450 ms</td>
<td>725 ms</td>
</tr>
</tbody>
</table>
## Challenges for Running CNN on Mobile Devices

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Size</th>
<th>Weights</th>
<th>Mult.s</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>243MB</td>
<td>61M</td>
<td>725M</td>
</tr>
<tr>
<td>VGG-S</td>
<td>393MB</td>
<td>103M</td>
<td>2640M</td>
</tr>
<tr>
<td>VGG-16</td>
<td>552MB</td>
<td>138M</td>
<td>15484M</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>51MB</td>
<td>6.9M</td>
<td>1566M</td>
</tr>
</tbody>
</table>

Statistics of some popular CNNS

Reference:
Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications
## Computational Resource on iPhone and iPad

<table>
<thead>
<tr>
<th></th>
<th>iPhone 6S (Plus)</th>
<th>iPad Air 2</th>
<th>iPad Pro (12.9/9.7)</th>
<th>iPhone 7 (Plus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoC</td>
<td>A9</td>
<td>A8X</td>
<td>A9X</td>
<td>A10 Fusion</td>
</tr>
<tr>
<td>CPU</td>
<td>2x Twister @ 1.85 GHz</td>
<td>3x Typhone @ 1.5 GHz</td>
<td>2x Twister @ 2.26 GHz</td>
<td>4-core</td>
</tr>
<tr>
<td>GPU</td>
<td>PVR GT7600 (6 cluster)</td>
<td>PVR GXA6850 (8 cluster)</td>
<td>PVR 12 Cluster Series 7</td>
<td>6 cluster GPU?</td>
</tr>
<tr>
<td>RAM (shared memory)</td>
<td>2GB LDDR4</td>
<td>2GB LDDR3</td>
<td>4GB LDDR4</td>
<td>3GB on Plus?</td>
</tr>
<tr>
<td>Memory bus width</td>
<td>64-bit</td>
<td>128-bit</td>
<td>128-bit</td>
<td>?</td>
</tr>
<tr>
<td>Max # of threads per group</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>?</td>
</tr>
</tbody>
</table>
Neuron Importance Score Propagation (NISP, Yu et al 2018)
Methods for Running CNNs on Mobile Devices

- Sending CNN jobs to cloud
- Compression (pruning) of CNN
  + Speeding up CNN
  + Sending CNN jobs to cloud
Thinking Differently

- All existing methods can be viewed as approximations of an overly-redundant CNN. But do we really need such a CNN as the starting point?
Slim CNN

- Slim CNN leads to:
  - less storage space
  - less memory usage
  - less computation
  - less power consumption
Feature Selection on CNN

- CNNs can be viewed as a set of "overly-redundant" feature extractors
A method for Pruning Redundant Neurons and Kernels of A pre-trained CNN

1. Extract CNN Responses
2. Measure the Importance of Feature Extractors
3. Prune Model
4. Fine-tuning

Apply thermal
A method for Pruning Redundant Neurons and Kernels of Deep Convolutional Neural Networks (NISP)

- Intractable → tractable
- Inconsistent → consistent

A pre-trained CNN → Extract Responses of a High-level Layer → Measure the Importance of Feature Extractors → Back-propagate the Importance & Prune Model → Fine-tuning

Forward Propagation

Input layers → Response → ... → Response → ... → Response → ... → FC layers

Important Score Back Propagation and Pruning
Fine-tuning the Pruned Model

- Our method outperforms the baselines in three aspects
  - Very small accuracy loss at the beginning ==> retains the most important neurons
  - Converges much faster than baselines
  - For LeNet on MIST, our method only decreases 0.02% top-1 accuracy with a running ratio of 50% as compared to the pre-pruned network.

(a) MNIST
Fine-tuning the Pruned Model

- The pruned model consists of important feature extractors, but will suffer loss of accuracy due to loss of redundant features
  - Good starting point on the learning curve due to feature selection
  - Fine-tuning the pruned model with a lower learning rate to recover the performance
Stream Analyses on Social Media Data
Social Media Stream Monitoring

Goal 1: Detect, classify, measure and track the
(a) formation, development, and spread of ideas & concepts (memes)
(b) purposeful or deceptive messaging and misinformation

Goal 2: Recognize persuasion campaign structures and influence operations across social media sites and communities

Goal 3: Identify participants and intent, and measure effects of persuasion campaigns

