Machine learning using more data than is healthy: Streaming

Miles Osborne

July 2013

Streaming

Many problems involve dealing with unlimited quantities of data:

- Financial trades
  - Visa reported (peak) 11k transactions per second (2011).
- Social Media
  - Twitter processes (peak) more than 7k Tweets per second.
- Using the Web for Machine Translation
- Etc etc

Batch processing doesn't work –we need to deal with a stream

Streaming

Do not panic (too much)

Streaming: Implications

Space requirements:

- Streams never end and often encode an unbounded number of types.
  - A type is some distinct item.
    - New names, places, spelling mistakes etc.
  - A token is an instance of a type.
- In text processing this is called Heap's Law.
- A model that encodes all of the stream might have unbounded size

We need to use constant space.

Streaming: Implications

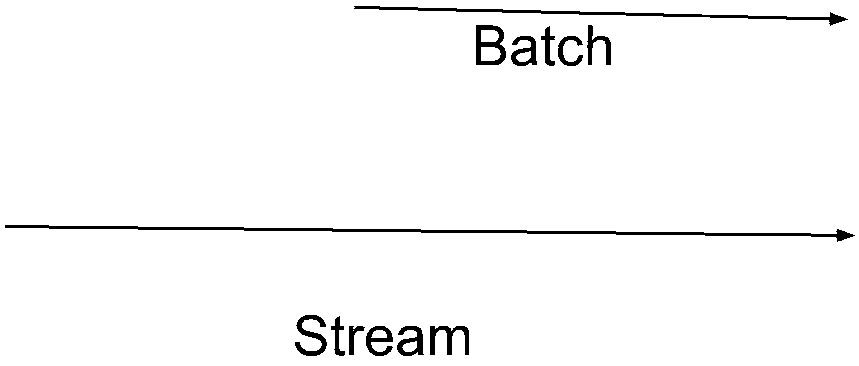
Time requirements:

- ML algorithms typically make multiple passes over the data.
  - Often IO or network bandwidth is the dominant computational cost.
- Processing time should not get slower and slower as we see more and more data.

We need to take constant time.

Streaming: Approach 1

Truncate the stream as a fixed-sized window and treat it as a batch:



Streaming: Approach 1

Truncate the stream as a fixed-sized window and treat it as a batch.

- A fixed sized batch can be processed using standard techniques.
- This approach is commonly used.
- Unclear when to truncate.
- Need to rebuild the model from scratch each time we see new data (incremental learning??).

Streaming: Approach 2

Represent as much of the stream as possible:

- Extend randomised methods to deal with a stream.
  - Randomised methods can encode more of the data than exact methods.
- Support constant time and space operations.
- Where possible support incremental processing.

Class Structure

- Streaming techniques.
- Streaming infrastructure.
- Case study –event detection in Social Media.

Streaming Techniques

CM sketch:

- It summarises the whole stream.
- Uses constant space and takes constant time.
  - Errors: false positives, wrong estimates.
  - No false negatives (returning a zero for some item in the stream).
- The error rate will increase as the stream unfolds unless items are removed.

Items can be deleted by **subtracting** counts.

Reservoir Sampling

Another way to manage a stream is sampling from it:

- A sample is a managable chunk of data.
- A sample will not have false positive errors.
  - Samples can have false negatives –not including items in the data.
- This sample might be representative (but perhaps we want to bias it).

A model produced from a sample might well approximate the stream.

Reservoir Sampling

Biased sampling is easy:

- Just use the last *n* items.
- This puts a lot of trust in the most recent items

What if we want an *unbiased* sample?

- An unbiased sample would represent the full stream until now.

Reservoir Sampling

If we knew the size of the stream ahead of time:

- Randomly select an item.
- Add it to the sample.

But usually we can only see the stream once and we don’t know how big it is.

Reservoir Sampling

How it works:

- Each time we see a new item, assign a random number to it.
- If that random number is higher than any previously generated random number, add the item to the sample.
- If we have reached the maximum number of items in our sample, remove the item with the lowest random number.

At any stage when processing the stream we can use the data in the sample to build an unbiased model.

