presentation starting soon… sit down

Introduction to Online Machine Learning Algorithms

Dataworks Summit, San Jose 2017

Trevor Grant
@rawkintrevo
June 15th, 2017

Copyright Notice

Live tweeting at
#trevolive
@rawkintrevo

I probably violated most if not all copyright, copyleft, and fair useage agreement and treaties while collecting gifs, memes and images for this presentation. If you see something that's yours and want it taken down, I'll just delete the whole deck.

As such, I can't reasonably expect anyone to respect my IP here, so take whatever you want and do whatever you want with it. If you claim it as your own original work, I only ask that you go do some good deed (buy a homeless person lunch, call your grandma just to say hi, etc) to balance out your karma.

Table of Contents

#trevolive

- Who is this guy?
- Why should I care?
- What's going on here?
- This seems boring and mathy, maybe I should leave…

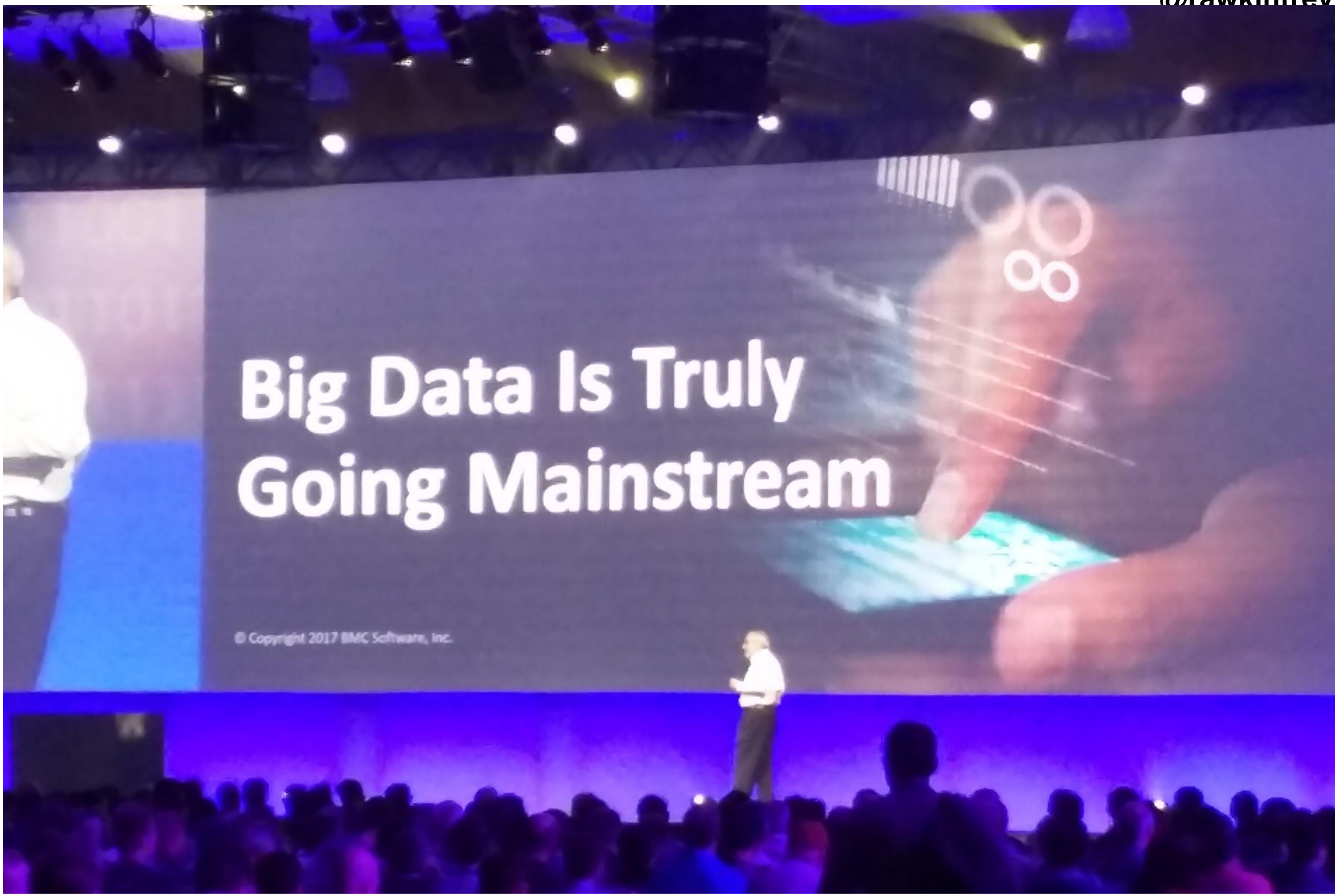
Branding

Live tweeting at
#trevolive
@rawkintrevo

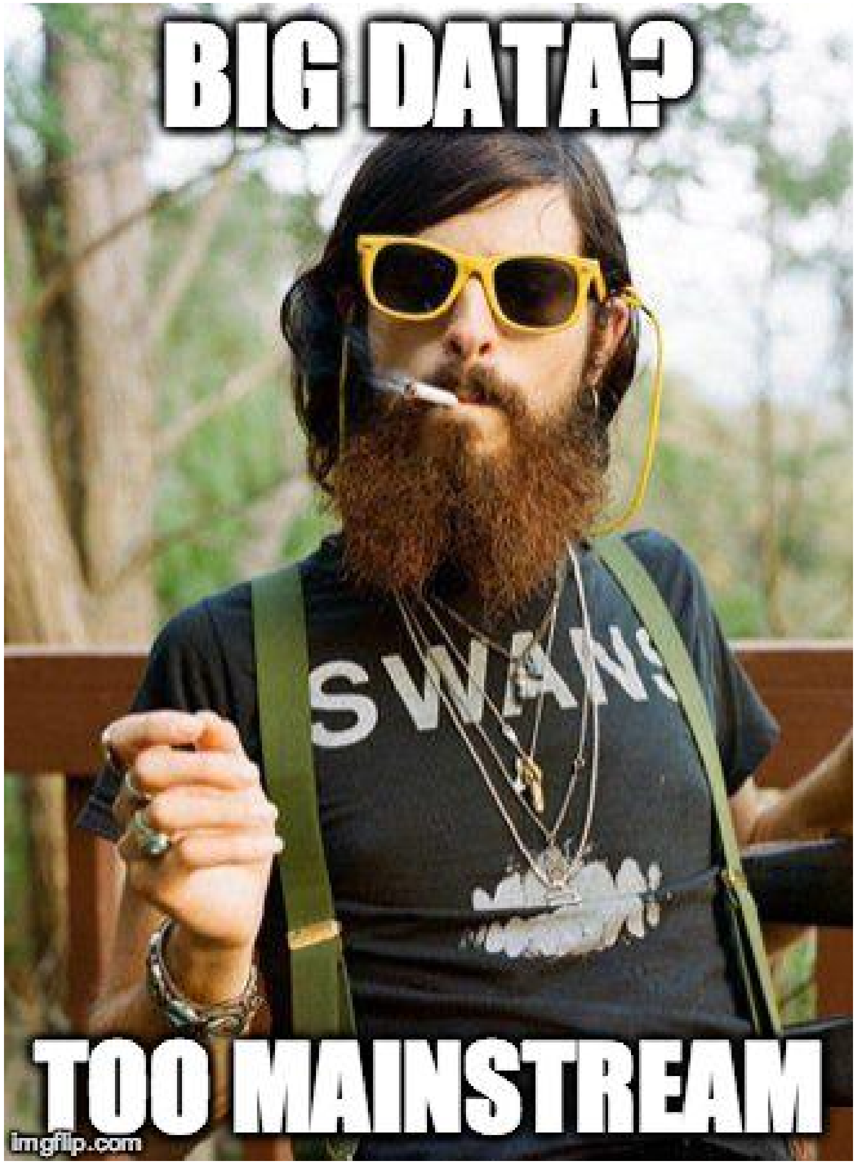
- Trevor Grant
- Things I do:
  - Open Source Technical Evangelist, IBM
  - PMC Apache Mahout, PPMC Apache Streams-Incubating
  - Blog: http://rawkintrevo.org
- Schooling
  - MS Applied Math, Illinois State
  - MBA, Illinois State
- How to get ahold of me:
  - @rawkintrevo
  - trevor.grant@ibm.com / rawkintrevo@apache.org
  - Mahout Dev and User Mailing Lists

Guess when I finished my slides?