Goal 4: Counter messaging of detected adversary influence operations

53+ papers published, accepted, & submitted
12+ patents filed
ACM CIKM 2012 Best Paper Award
IEEE BigData 2013 Best Paper Award
PNAS Cover Article Jan 2013
Science (1)
Nature (2)

Approach: Modeling, Tracking and Affecting Information Dissemination in Context

Thrust 1. Modeling Information Dissemination in Context:
Models of Trust and Social Capital, Information Morphing, Persuasiveness and Competition of Memes, Dynamics of Social Influence

Thrust 2. Detecting and Tracking Information Dissemination in Context:
Detecting Malicious Info Propagation, Evolution History and Authenticity of Multimedia Memes,

Thrust 3. Affecting Information Dissemination in Context:
Automated Generation of Counter Messaging, Influencing the Outcome of Competing Memes and Counter Messaging

Social Media Analytics Infrastructure

Goal 1: Detect, classify, measure and track the
(a) formation, development, and spread of ideas & concepts (memes)
(b) purposeful or deceptive messaging and misinformation

Goal 2: Recognize persuasion campaign structures and influence operations across social media sites and communities

Goal 3: Identify participants and intent, and measure effects of persuasion campaigns

Goal 4: Counter messaging of detected adversary influence operations

53+ papers published, accepted, & submitted
12+ patents filed
ACM CIKM 2012 Best Paper Award
IEEE BigData 2013 Best Paper Award
PNAS Cover Article Jan 2013
Science (1)
Nature (2)
Social Media Solution

**Live Monitoring**
Monitoring real-time tweets on keyword:
- Monitor live tweets »

**Trend Monitoring**
Analyzing trend of conversations based on hashtags
- View trends »

**Multimedia Monitoring**
Recognizing visual content and analyzing visual sentiments
- View multimedia »

**Geo Monitoring**
Monitoring the places that people are sending out tweets
- View places »

**Scope Identification**
Define user-specified sets of keywords for monitoring and analytics
- Define scopes »

**Concept Analytics**
Analyzing statistics of groups based on time, topics, etc
- Concept searches »

**Link Exploration**
Visualizing relationships, discussion sequences and graphs
- View relationships »

**Impact Prediction**
Analyzing conversations and predicting their impact to business
- View impacts »

**Story Detection**
Detecting live developing stories on social media and their evolution
- View stories »

**Person Analytics**
Analyzing a person’s personality, trustworthiness, etc.
- View person »

**Target Discovery**
Inspecting potential users for bot detection, marketing, or influencing
- Inspect targets »

**Forensic Analytics**
Analyzing retweet sequences and displaying anomalies
- View anomalies »
Social Media Stream Monitoring Applications

- **Live Monitoring**: Monitoring real-time tweets on keyword.
  - Monitor live tweets ➔ View trends ➔ View multimedia ➔ View places ➔ Define scopes ➔ Concept searches

- **Trend Monitoring**: Analyzing trend of conversations based on hashtags.

- **Multimedia Monitoring**: Recognizing visual content and analyzing visual sentiments.
  - View multimedia ➔ View places ➔ Define scopes ➔ Concept searches

- **Geo Monitoring**: Monitoring the places that people are sending out tweets.
  - View places ➔ Define scopes ➔ Concept searches

- **Scope Identification**: Define user-specified sets of keywords for monitoring and analytics.
  - Define scopes ➔ Concept searches

- **Concept Analytics**: Analyzing statistics of groups based on time, topics, etc.
  - Concept searches

- **Link Exploration**: Visualizing relationships, discussion sequences and graphs.
  - View relationships ➔ View impacts ➔ View stories ➔ View person ➔ Inspect targets ➔ View anomalies

- **Impact Prediction**: Analyzing conversations and predicting their impact to business.
  - View impacts ➔ View stories ➔ View person ➔ Inspect targets ➔ View anomalies

- **Story Detection**: Detecting live developing stories on social media and their evolution.
  - View stories ➔ View person ➔ Inspect targets ➔ View anomalies

- **Person Analytics**: Analyzing a person’s personality, trustworthiness, etc.
  - View person ➔ Inspect targets ➔ View anomalies

- **Target Discovery**: Inspecting potential users for bot detection, marketing, or influencing.
  - Inspect targets ➔ View anomalies