Reservoir Sampling

Cloudera use RS:

- Used for finding representative examples for K-means.

http://blog.cloudera.com/blog/2013/04/hadoop-stratified-randosampling-algorithm/

Bloom Filters

BF:

- As we add more and more items, the error rate will increase.
- Items cannot be deleted from a BF:
  - Removing one item may delete other items.
- BFs cannot be resized without reinserting all items:
  - The hash functions will change as they contain the table size.

We need to modify a BF to make it stream-friendly.

Bloom Filters

Dynamic BF:

- Supports insertions and deletions.
- Takes more space than a standard BF.
- Has false positives and possibly false negatives.

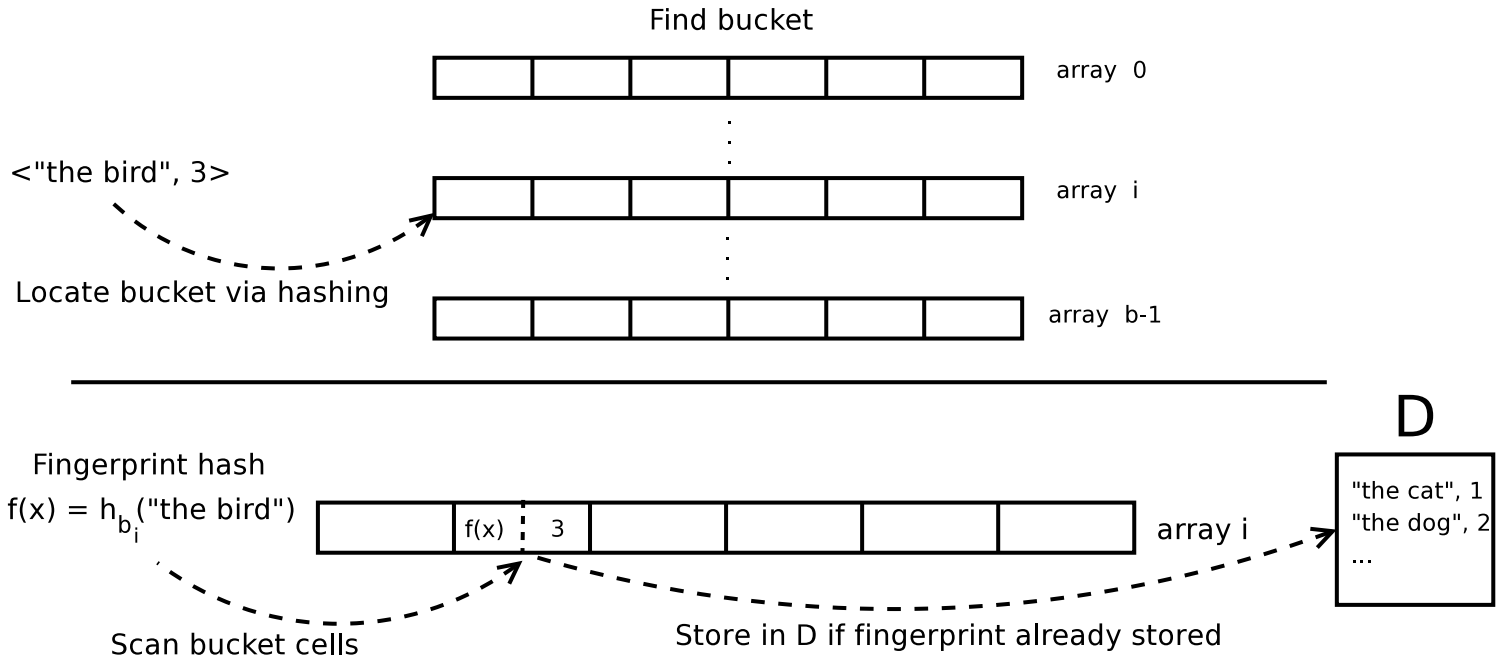
Bloom Filters

Dynamic BF:

- Two part data-structure
- First part: a standard Bloomier Filter.
  - A Bloomier Filter is word-based and stores fingerprints.
- Second part: an exact overflow dictionary.

Bloom Filters

Dynamic BF:



Bloom Filters

To insert a K-V pair:

- Locate entry and compare finger prints.
  - If not present, add to the BF.
  - (Collision) otherwise add to the Dictionary.

Bloom Filters

To recover a K-V pair::

- Check if the pair is in the dictionary.
- If yes, return it. (No errors)
- Otherwise check the BF and return the value (Possible errors).