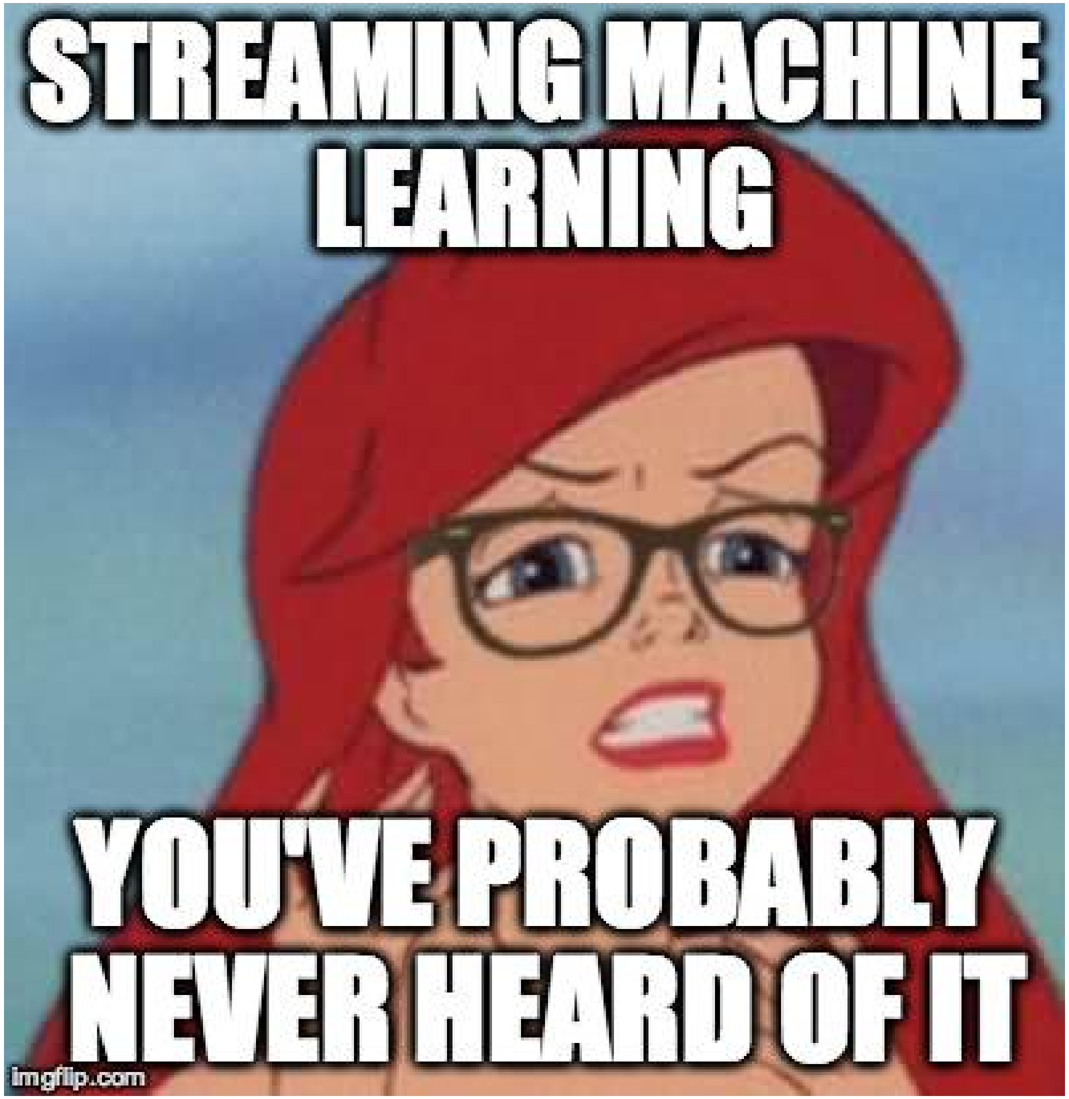
Live tweeting at
#trevolive
@rawkintrevo



Live tweeting at
#trevolive
@rawkintrevo



Live tweeting at
#trevolive
@rawkintrevo



Live tweeting at
#trevolive
@rawkintrevo

Interactive Theater

Live tweeting at
#trevolive
@rawkintrevo

Please excuse me while I live tweet things I'm doing.

You can join in- Tweet with hashtag **#trevolive** to help my demo

Tweet to @rawkintrevo if you want me to read it later

Table of Contents

#trevolive

- On the virtues of not throwing around buzzwords…
- Online vs. Offline
- Lambda vs. Kappa (w.r.t. machine learning)
- Statistical vs Adversarial
- Real-Time (one buzzword to rule them all)

Online vs. Offline

Live tweeting at
#trevolive
@rawkintrevo

Online

- Input processed piece by piece in a serial fashion
- Each new piece of information generates an event
  - Not mini-batching
  - Possibly on a sliding window of record 1
- Not necessarily low latency

Offline

- Input processed in batches
- Not necessarily high latency

Online (Processing in Serial)

Live tweeting at
#trevolive
@rawkintrevo

Offline (Processing in Batch)

Live tweeting at
#trevolive
@rawkintrevo

Fast offline, slow online and stack order

Live tweeting at
#trevolive
@rawkintrevo

Slow Online

Stock broker in Des Moines Iowa writes Python program that get's EOD prices/statistics as they are published and then executes orders.

Fast Offline

HFT algorithm, executes trades based on tumbling windows of 15 milliseconds worth of activity

Online doesn't mean fast, online doesn't mean streaming, online ***only means that it processes information as soon as it is received***.

Consider an online algorithm (the slow online example), exists behind an offline EOD batch job.

- This is an extreme case, but no algorithm receives data as it is created.
- Best case- limited by speed of light (?)

Slow Online

Live tweeting at
#trevolive
@rawkintrevo

Fast Offline

Live tweeting at
#trevolive
@rawkintrevo

Lambda vs. Kappa (Machine Learning)

Live tweeting at
#trevolive
@rawkintrevo

<u>Lambda</u>

Learning happens (i.e. models are fitted) offline

Model used by streaming engine to make decisions online

<u>Kappa</u>

Learning happens (i.e. models are fitted) online

Online decision model updates for each new record seen

Model can change structure e.g. new words in TF-IDF or new categories in ‘factor model linear regression’

Lambda with Novell Information

Live tweeting at
#trevolive
@rawkintrevo

Lambda with Novell Information

Live tweeting at
#trevolive
@rawkintrevo

- A ***trained model*** expects structurally the same as training data.
- In linear regression, categorical features are “one-hot-encoded”. A feature with 3 categories expressed as a vector in 2 columns.
- What if a new category pops up?
  - Depends how you program it-
    - ignore the input
    - serve a bad response
- Consider clustering classification on text… new words?
  - Ignore: (probably what you’ll do)
  - Word might be very important...

Kappa with Novell information

Live tweeting at
#trevolive
@rawkintrevo

Kappa with Novell Information

Live tweeting at
#trevolive
@rawkintrevo

- In Kappa, training happens with each new piece of data
  - Model data can account for structural change in data instantly

- New words can be introduced into TF-IDF
- New categories into a factor variable
- Both examples (and others) causes input vector to change.

Statistical vs. Adversarial

Live tweeting at
#trevolive
@rawkintrevo

<u>Traditional</u>

Common statistical methods

- Supervised
- Unsupervised

Graded by

- Statistical Fitness Tests
- Out of core testing
- E.g.
  - Confusion Matrix, AuROC
  - MSE, MAPE, R2, MSE

<u>Adversarial</u>

Algorithm Versus Environment

- vs. Spammers
- vs. Hackers
- vs. Nature

Graded by

- Directionally can use some tests
- Really A/B testing
  - Adversaries may get smarter over time
  - Type of test where you automate adversary.
- Multi armed Bandit

Real-time

Live tweeting at
#trevolive
@rawkintrevo

- Subjective
- A good buzzword for something that:
  - Doesn't fall into any of the above categories cleanly
  - Doesn't fall into the category you want it to fall into
  - You're not really sure which buzzword to use, so you need a 'safe' word that no one can call you on.
  - Days
  - Weeks?
  - My Old jobs
    - JJs
    - Media Strategy
    - CPG Analytics

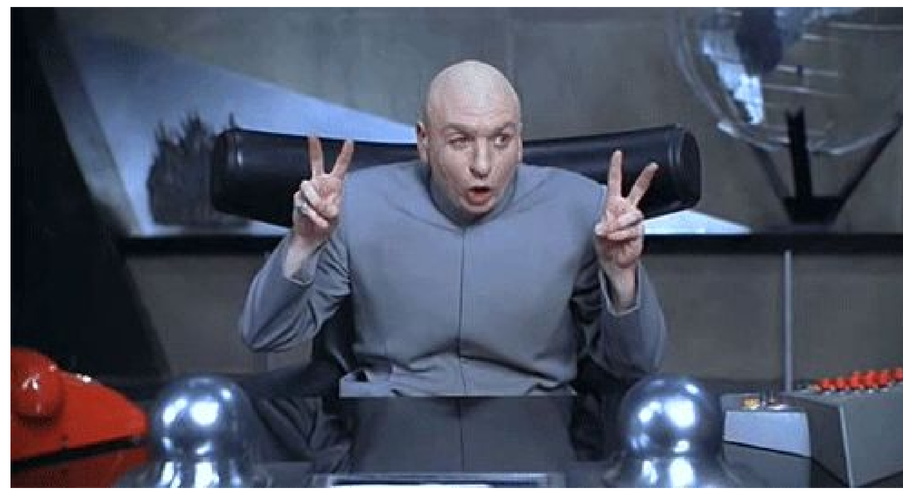


Table of Contents

#trevolive

- Streaming K-Means
- Streaming Linear Regression
- Why would I ever do with this?