- **Forensic Analytics**: Analyzing retweet sequences and displaying anomalies.
  - View anomalies

- **Images in tweets that belong to one story**
- **Text of the newest tweet in this story**
**Bank Use Case**

- **Objective:** Detect unexpected social media movements that may impact a major bank’s business

- **X-Bank:**
  - Major bank in Spain

- **Client needs:**
  - Monitor Catalan independence movement: independence may bring bankruptcy since X-Bank needs ECB support
  - Detect potential PR crisis by analyzing the formation and spreading of grassroots opinion on their employees and services

- **Challenges:**
  - Existing social media monitoring tools miss important tweets that don’t contain specified keywords and are not from specified users
  - Existing tools lack of predictive capability of tweets’ potential influence

An image tweet (without mentioning “the bank name”) sparks a lot critiques of their unfair practice
City Government Use Case

**Objective:** Detect unexpected social media discussions that may impact a government’s campaign

**X-Government:**
- A government in Asia

**Client needs:**
- Foster society diversity and fusion: Need to understand the citizens’ responses to a government policy on social welfare on minority
- Detect growing topic trends and multimedia discussions around the policy

**Challenges:**
- Detecting emerging trends as well as the powerful multimedia memes
- Existing tools lack of predictive capability of tweets’ potential influence
Live Monitoring
Live Monitoring

Monitoring categories

Monitoring filter

Growing Influential Retweet Graphs

Real-Time Translation, Locations, Top Retweets

Live Tweets, Sentiment, Keywords
Real-time Trend Analytics

The chart shows top 4 hashtags for 'isis' topic during February 27, 2015 to March 2, 2015

(285667 tweets for hashtag iraq loaded. 236532 tweets for hashtag isis loaded. 199351 tweets for hashtag eu loaded. 198716 tweets for hashtag suuniiraqciviliansvoice loaded.)

Click on the histogram below to see raw tweets.
Multimedia Monitoring
To be integrated soon — Automatic Affective Commenting

- **VAC Prediction**
  - “wonderful,” “lovely,” “peaceful,” “moody,”...

- **Candidate Comments**
  - “Awesome and cute!”
  - “Lovely moody shot - so peaceful”

- **Automatic Commenting**
  - “lovely moody shot - so peaceful!”
What and Where of Image Sentiment Analysis

What’s in the image that makes a dog cute vs. scary?

What makes a beautiful landscape beautiful?
Detecting Politically Persuasive Web Content

Summary

- Extract Affective and Semantic Information
  - Audio, Visual and Textual Modalities

- Detection of Politically Persuasive Content
  - Additionally predict viewer response in advance
Scope Identification
Scope Discovery

User Initiative -- Common_Core_01

Topics and terms from Common_Core_01

<table>
<thead>
<tr>
<th>School</th>
<th>school x</th>
<th>high school x</th>
<th>middle school x</th>
<th>elementary school x</th>
<th>class x</th>
<th>gradeschool x</th>
<th>grade schoo. x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Core Test</td>
<td>educational standards x</td>
<td>state standards x</td>
<td>state testing x</td>
<td>standardized testing x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Professional Devel</td>
<td>Training x</td>
<td>curriculum materials x</td>
<td>curriculum development x</td>
<td>professional development x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teacher</td>
<td>Teacher x</td>
<td>Lecturer x</td>
<td>Tutor x</td>
<td>Instructor x</td>
<td>teach person x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add a topic and terms to Common_Core_01

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>Add</th>
</tr>
</thead>
</table>

Advance Settings for Segment Analytics
Initiative: ISIS_07

Foci

- _ISIS_Name_Occurrences: 21426
- _ISIS_In_Iraq_or_Syria: 9734
- _ISIS_Marriage_History: 108
- _ISIS_Extremist_Military_Action: 266
- _ISIS_Military_Action: 833
- _Syrian_Opposition_Groups: 15162
- _ISIS_Social_Media_Recruiting: 1727
- _Islamic_Religion_History: 16895
- _Western_Political_Response_History: 3078
- _ISIS_Travel: 8969

Map:

- U.S. Only
- Worldwide

State: Ohio: 1682

# of profiles in states

- Parents: 50%
- Male: 28%
- NotMale: 72%
- NotParent: 50%
- Unknown: 27%
- Married: 96%
- NotMarried: 4%
- BusinessOwner: 96%
- NotBusinessOwner: 4%
Impact Prediction
# Impact Prediction

## Real-time hashtag monitoring

Predicting the business impact of tweet messages grouped by hashtags.