Bloom Filters

To delete a K-V pair::

- Locate the K-V pair.
- Set that entry to a null value.

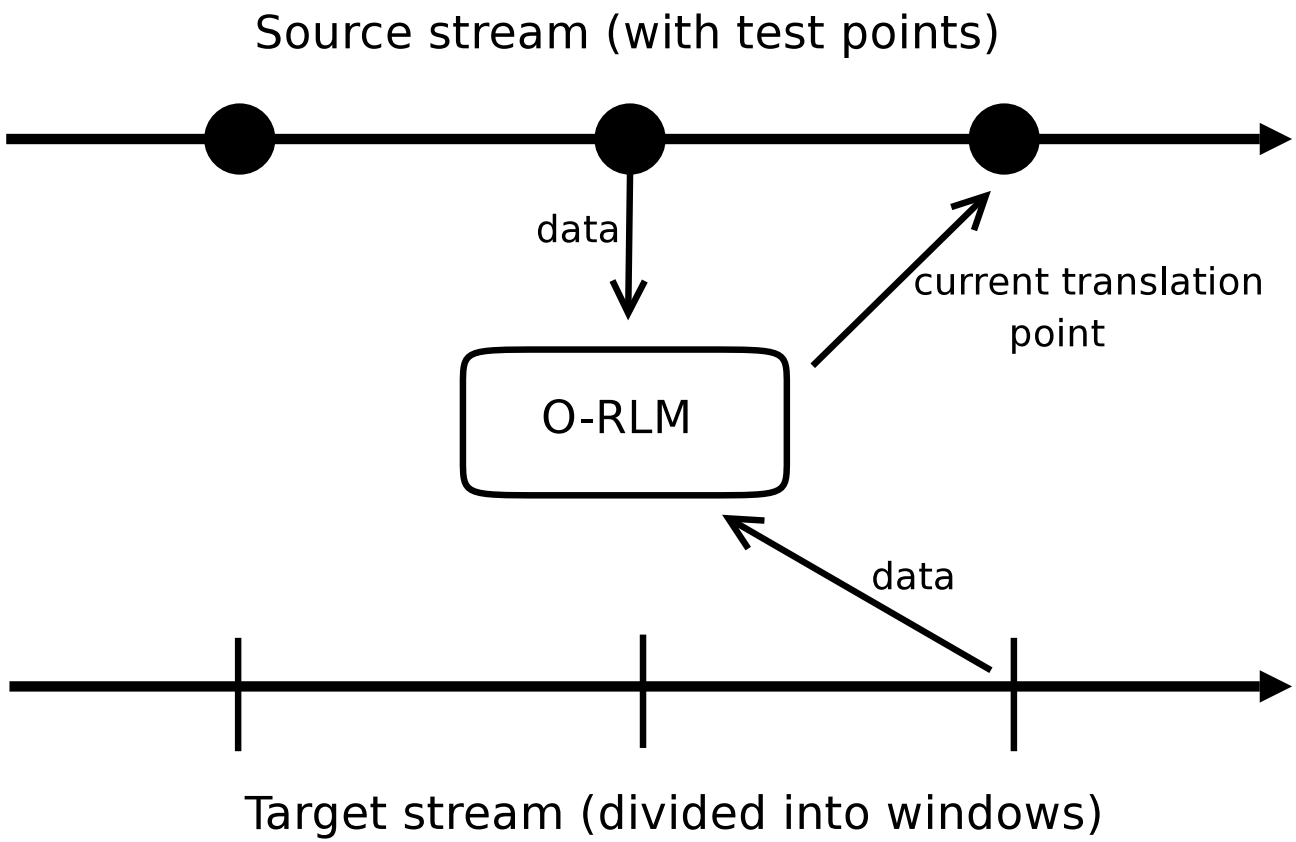
Deleting a K-V pair will not affect any other items.

Bloom Filters

Dynamic BF Example:

- Maintain a language model over streaming newswire.
- The LM is represented as a D-BF.
- Incrementaly retraining. Constant space.

Using a Dynamic RLM



Bloom Filters

LM	RAM
Exact	7450MB
Bloom	390MB
G LM	640MB
D-RLM	705MB

- G-LM is the Google Perfect Hashing LM (batch-based).
- Supporting insertions and deletions makes the LM less space efficient.