K-Means

Live tweeting at
#trevolive
@rawkintrevo

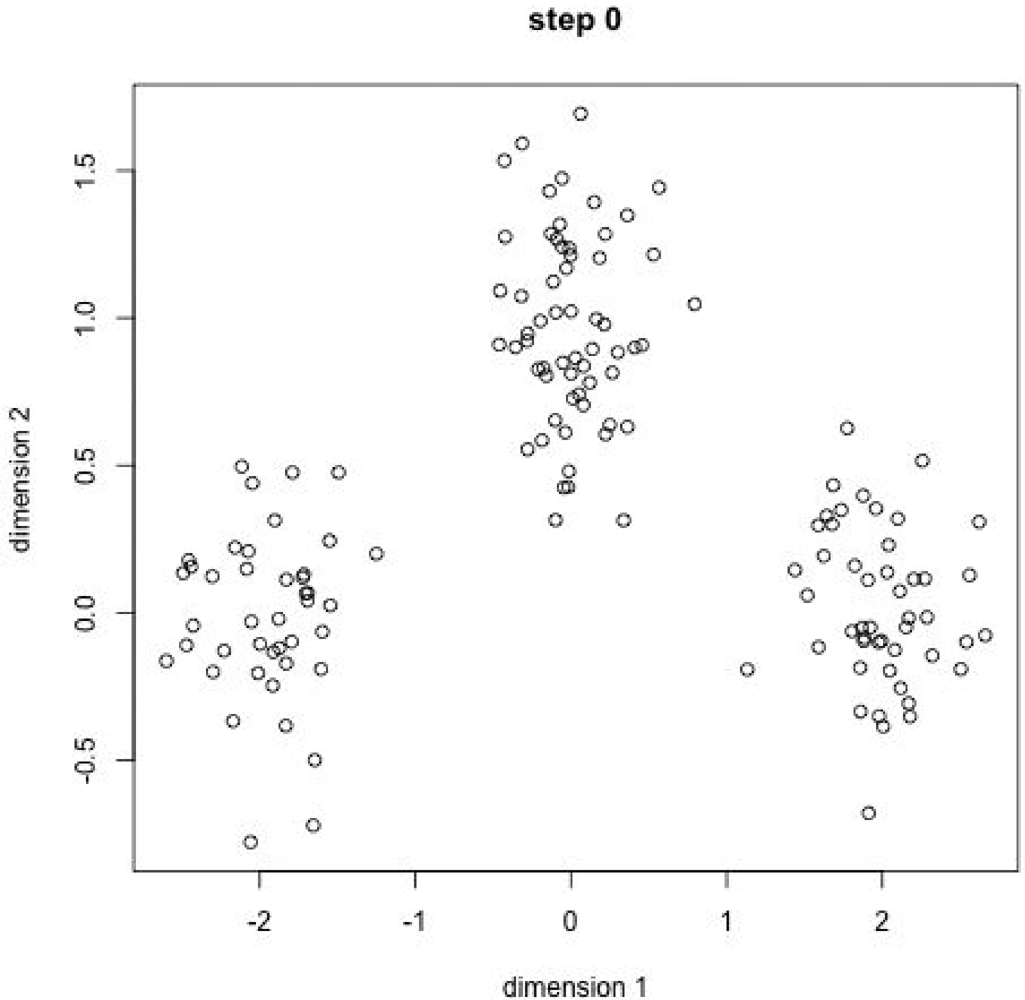


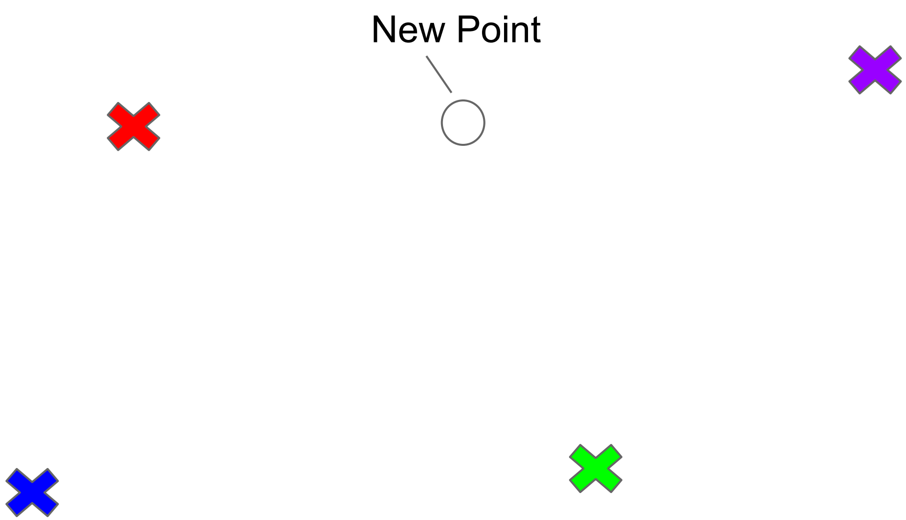
Photo Credit: Simply Statistics http://simplystatistics.org/wp-content/uploads/2014/02/kmeans.gif

Online K-Means

Live tweeting at
#trevolive
@rawkintrevo

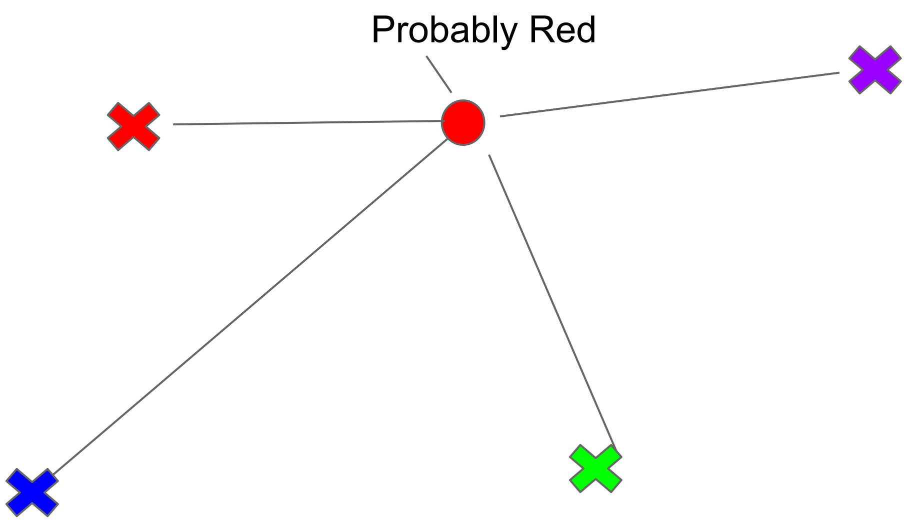
Online K-Means

Live tweeting at
#trevolive
@rawkintrevo



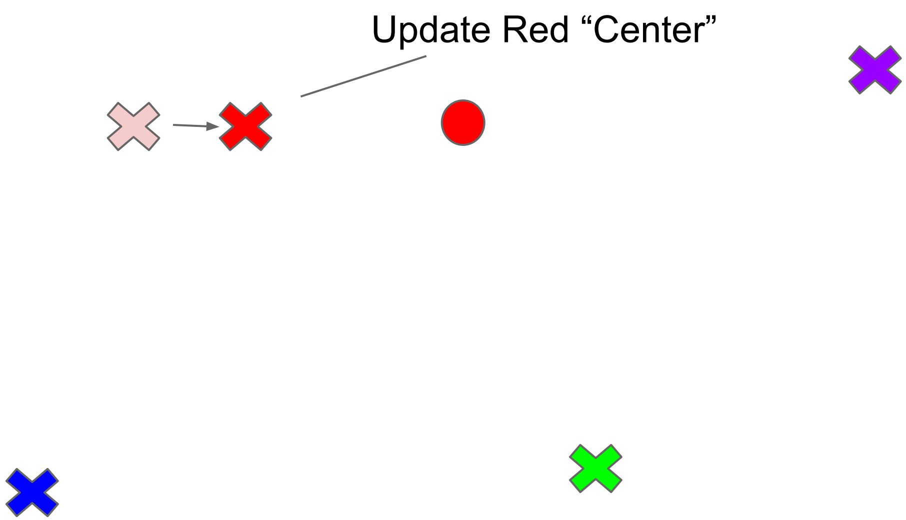
Online K-Means

Live tweeting at
#trevolive
@rawkintrevo



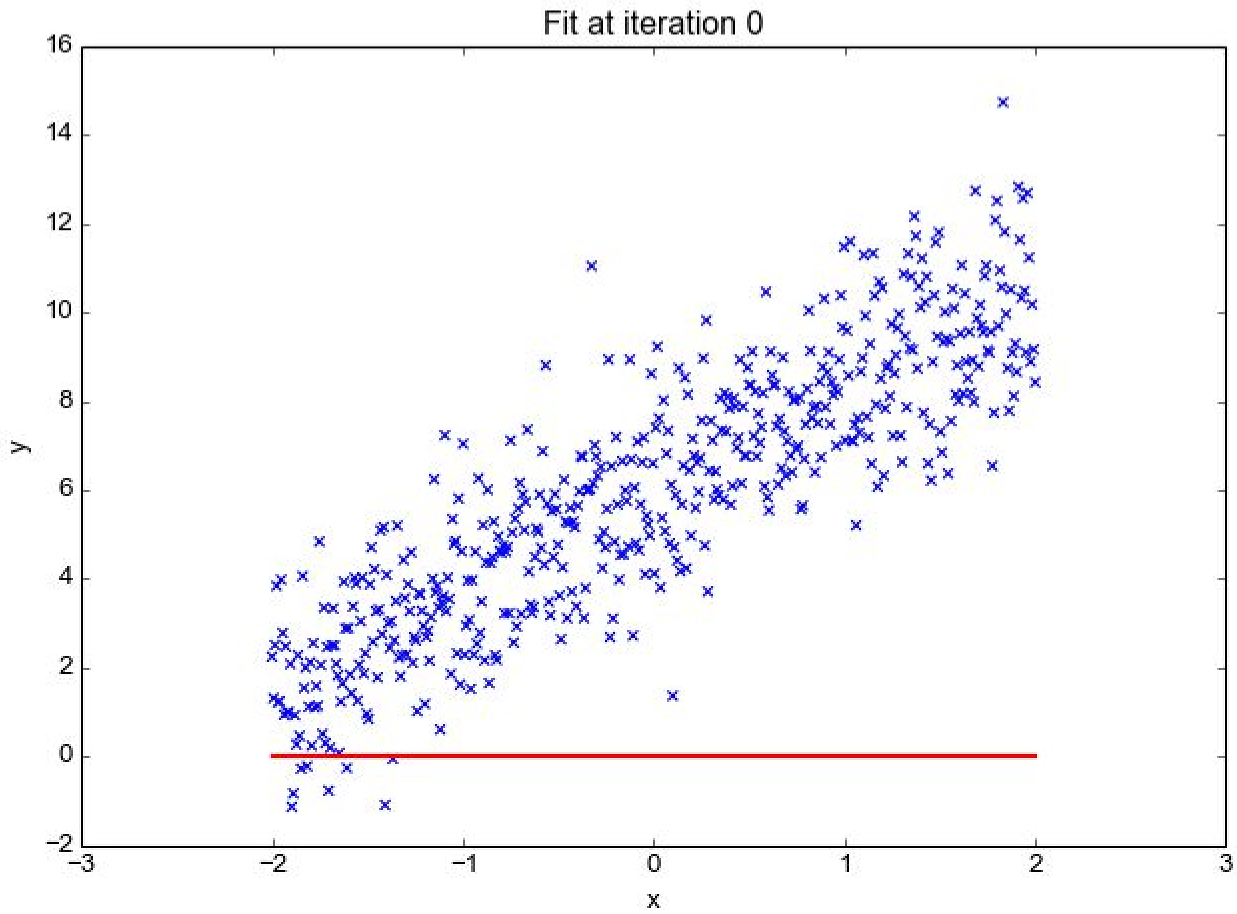
Online K-Means

Live tweeting at
#trevolive
@rawkintrevo



Linear Regression (Stochastic)