Please click on the "hashtag" to learn more about each conversations content.

Last updated at 2015-03-03 03:20:01 GMT

<table>
<thead>
<tr>
<th>#</th>
<th>Conversations</th>
<th>Impact</th>
<th>Impact Score</th>
<th>Prediction Detail</th>
<th>First Tweet Time</th>
<th>Last Tweet Time</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>russia, turkey</td>
<td>HIGH</td>
<td>52.93</td>
<td>URL</td>
<td>2015-02-18 17:36:59</td>
<td>2015-03-01 09:10:57</td>
<td>265 hours</td>
</tr>
<tr>
<td>2</td>
<td>isis, syria</td>
<td>HIGH</td>
<td>44.66</td>
<td>URL</td>
<td>2015-02-20 14:29:51</td>
<td>2015-03-01 08:58:20</td>
<td>210 hours</td>
</tr>
<tr>
<td>3</td>
<td>isis, syria, turkey</td>
<td>HIGH</td>
<td>42.1</td>
<td>URL</td>
<td>2015-02-20 07:12:33</td>
<td>2015-03-01 08:58:20</td>
<td>217 hours</td>
</tr>
<tr>
<td>4</td>
<td>syria, turkey</td>
<td>HIGH</td>
<td>40.77</td>
<td>URL</td>
<td>2015-02-27 14:31:05</td>
<td>2015-03-01 08:13:34</td>
<td>42 hours</td>
</tr>
<tr>
<td>5</td>
<td>erdogan</td>
<td>HIGH</td>
<td>39.92</td>
<td>URL</td>
<td>2015-02-27 23:35:06</td>
<td>2015-03-01 09:12:05</td>
<td>33 hours</td>
</tr>
<tr>
<td>6</td>
<td>syria, turkey, us</td>
<td>HIGH</td>
<td>38.28</td>
<td>URL</td>
<td>2015-02-19 02:45:53</td>
<td>2015-02-28 11:50:52</td>
<td>225 hours</td>
</tr>
<tr>
<td>7</td>
<td>erdogan, turkey</td>
<td>HIGH</td>
<td>36.89</td>
<td>URL</td>
<td>2015-02-24 02:16:10</td>
<td>2015-03-01 09:12:05</td>
<td>126 hours</td>
</tr>
</tbody>
</table>
Flow Analytics
Collective intelligence and predicting market movement

Data:
- instant messages and trades by employees of a large hedge fund
- 24 traders, 95 analysts, 63 portfolio managers, 8646 outside contacts
- 47K trades
- 12 million IMs (2010–2011)

Findings: We identify two behavioral patterns that correlate with changes in the market:
- Reaction to IMs containing relevant information
- Attention to people who work with the same/different stocks

Using these two signals, we can make predictions of the market movement with better accuracy than the hedge fund did.
13:11:33, I was thinking all this AAPL anti-trust might be actionable
13:11:42', not great for AAPL
13:11:47, When GOOG had that big issue in Europe stock underperformed right?
13:11:52, true
13:14:01, Also not sure if you caught, but GSCO is going to allow employees to bring own phone
device for corporate email
13:14:24, Maybe GSCO allowing that could be positive for AAPL, as security focused firm saying
iPhone works
13:14:35, But bad for RIMM
13:14:42, Maybe all this is priced in
13:16:50, Did you see speculation that Bing is actually quietly going to be default search on
iPhone 4?
13:17:18, heard a lot of talk of that
13:17:23, but didn't see that specifically like that
13:17:44, Okay let me figure out where I saw that and get back to you
13:17:45, One sec
Relay: How urgently do employees relay information?

\[ T_2 - T_1 < h \text{ hours} \]

\[ R_h(t) = \text{fraction of times users send an IM containing a stock symbol to another user within at most } h \text{ hours after having received an IM containing the stock symbol} \]

“I think AAPL is going up”

“Pay attention to AAPL”

“I heard AAPL is doing well”
Concentration: Do employees focus on others who discuss the same stock symbols?

\[
C_{(A,B)}(t) = \text{fraction of messages between } A \text{ and } B \text{ and all messages of } A \text{ and } B
\]

In example: \(10/(20+10+5) = 0.29\)

\[C_k(t) = \text{average } C_{(A,B)}(t) \text{ among all pairs } (A, B) \text{ that share } k \text{ stock symbols}\]
• Motivation
Emotional states can effect how information is processed. Good information can be undermined or strengthened by emotional states.