Bloom Filters

	Naive G-LM		Batch Retrained G-LM		D-RLM	
Test Date	200MB	300MB	200MB	300MB	200MB	300MB
Jan	35.94	37.12	35.94	37.12	36.44	37.17
Apr	33.55	35.79	36.01	35.99	35.87	36.10
Aug	22.44	26.07	28.97	29.38	29.00	29.18
Avg	30.64	32.99	33.64	34.16	33.77	34.15

- Naive approach: always storing the full stream as it evolves.
- The batch G-LM fully retrains from scratch.
- The stream-based D-RLM incrementally retrains.

Class Structure

- Streaming techniques.
- Streaming infrastructure.
- Case study –event detection in Social Media.

Storm

Streaming problems require streaming infrastructure:

- Hadoop (Map Reduce) is batch-based.
  - We need to wait for it to finish before we get any results.
- Hadoop can have unpredictable latency.

Storm

Open source distributed processing (Twitter)

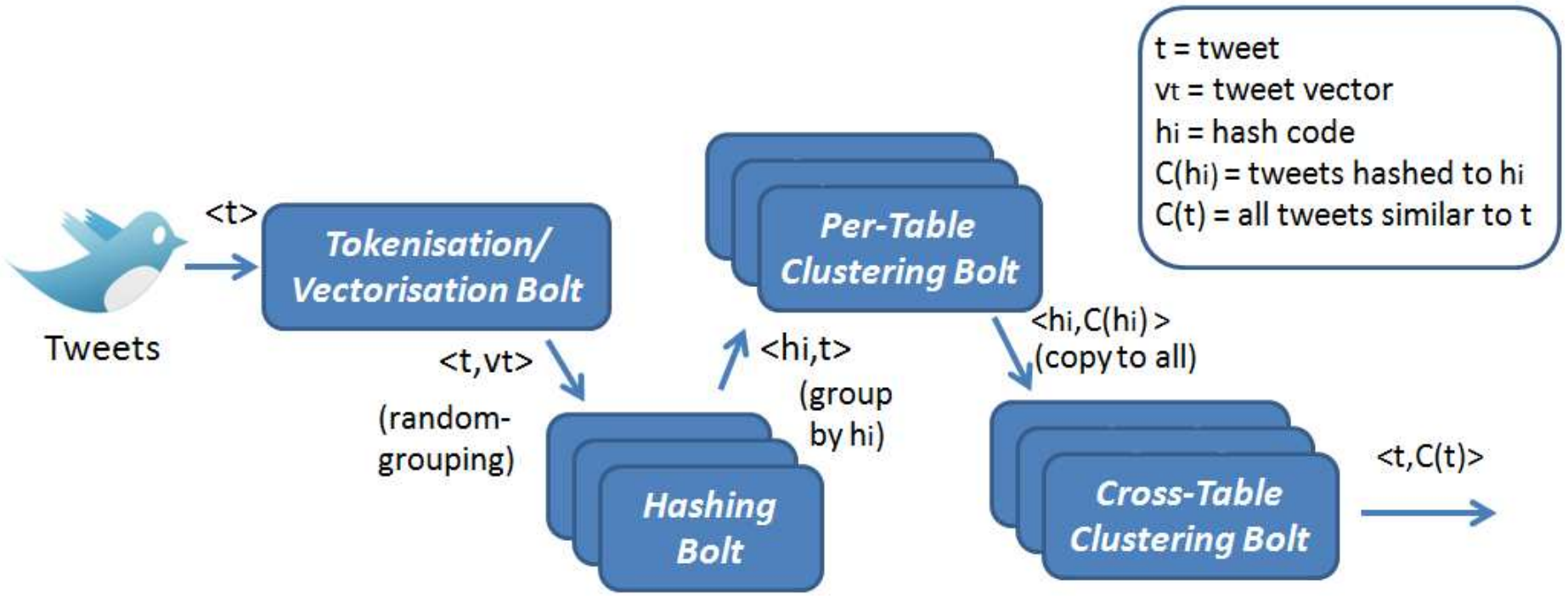
- ‘Real-time Hadoop’
- Low-latency.
- Suitable for incremental processing.

http://storm-project.net/

Storm

- Everything runs in-memory, across multiple machines.
- Data is injected into a *topology*.
  - A job is represented as a graph of communicating tasks
- Computation never ends.