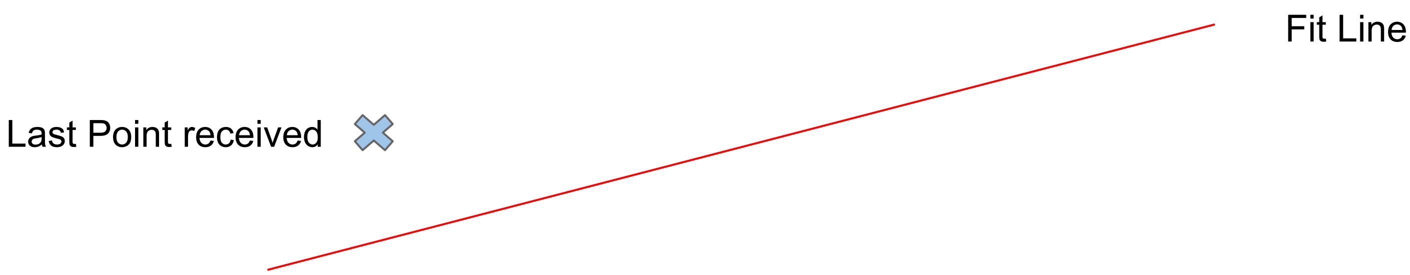
Live tweeting at
#trevolive
@rawkintrevo



http://eli.thegreenplace.net/images/2016/regressionfit.gif

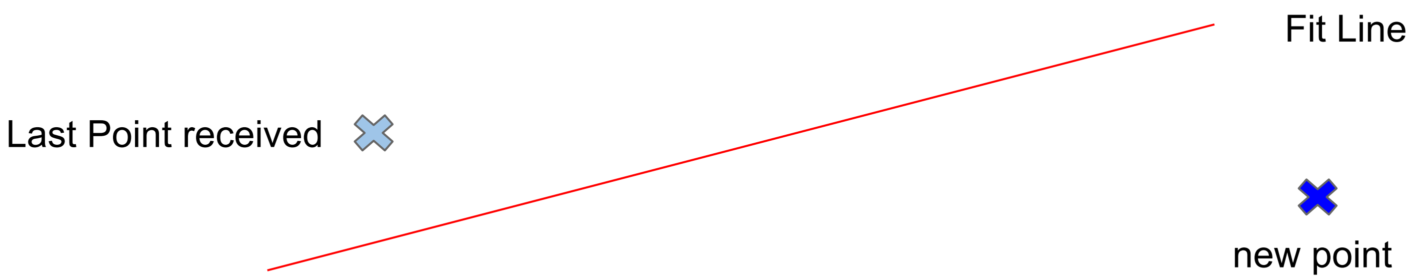
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



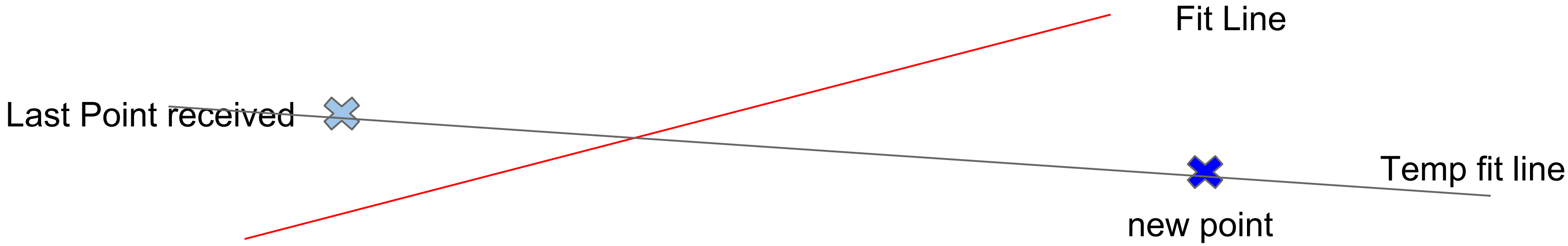
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



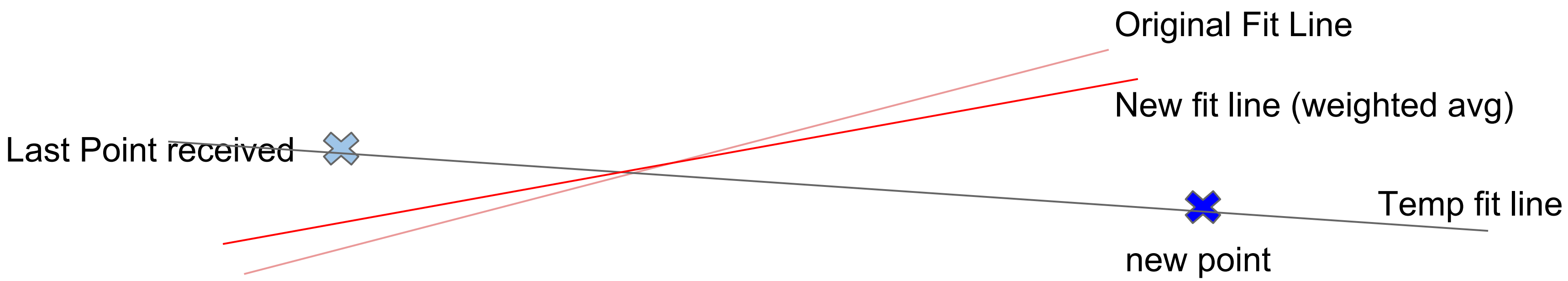
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



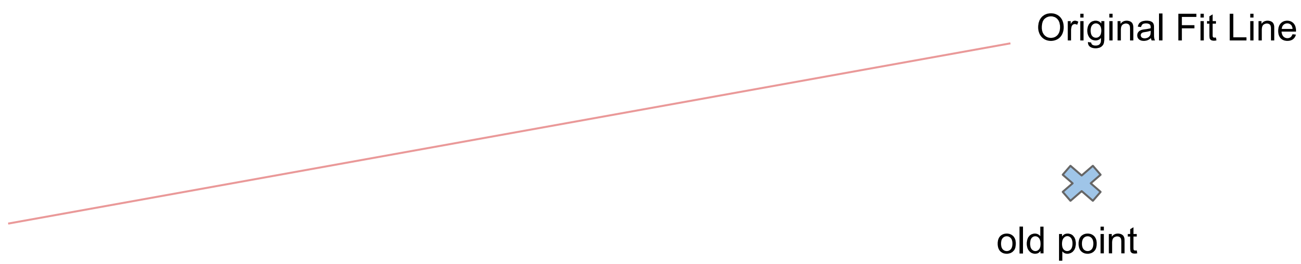
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



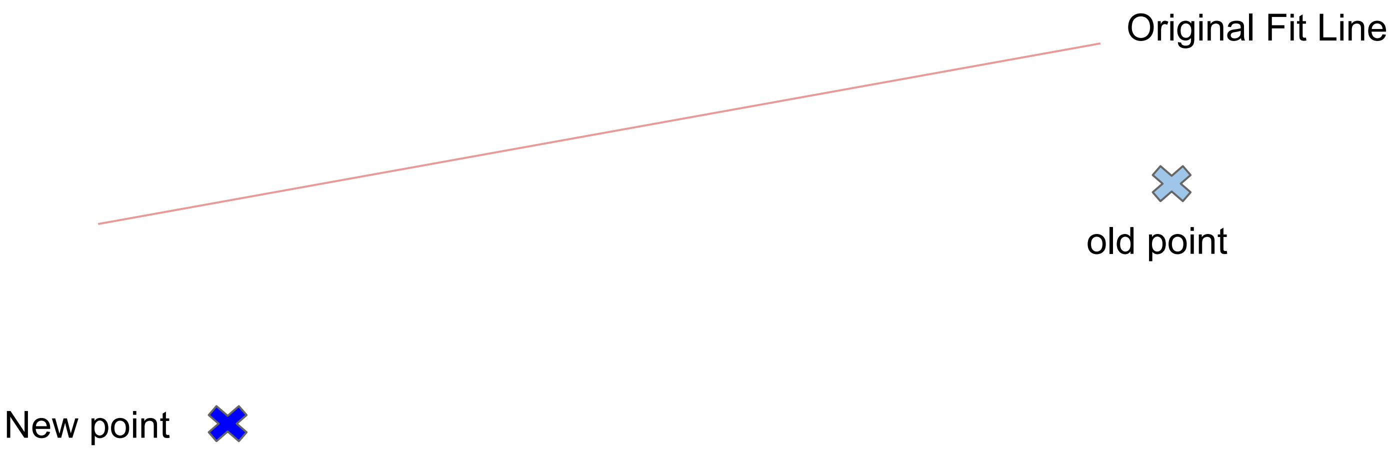
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



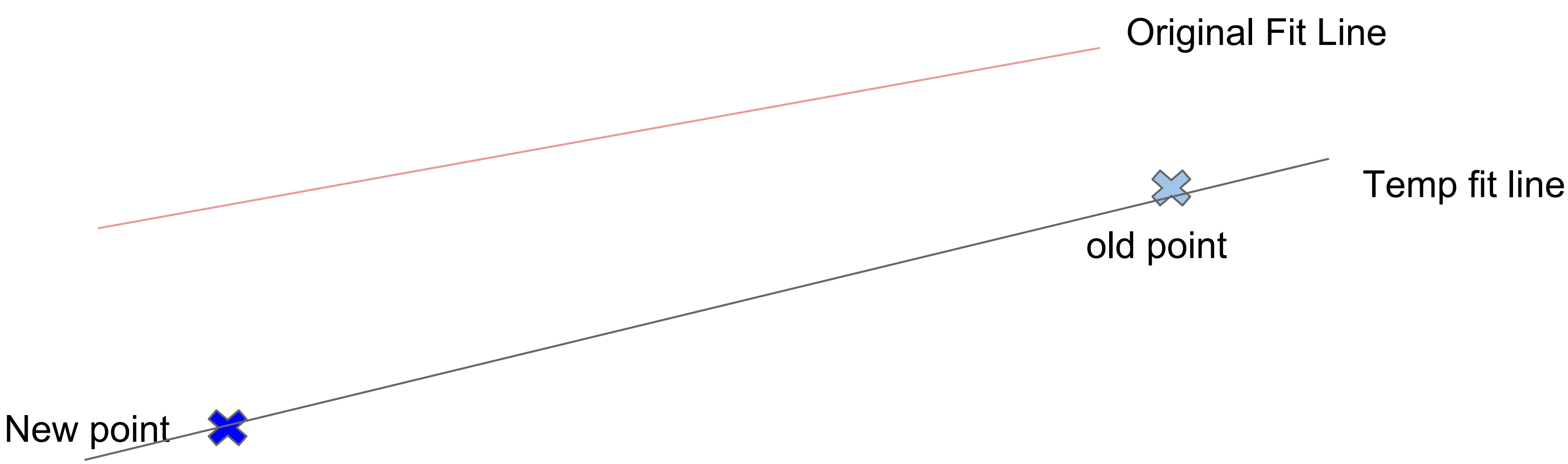
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