• Approach
  – Measure emotional activation in tweets using the ANEW dictionary
  – Control for the number of words in the text

• Preliminary experiments
  – Traders at a hedge fund are more likely to make decision errors when they are very emotionally activated or very emotionally deactivated.
  – Users who retweeted the 20 detected most anomalous sequences tend to post tweets with higher level of emotion than a baseline of 20 million tweets from June 2009.
Finding optimal edges to slow down propagation

- **Task:** Minimize propagation by deleting edges of a graph

![Graphs at different time steps](image)

- **Problem:** Given a graph $G$ and a budget $k$, find the $k$ edges to delete in order to get the largest drop in the leading eigenvalue $\lambda_1$ of $G$

Yahoo IM!

- **NetMelt**
- **MET**

  - **Red edges** are selected for deletion
  - **NetMelt** [Tong et al., CIKM 2012] tracks one eigenvalue, $\lambda_1$
  - Our **MET** tracks on average 5.17 eigenvalues
Measuring Human Essential Traits in Social Media

- **Personality:** Mapping personal/organizational social media postings to scores of BIG 5 Personality *(Openness, Conscientiousness, Extraversion, Agreeableness, and Neurocism)*

- **Needs:** Mapping personal/organizational social media postings to scores of Harmony, Curiosity, Self-expression, Ideal, Excitement, and Closeness.

- **Values:** Mapping personal/organizational social media postings to scores of Self-Enhance, Conservation, Open-to-Change, Hedonism, and Self-Transcend.

- **Trustingness and Trustworthiness:** Deriving from interaction and propagation history between the user and his followers and the people he follows.

- **Influence:** Total attention received by user as leader across all discovered flows.

  => Preliminary studies showed propagation behavior is related to these social cognitive traits.
Target Discovery
Target Discovery
Detecting Anomalous Information Spreading

• Motivation
  – People’s dynamic reactions to information (e.g., retweeting) give clues to information credibility and quality
    • For example, trustworthy people may take time to verify uncertain information from strangers before spreading it

• Approach
  – Use one-class conditional random field to model people’s behavior in information spreading sequences and detect anomalous sequences
    • Features: content features such as emotion, network features such as tie strengths and clustering coefficients

• Preliminary experiments
  – Detect anomalies in retweeting sequences during Hurricane Sandy
    • Including hijacker, fake pictures spreaders
Photo Tweet Sentiment Tracking during Hurricane Sandy (Columbia University)

- **Goal:** Detect sentiment during Hurricane Sandy.

- **Data collection:**
  - Date: Oct 25 – Nov 02
  - Hashtags (based on popularity): #prayforusa, #frankenstorm, #nyc,#hurricane,#sandy,#hurricanesandy, #staysafe, #redcross,#myheartgoesouttoyou,…
  - 2000 Photo Tweets collected

- **Ground Truth Labeling:**
  - 1340 unanimously agreed labels from 2 individuals

- **Training Classifier:**
  - Text (SentiStrength)
  - Visual(SentiBank, Logistic Regr.)
  - Training/Testing ratio: 4:1
  - 5-fold cross-validation
  - Accuracy(Text-Visual Combined): 72%
Forensic System

Detecting Anomalous Sequences that may be disinformation

RT sequence extraction & filtering (keyword, loc etc)

Emotion Analytics
Egonet Analytics
Graph Similarity (Rutgers)
Influence/Trust (UMN)
Human Personality & Value
Emotion Activation (UNW)

Multimedia Sentiment Tracking

Anomalous Sequence Detection By One-Class CRF

Top K Sequences

*Note: Components in blue are work in progress and applied to the top-K sequences
Users can interactively analyze the personality traits, emotion activation, value and other traits of the people involved.
Finding Anomalies

Detect disinformation spreading (retweet) during Hurricane Sandy by people’s dynamic behavior in response to the information

Detected rumor: Shark in NJ

The rumor sequence has low trustworthiness

Users involved

Human personality, value traits to show (showing Trustworthiness)

User info (e.g., personality chart)
Homework #2 (Due 10/22/2021, 5pm)

See HW2 and Tutorial Slides