Storm


t = tweet
vt = tweet vector
hi = hash code
C(hi) = tweets hashed to hi
C(t) = all tweets similar to t
Tweets
<t>
Tokenisation/ Vectorisation Bolt
<t,vt>
(random-grouping)
Hashing Bolt
<hi,t>
(group by hi)
Per-Table Clustering Bolt
<hi,C(hi)>
(copy to all)
Cross-Table Clustering Bolt
<t,C(t)>

Class Structure

- Streaming techniques.
- Streaming infrastructure.
- Case study –event detection in Social Media.

Event Detection in Social Media

Case study demonstrating streaming techniques

- LSH.
- Constant time, constant space.

The Task

- Given thousands of posts per second
  - 95%+ of which are garbage
- Find in real time all breaking news
  - And do it with no supervision

Twitter

Characteristics:

- Low signal-to-noise ratio
- Multilingual
  - About 65% of Tweets are in English
- > 400 million posts / day
- > 500 million registered users (2012)

Twitter

(Good) Examples:

- RT @ channel4news Scottish airspace will reopen from 7am tomorrow, air traffic control company Nats have confirmed. #ashtag LDN to follow?
- So I’m scheduled on the United flght out of LHR on Wed morning - what are the changes? #ashtag #ashcloud
- Still stuck in S Italy needing to get to Ireland anyone have a boat? #getmehome #ashtag

Sun CEO resignation

Today's my last day at Sun. I'll miss it. Seems only fitting to end on a #haiku. Financial crisis/Stalled too many customers/CEO no more

Events in Twitter

Twitter has been seen as a good source of (breaking) news:

- Real time updates
- Citizen journalism
  - Earthquakes, numerous natural disasters
  - Plane crashes, riots, . . .
  - Gossip
  - Etc etc
- Twitter acts as an aggregator of traditional news
- People react and comment on posts (sentiment)

Event Detection and Finance

Events in Twitter can have financial impact:

- (April 24 2013) Dow Jones dropped 150 points due to a fake tweet

Knowing when something is happening before others (eg reported in Reuters, Bloomberg) can give an edge:

- (Energy Sector) Forest fires
- Long tail events *not reported* in Newswire?

Unlike in Newswre events are not clearly marked

Finance-related Events Detected

1. Update: Dow Jones industrial average down more than 500 points hortly before closing - AP
2. Reuters FLASH: Appeals court rules that Obama's healthcare law's individual mandate to own health insurance unconstitutional
3. RT @BreakingNews: Obama says sanctions will prohibit US people from operating or investing in Syria - Reuters
4. RT @jaketapper: BREAKING > Govt official tells ABC News US expect S&P downgrade > http: . . . #BreakingNews
5. Europe Stocks Extend Recovery Rally (New York Times) http:/ . . .

Event Detection

Find breaking news as quickly as possible:

- Earthquake happens on Monday at 9am.
- Report story as soon after 9am as possible,
- Don’t report follow-up mentions.

Intensively studied as part of *Topic Detection and Tracking* (DARPA TIDES programme, 1997 – 2004)

First Story Detection

Typical FSD system:

- Map stories into a vector representation:
  - Each term is a component in a vector.
  - Components are weighted.