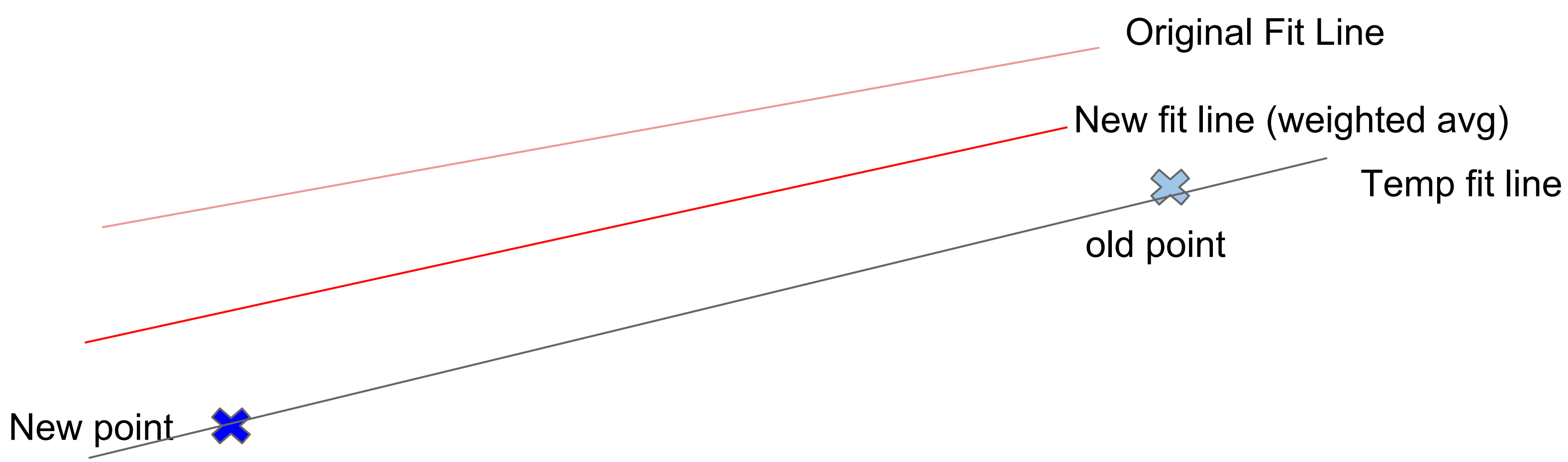
Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



Online Linear Regression

Live tweeting at
#trevolive
@rawkintrevo



Deep learning

Live tweeting at
#trevolive
@rawkintrevo

This would work on neural networks too.

Also “Deep Learning” is another buzz word.

Why?

Live tweeting at
#trevolive
@rawkintrevo

- Mostly Anomaly Detection (moving average, then something deviates)
  - A very popular use case of online/streaming algorithms (more talks today about this)
  - Algorithm learns what is normal (either online or offline)
  - When normality is sufficiently violated- the algorithm sounds an alarm
  - All anomaly detections some flavor of this. Usually referred to as: _______ Anomaly Detection, only to specify what algorithm was used for defining normality (or lack there-of).
  - <u>Architecture:</u> online-offline training choices depend primarily on how fast 'normality' changes in your specific use case

Table of Contents

#trevolive

- Adversarial Analysis
- Scoring in Real Time (how do you know you're right?)
- A/B Tests
- Multi-Arm Bandits

Learning in real-time with supervised methods (challenge)

Live tweeting at
#trevolive
@rawkintrevo

CHALLENGE:

How do you know how far you ‘missed’ prediction? In real life ‘correct’ answers may arrive later.

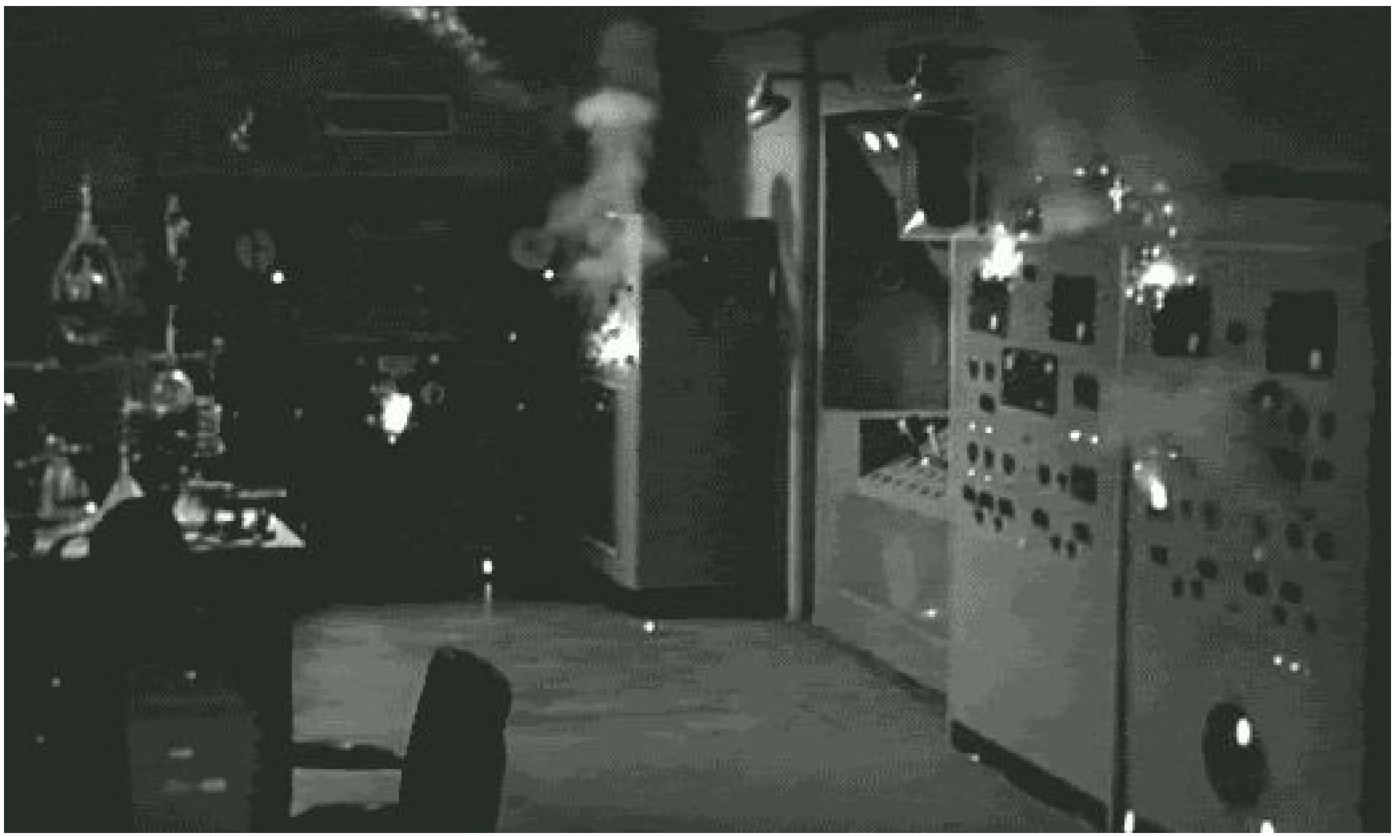
Corollary: If you have ‘correct’ answer why are you trying to predict it?

Not insurmountable, but prevents ‘one size fits all’ approaches (context dependence).

Latency and Normal Streaming Problems

Live tweeting at
#trevolive
@rawkintrevo

You've only got so much hardware.



Adversarial Analysis

Live tweeting at
#trevolive
@rawkintrevo

Simple Adversary-
How well does the algorithm do against “offline” version?

Consider Linear Regression with SGD
- Offline algorithm gets over full data set, then predicts
- Online model gets single pass to train and predict

How much worse is online than offline?

A/B Tests

Live tweeting at
#trevolive
@rawkintrevo

Online algos are often *interacting* with the environment.

Learning rates, other knobs.

Multi Arm Bandits

Live tweeting at
#trevolive
@rawkintrevo

A type of A/B test, sort of.