Hello how are you $\rightarrow (0.1, 0.0, 0.8, \ldots)$

First Story Detection

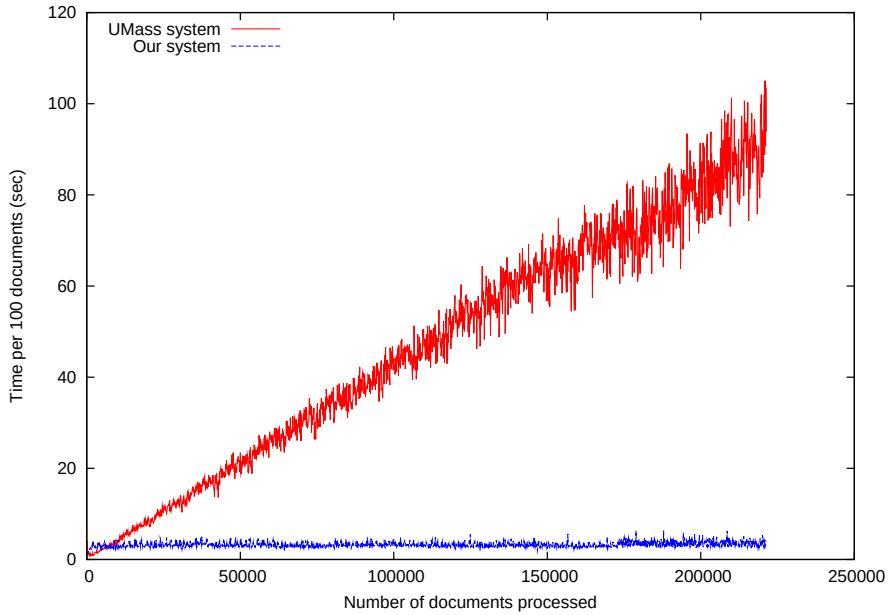
Typical FSD system:

- Store stories (vectors) as they are seen.
- Need to compare stories with each other using a *distance* metric.
- For some new story, find nearest neighbouring story.
- If the new story is ‘far away’ from its nearest story, announce it as a new story.

NN search implies comparing *all* stories against the current one

Newswire Experiments

We compared LSH against an exact system:



Parallelising FSD

- LSH enables us to process each incoming post efficiently.
  - We add and delete Tweets to maintain constant space and time.
- We still need to process thousands of posts per second.
  - Use Storm.

Throughput Experiments

- Compared equivalent *Hadoop* and *Storm* LSH-based FSD implementations
- Task: process 1 million Tweets, looking for novelty.
- Each Tweet is hashed 70 * 13 times.

Throughput Experiments

Parallelism (# Machines)	Hadoop FSD		Storm FSD	
	Total Time (seconds)	Through. (tweets/sec)	Total Time (seconds)	Th. (tweets/sec)
1 Machine (3 cores)	3267	306.09	751.43	1330.80
2 Machines (6 cores)	1556	642.67	559.93	1785.93
3 Machines (9 cores)	1004	996.02	501.81	1992.79
4 Machines (12 cores)	831	1203.37	516.65	1875.85
5 Machines (15 cores)	730	1369.86	502.45	1990.24
6 Machines (18 cores)	606	1650.17	509.09	1964.30
7 Machines (21 cores)	681	1468.43	512.23	1952.25
8 Machines (24 cores)	620	1612.90	515.47	1939.98

- Hadoop has a 24 minute latency; Storm produces results immediately

Event Detection in Twitter

- Less than 5% of Tweets carry news-related content
- Running a traditional FSD system on Twitter will produce a tremendous number of false positives
  - Less than 1% of events detected in Twitter are news related

Examples of spurious events

1. Juicy Couture, Ed Hardy, Coach, Kate Spade and many more! Stay tuned for more brands coming in http://. . .
2. i lovee my nephew hair :D
3. Going to look at houses tomorrow. One of them is & right behind Sonic Taco Casa. If I live there, I might weigh 400 lbs within a year.
4. Hope a bad morning doesnt turn into a bad day...

Quality Improvements

Two strategies:

1. Wait for evidence to accumulate
   - Event detection trades time for fewer false positives
2. Filter false positives using other streams
   - Attempt to remove false positives

Data

Stream	Volume per day	Total Volume	Units
Twitter	662,000	51 Million	Tweets
Wikipedia	240 million	18.5 Billion	Page requests
Newswire	610	47,000	posts

- June 30^{th} to September 15^{th} 2011 (77 days)
- Identified 27 events.
- Task: find breaking news corresponding with these events.

Events in 2011

Event	Newswire	Twitter	Lead
Amy Winehouse dies	**07-23 16:10**	07-23 16:11	-0:01
Atlantis shuttle lands	07-21 09:59	**07-21 09:56**	+0:03
Betty Ford dies	**07-09 00:00**	07-09 00:57	-0:57
Richard Bowes killed in riots in England	**08-11 23:18**	08-11 23:31	-0:14
Flight 4896 crash	**07-13 11:37**	07-13 11:46	-0:09
S&P downgrade US credit rating	**08-06 00:11**	08-06 00:18	-0:07
US increases debt ceiling	08-01 23:06	08-01 23:06	0:00
Terrorist attack in Delhi	09-01 05:12	**09-07 04:53**	+0:19
Earthquake in Virginia	08-23 18:24	**08-23 17:53**	+0:31
First victim of London riots dies	08-09 11:46	**08-09 11:45**	+0:01
War criminal Goran Hadzic arrested	**07-20 07:56**	07-21 05:42	-21:46

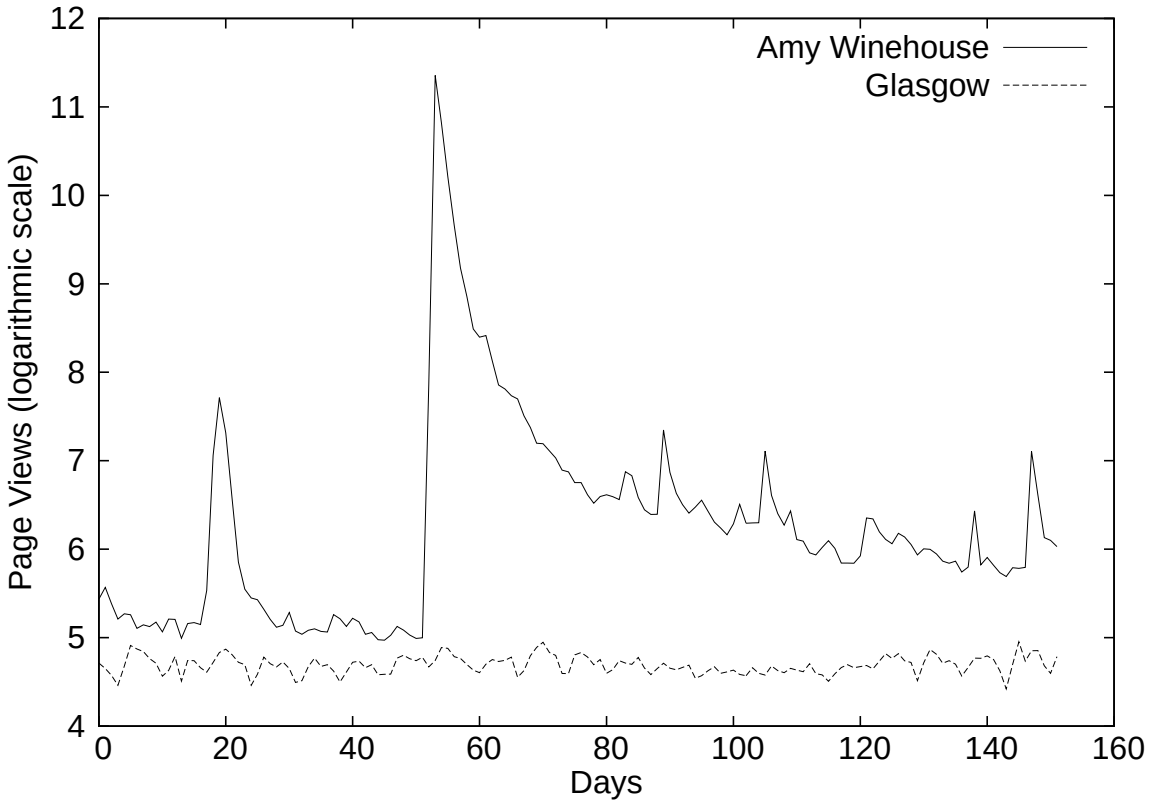
Events in 2011

Event	Newswire	Twitter	Lead
India and Bangladesh sign a border pact	**09-06 07:15**	09-06 14:24	-7:09
Plane with Russian hockey team Lokomotiv crashes	**09-07 12:51**	09-07 12:59	-0:08
Explosion in French nuclear plant in Marcoule	09-12 11:42	09-12 11:42	0:00
NASA announces there might be water on Mars	08-04 18:08	08-04 18:08	0:00
Google announces plans to buy Motorola Mobility	08-15 11:43	**08-15 11:38**	+0:05
Car bomb explodes in Oslo, Norway	07-22 13:57	**07-22 13:38**	+0:19
Gunman opens fire in youth camp in Norway	**07-22 16:13**	07-22 16:14	-0:01