A/B test typically totally randomized

10% of the time you split records between two models (exploration)

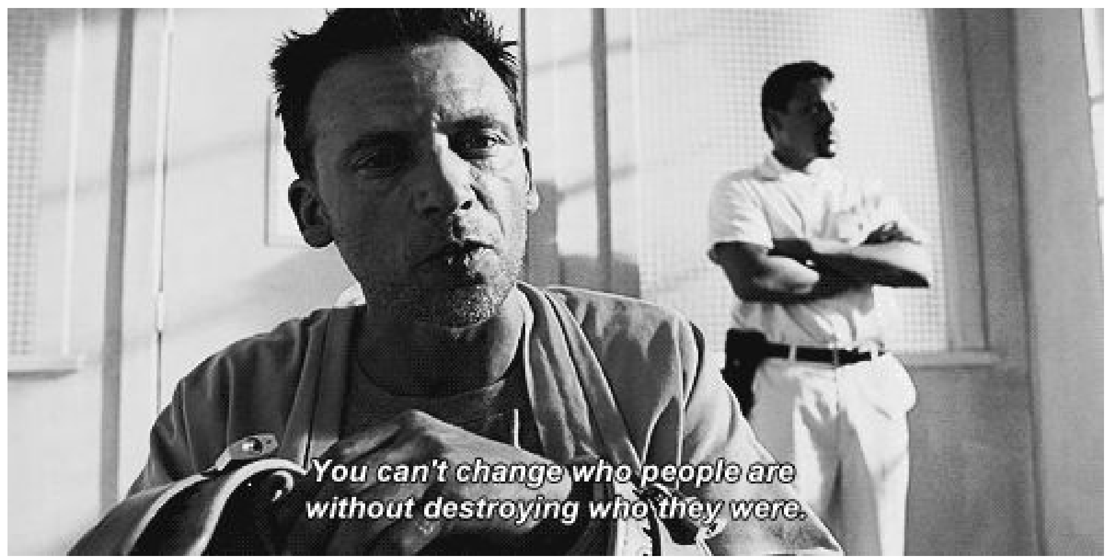
90% of the time you use the model which has been performing best in exploration (exploitation)

Key points

Live tweeting at
#trevolive
@rawkintrevo

Your algorithm is part of the environment is an actor in the environment which it is operating.

History is forever changed because of your algorithm



You can “back test” algorithms in the same way you could back test moves in a game of chess you played last week. (You can’t)

Live tweeting at
#trevolive
@rawkintrevo

E.g. How do you know if your recommendation engine made the best recommendations? You never will.

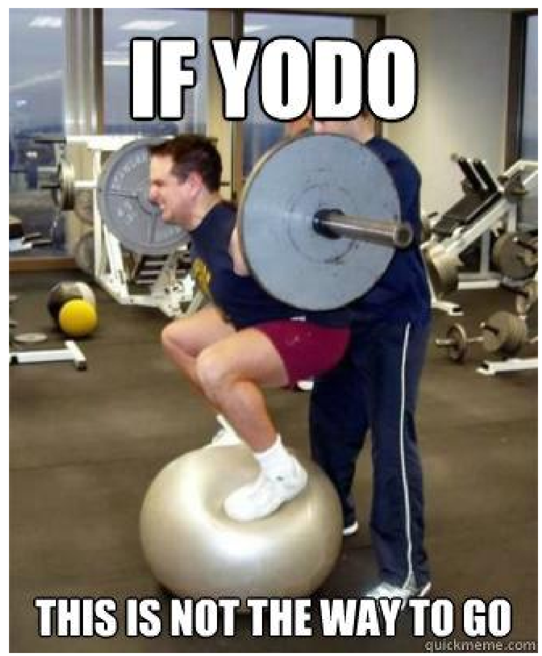
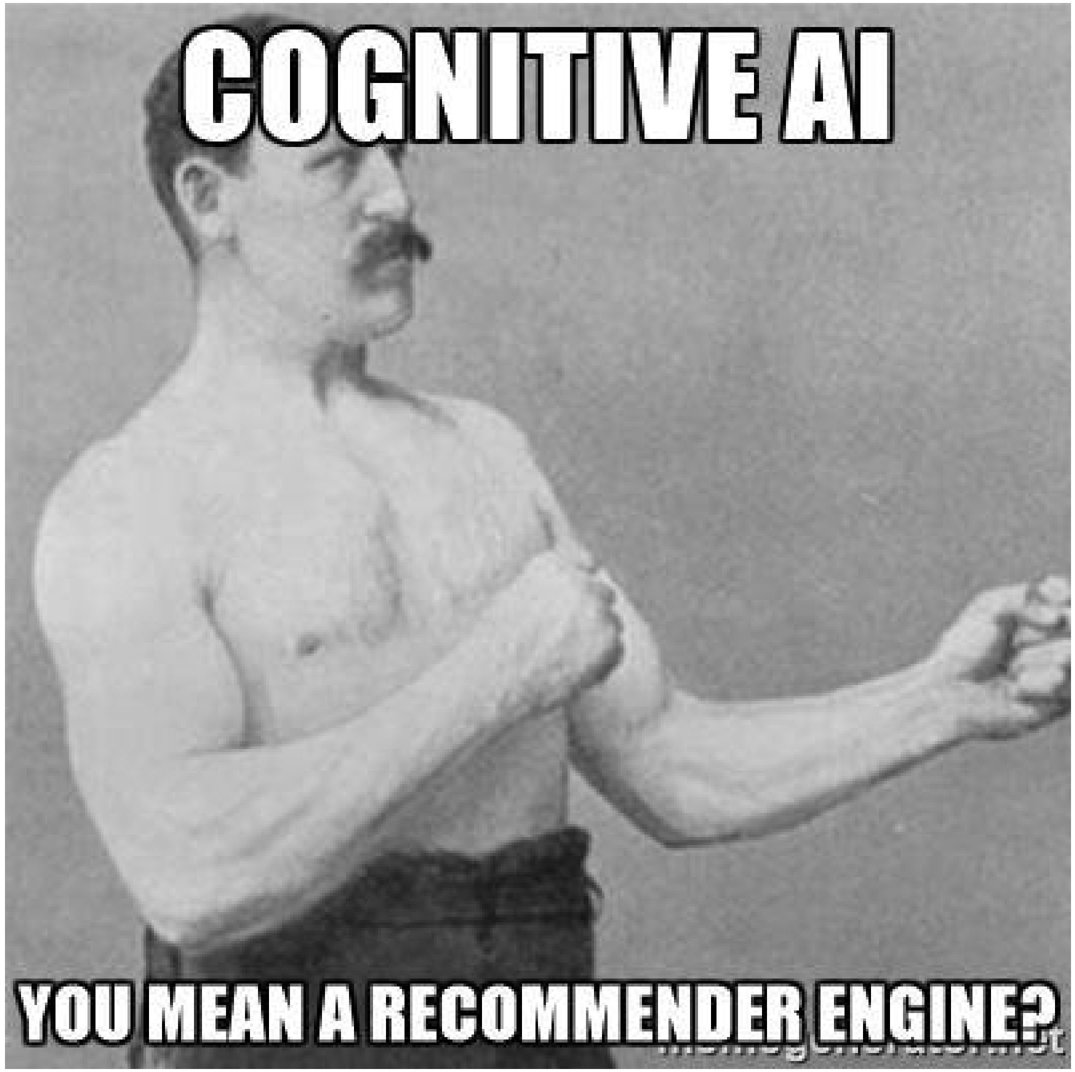


Table of Contents

#trevolive

- Correlated Co Occurrence – Brief Primer
- Architecture Overview
- Code walk through
- Looking at (pointless) results.

Live tweeting at
#trevolive
@rawkintrevo



Apache Mahout
tweet this @apachemahout

#trevolive

We've been here for years

Correlated Co Occurrence Recommender: Overview / Benefits

Live tweeting at
#trevolive
@rawkintrevo

- Overview of CCO
  - Collaborative Filtering (Like ALS, etc.)
  - Behavior Based (also like ALS)
  - Uses co-occurrence (no matrix factorization, unlike ALS)
  - Multi-modal: more than one behavior considered (unlike ALS / CO)

- Benefits of CCO
  - Many types of behaviors can be considered at once
  - Can make recommendations for users never seen before.

CCO Math
A Simple Co-Occurrence Recommender

Live tweeting at
#trevolive
@rawkintrevo

-

$$r = [P^T P] h_p$$

- r – recommendations
- P – history of all users on primary action (e.g. purchases)
  – Rows: user,
  – Columns: “Action” – e.g.(product1, product2, product3)
  – Then Row: Trevor, column: prodcuct2 => Trevor bought product 2
- [PtP] – Log Likelihood based correlation test
- hp- A user’s history on behavior p (could be new user)

CCO Math
Correlated Co-Occurrence Recommender

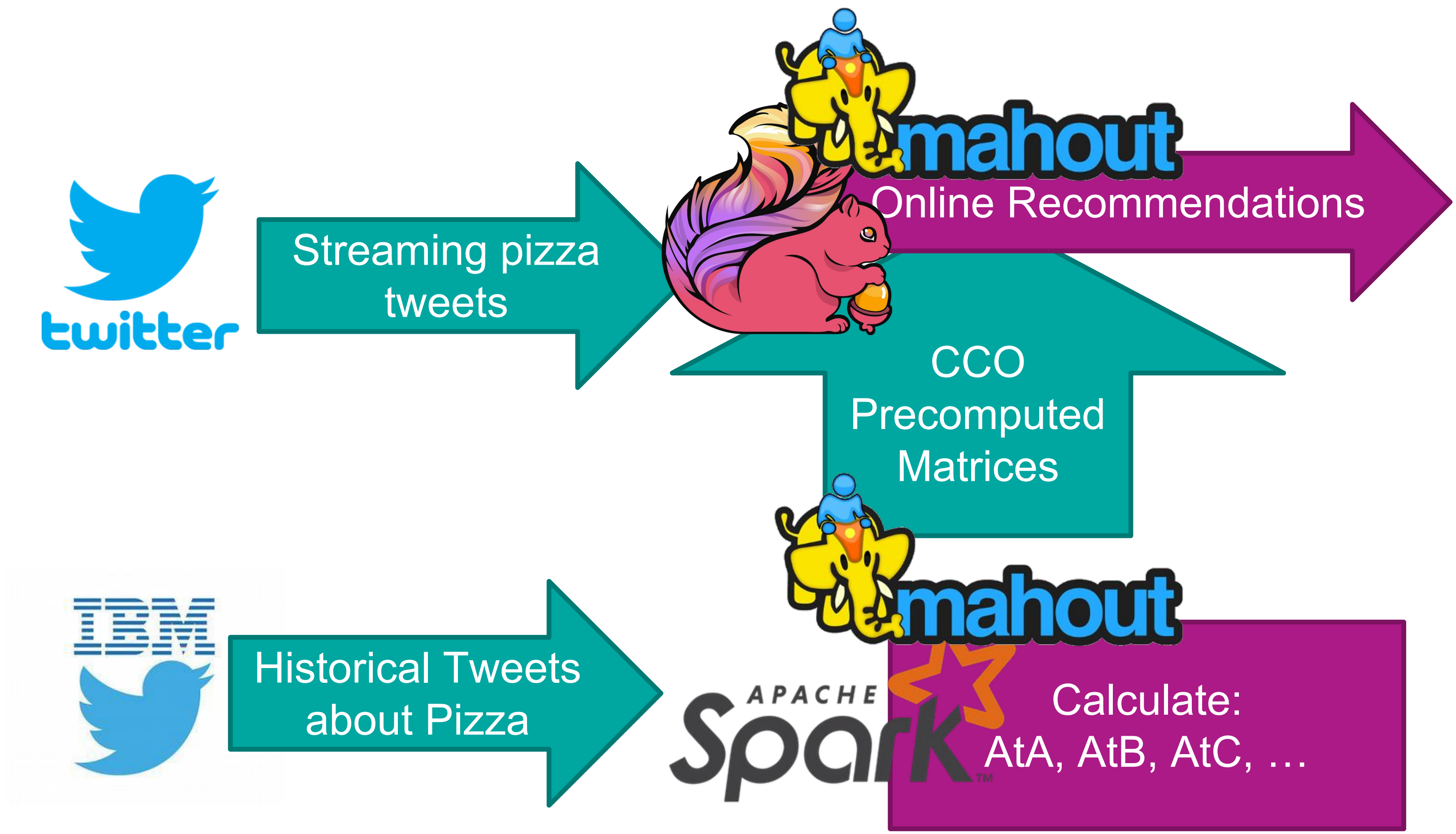
Live tweeting at
#trevolive
@rawkintrevo