Events in 2011

Event	Newswire	Twitter	Lead
First artificial organ transplant	**07-07 16:03**	07-07 16:25	-0:22
Petrol pipeline explodes in Kenya	**09-12 04:34**	09-12 08:17	-3:43
Famine declared in Somalia	07-20 07:21	07-20 07:21	0:00
South Sudan becomes independent country	**07-08 21:03**	07-08 21:05	-0:02
South Sudan becomes UN member state	**07-14 14:23**	07-14 14:31	-0:08
Three men die in riots in England	08-10 06:33	**08-10 05:45**	+0:48
Riots break out in Tottenham, England	08-06 21:13	**08-06 20:08**	+1:05
Rebels capture International Tripoli Airport	**08-21 08:00**	08-21 23:08	-15:08
Ferry sinks in Zanzibar	**09-10 04:21**	09-10 06:56	-2:35

Wikipedia Page Requests



Filtering Events using Wikipedia

Approach:

- Run FSD system over Twitter.
- Find all time-synchronous spiking Wiki pages.
- If a Tweet matches with a spiking page, emit it.

Filtering Events using Wikipedia

Detecting spiking pages:

- Look for outliers in page request statistics
- Use Grubbs' test:

$$G = \frac{X - \overline{X}}{S}$$

If $G > k$ then page request X is an outlier

Using Wikipedia as Filter

Rank	Event Tweet
1	I love Seth meyers! #ESPYs
2	@tanacondasteve amy whinehouse is dead
3	RT @katyperry: HAPPY 4TH OF JULY!!!!!!!!!!!!!! . . .
4	Yao Ming retired
5	Derek jeter 3000 hits.

Most highly ranked events (by distance to Wikipedia page)
Wikipedia has a 90 minute latency

Filtering Events using Newswire

Approach:

- Run FSD system over Twitter.
- Find all time-synchronous Newswire stories.
- If a Tweet is sufficiently similar to an aligned Newswire page, emit it.

Filtering Events using Newswire

1 http:www.weshopsongs.comnews.html Amy Winehouse, British Soul Singer With a Destructive Image, Dies at 27

2 On Baseball: Jeter Reaches Fabled 3,000, and It's a Blast: At Yankee Stadium, Derek Jeter became the 28th player... http:. . .

3 RT @AdamAndEvePR: Japan trade surplus grows in July: Japan's trade surplus widens by more than expected in July, boosting optimism... ht ...

4 RT @SkyNewsBreak: Petrol bombs thrown at officers and some cars set alight in Derry, Northern Ireland

5 Two arrested over Croydon death: Two men are arrested over the death of Trevor Ellis, who was found with bullet ... http:. . .

Sample events detected; Tweets

Summary

Tackling truly massive data requires:

- Using techniques from Randomised Algorithms to help manage the data.
- Deploying our models using parallel infrastructure.
- Using (online) ML methods that are efficient.

Online ML + randomised methods + parallel infrastructure = taming streams

Summary

Randomising ML brings a new perspective on problem solving:

- Errors are a part of life. Can we tolerate them?
- Randomised methods allow us to explore different kinds of error:
  - False positives (hallucinating data we never saw)
  - False negatives (ignoring parts of the data)
  - Numerical approximations (using simpler representations)
- And often we can trade one for another.
- Which matters will depend upon the problem.

Reading

BFs:

- Giuseppe Gallone. Randomised Features in Discriminative Machine Learning. MSc thesis, 2008. (BFs for ML).

- David Talbot and Miles Osborne. Randomised Language Modelling for Statistical Machine Translation. ACL, Prague, Czech Republic 2007 (Bloom Filters and MT),

- Abby Levenberg and Miles Osborne. Stream-based Randomised Language Models for SMT. EMNLP, Singapore, 2009. (Dynamic Bloom Filters and Language Models)

Reading

LSH:

- Sasa Petrovic, Miles Osborne and Victor Lavrenko. Streaming First Story Detection with application to Twitter. NAACL, Los Angeles, USA. June 2010 (LSH for FSD).

- Miles Osborne, Sasa Petrovic, Richard McCreadie, Craig Macdonald, Iadh Ounis. Bieber no more: First Story Detection using Twitter and Wikipedia. SIGIR 2012 Workshop on Time-aware Information Access, Portland, Oregon, US. August 2012. (Using Wikipedia and Twitter).