$$r = [P^T P]h_p + [P^T A]h_a + [P^T B]h_b + \ldots$$

- r – recommendations
- P – history of all users on primary action (e.g. purchases)
- [PtP] – Log Likelihood based correlation test
- A – history of all users on secondary action
  - Must have some rows (e.g. users)
- B – history of all users on tertiary action
  - Must have some rows (e.g. users)
- hp- A user's history on behavior p (could be new user)

Architecture: Lambda CCO (Logo soup)

Live tweeting at
#trevolive
@rawkintrevo



Python pulled in historical tweets and did this UserID - HashTag

Live tweeting at
#trevolive
@rawkintrevo

561918328478785536,None
561918357851897858,None
561909179716481024,pizzagate
561909179716481024,gamergate
561949040011931649,None
561948991777038336,None
561947869805285377,superbowl
561947869805285377,pizzapizza
561918920282476545,None
561926796778565632,gunfriendly
561927577351503873,None

Python pulled in historical tweets and did this UserID - Words

Live tweeting at
#trevolive
@rawkintrevo

561684486380068865,savethem000
561684486380068865,i
561684486380068865,dunno
561684486380068865,smiles
561684486380068865,want
561684486380068865,to
561684486380068865,get
561684486380068865,some
561684486380068865,pizza
561684486380068865,or
561684486380068865,something
561684441526194176,pizza
561684441526194176,de
561684441526194176,queso
561684441526194176,lista
561684441526194176,para

Some Spark Code

Live tweeting at
#trevolive
@rawkintrevo

```scala
import org.apache.mahout.sparkbindings.indexeddataset.IndexedDatasetSpark
import org.apache.mahout.math.cf.SimilarityAnalysis

val baseDir = "/home/rawkintrevo/gits/ffsf17-twitter-recos/data"
// We need to turn our raw text files into RDD[(String, String)]

val userFriendsRDD = sc.textFile(baseDir + "/user-friends.csv")
 .map(line => line.split(",")).filter(_.length == 2).map(a => (a(0), a(1)))
val userFriendsIDS = IndexedDatasetSpark.apply(userFriendsRDD)(sc)

val userHashtagsRDD = sc.textFile(baseDir + "/user-ht.csv")
 .map(line => line.split(",")).filter(_.length == 2).map(a => (a(0), a(1)))
val userHashtagsIDS = IndexedDatasetSpark.apply(userHashtagsRDD)(sc)

val userWordsRDD = sc.textFile(baseDir + "/user-words.csv")
 .map(line => line.split(",")).filter(_.length == 2).map(a => (a(0), a(1)))
val userWordsIDS = IndexedDatasetSpark.apply(userWordsRDD)(sc)

val hashtagReccosLlrDrmListByUser = SimilarityAnalysis.cooccurrencesIDSs(
 Array(userHashtagsIDS, userWordsIDS, userFriendsIDS),
 maxInterestingItemsPerThing = 100,
 maxNumInteractions = 500,
 randomSeed = 1234)
```

CCO Math
Spark+Mahout just Calculated these:

Live tweeting at
#trevolive
@rawkintrevo

$$r = \boxed{[P^T P]} h_p + \boxed{[P^T A]} h_a + \boxed{[P^T B]} h_b + \ldots$$

- r – recommendations
- P – history of all users on primary action (e.g. purchases)
- [PtP] – Log Likelihood based correlation test
- A – history of all users on secondary action
  - Must have some rows (e.g. users)
- B – history of all users on tertiary action
  - Must have some rows (e.g. users)
- hp- A user's history on behavior p (could be new user)

Some Flink Code

Live tweeting at
#trevolive
@rawkintrevo

```
streamSource.map(jsonString => {
 val result = JSON.parseFull(jsonString)

 val output = result match {
  case Some(e) => {
   /*****************************************************************************************
    * Some pretty lazy tweet handling
   val tweet: Map[String, Any] = e.asInstanceOf[Map[String, Any]]
   val text: String = tweet("text").asInstanceOf[String]
   val words: Array[String] = text.split("\\s+").map(word => word.replaceAll("[^A-Za-z0-9]", "").toLowerCase())

   val entities = tweet("entities").asInstanceOf[Map[String, List[Map[String, String]]]]
   val hashtags: List[String] = entities("hashtags").toArray.map(m => m.getOrElse("text","").toLowerCase()).toList
   val mentions: List[String] = entities("user_mentions").toArray.map(m => m.getOrElse("id_str", "")).toList

   /****************************************************************************************
    * Mahout CCO
   val hashtagsMat = sparse(hashtagsProtoMat.map(m => svec(m, cardinality = hashtagsBiDict.size)):_*)
   val wordsMat = sparse(wordsProtoMat.map(m => svec(m, cardinality= wordsBiDict.size)):_*)
   val friendsMat = sparse(friendsProtoMat.map(m => svec(m, cardinality = friendsBiDict.size)):_*)

   val userWordsVec = listOfStringsToSVec(words.toList, wordsBiDict)
   val userHashtagsVec = listOfStringsToSVec(hashtags, hashtagsBiDict)
   val userMentionsVec = listOfStringsToSVec(mentions, friendsBiDict)

   val reccos = hashtagsMat %*% userHashtagsVec + wordsMat %*% userWordsVec + friendsMat %*% userMentionsVec

   /*****************************************************************************************
    * Sort and Pretty Print
```

CCO Math
Flink+Mahout just Calculated these:

Live tweeting at
#trevolive
@rawkintrevo

$$r = [P^T P]\boxed{h_p} + [P^T A]\boxed{h_a} + [P^T B]\boxed{h_b} + \ldots$$

- r – recommendations
- P – history of all users on primary action (e.g. purchases)
- [PtP] – Log Likelihood based correlation test
- A – history of all users on secondary action
  - Must have some rows (e.g. users)
- B – history of all users on tertiary action
  - Must have some rows (e.g. users)
- hp- A user's history on behavior p (could be new user)

Tweets

Live tweeting at
#trevolive
@rawkintrevo

text: joemalicki josephchmura well i can make a pizza i bet he cant so there
userWordsVec: so a well i there can he make pizza cant
hashtags used: List()
hashtags reccomended:
(**ruinafriendshipin5words** : 13.941270843461098)
(**worstdayin4words** : 8.93444123705558)
(recipes : 8.423061768672596)

text: people people dipping pizza in milk im done
userWordsVec: people in im done pizza
hashtags used: List()
hashtags reccomended:
(None : 18.560367273335828)
(**vegan** : 10.84782189800353)
(fromscratch : 10.84782189800353)

*Results were cherry picked- no preprocessing, this was a garbage in-garbage out algo for illustration purposes only.

Buzzword Soup

Live tweeting at
#trevolive
@rawkintrevo

Cazognitive AI

Hybrid Lambda Architecture

Online Recommendations

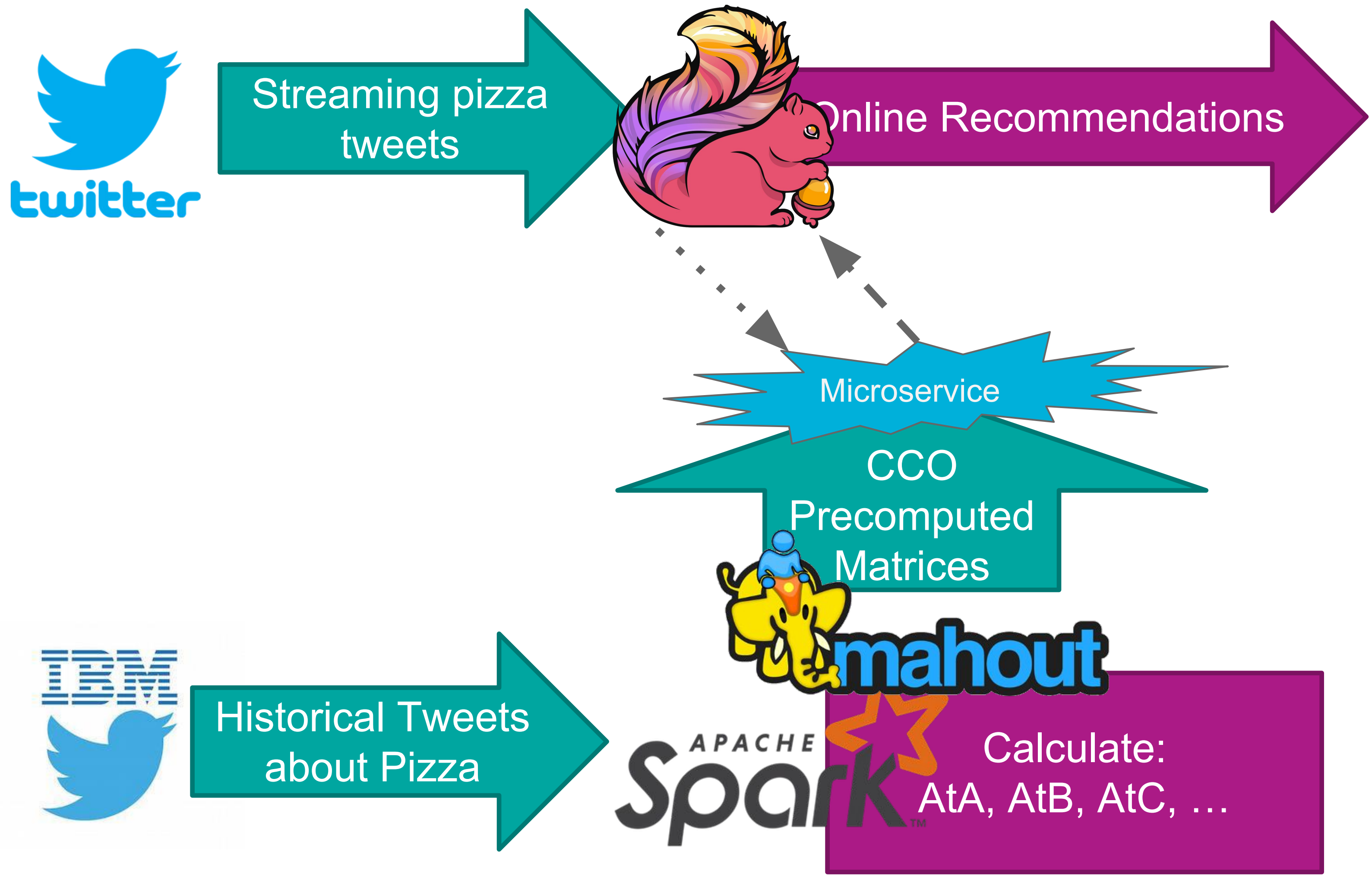
GPU Accelerated

Adversarial

Algorithm

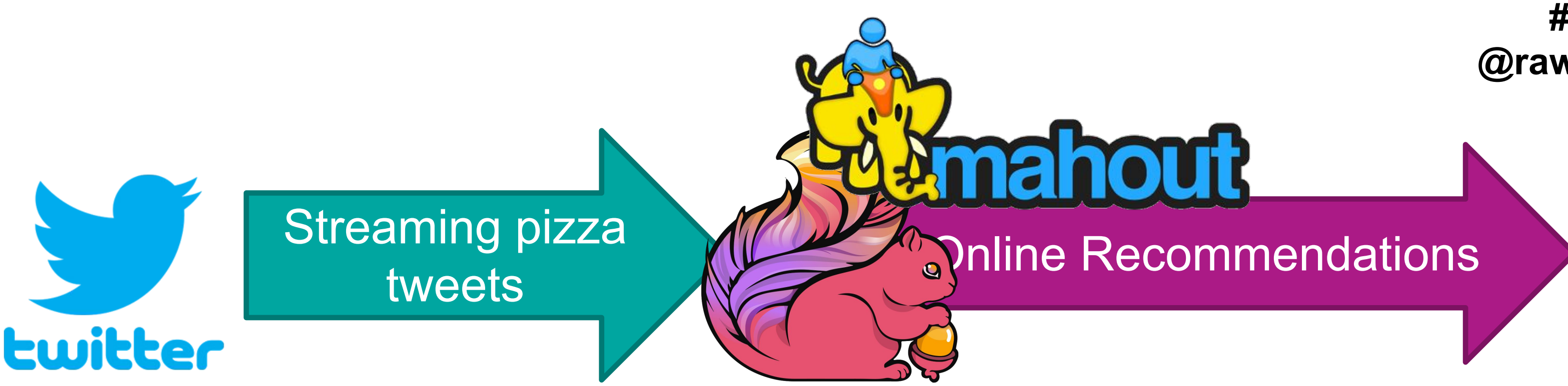
Architecture: Lambda CCO (Micro architecture)

Live tweeting at
#trevolive
@rawkintrevo



Architecture: Kappa Architecture (Neural Nets)

Live tweeting at
#trevolive
@rawkintrevo



Auto Encoder "Deep Learning"-
Input- Tweet
Output - Same Tweet

If error too high, user is maybe drunk- ask if they are SURE they want to tweet (or maybe spell check or go to sleep).

**Apache Mahout Deep Learning [WIP] Are Online Neural Nets by Default.*

Are you sure you want to tweet that?

Live tweeting at
#trevolive
@rawkintrevo

Table of Contents

#trevolive

- Trevor attempts to tie everything together into a cohesive thought
- Audience members asks easy questions
- Audience members buy speaker beer at after party

Final Thoughts

Live tweeting at
#trevolive
@rawkintrevo

A lot of buzzwords have been flying around especially with respect to machine learning and streaming.

- Online
- Lambda / Kappa architecture
- Streaming machine learning
- Real time predictive model
- machine learning
- artificial/machine/cognitive intelligence
- cognitive
- blah- ^^ pick 2.

Final Thoughts

Live tweeting at
#trevolive
@rawkintrevo

Now that you've sat through this talk hopefully you can:

1. Call people out for trying to make their product/service/open source project/startup sound like a bigger deal than it is
2. Church up your product/service/open source project/startup to get clients/VC dummies excited about it without *technically* lying

Unsolicited Sales Pitches

What Silicon Valley Conferences are all about

Software Freedom Conservancy

Live tweeting at
#trevolive
@rawkintrevo

Legal Guardians of Open Source

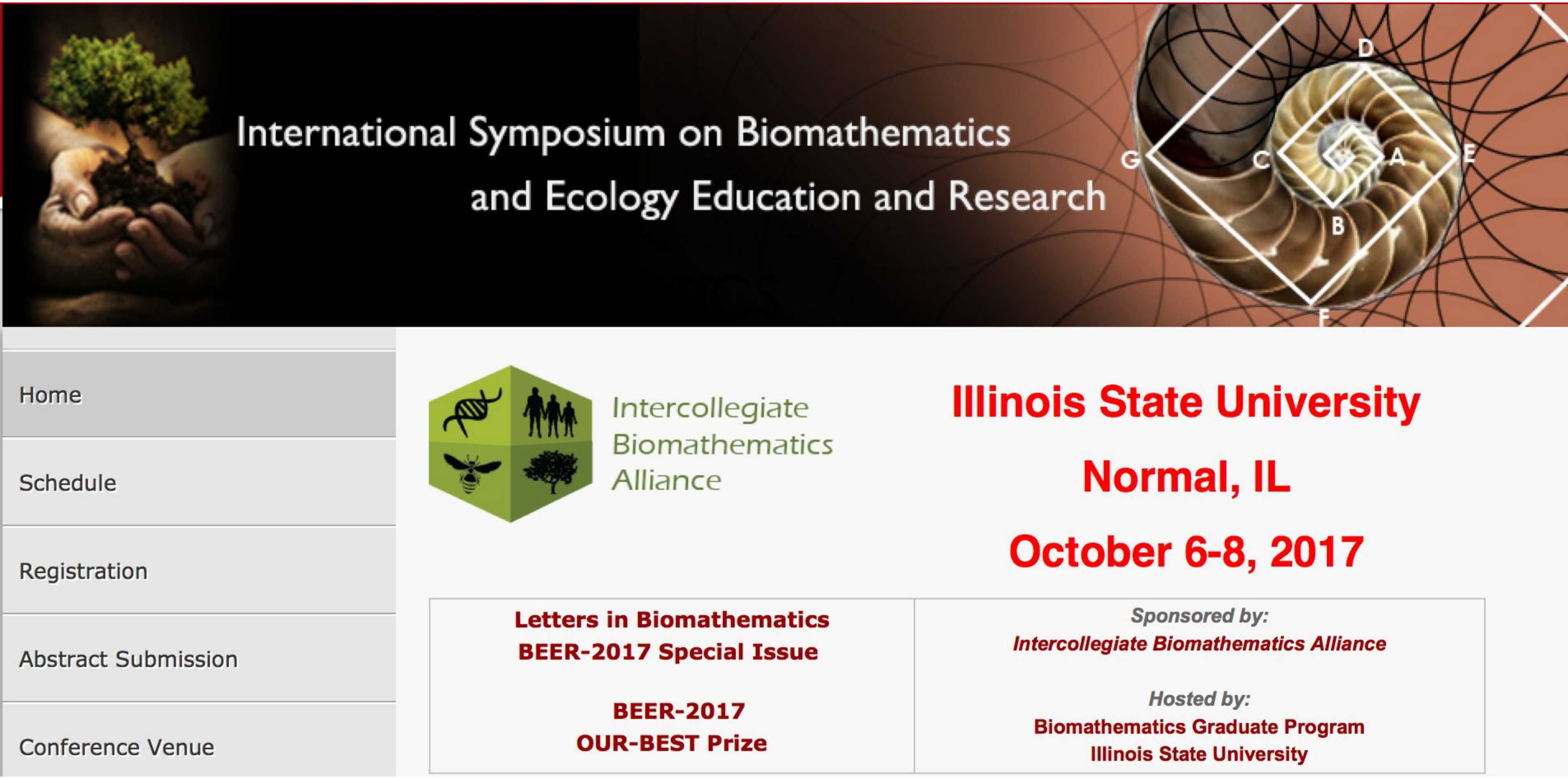
Please tweet a photo of this @conservancy

ChicagoVets.org/vethub
@chicagovets

Live tweeting at
#trevolive
@rawkintrevo

10th Annual
Biostats, Ecology, Education, and Research

Live tweeting at
#trevolive
@rawkintrevo



Questions?

Buy trevor beers.

https://github.com/rawkintrevo/fsf17-twitter-recos