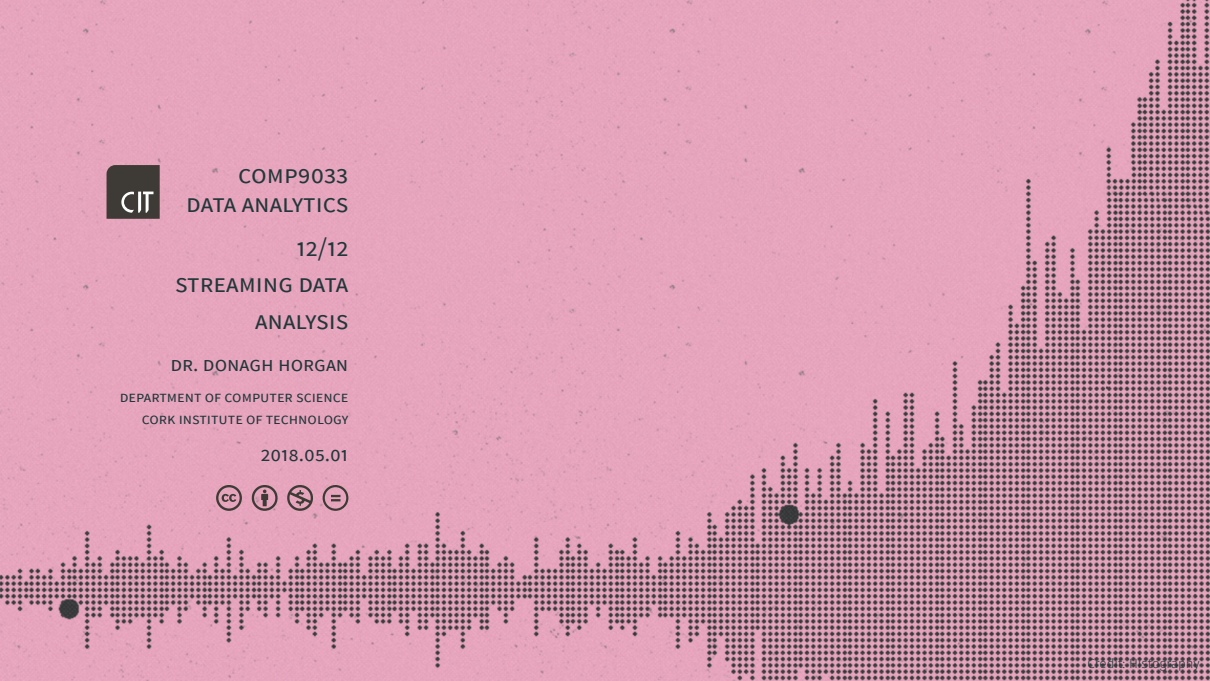


COMP9033
DATA ANALYTICS

12/12

STREAMING DATA ANALYSIS

DR. DONAGH HORGAN

DEPARTMENT OF COMPUTER SCIENCE
CORK INSTITUTE OF TECHNOLOGY

2018.05.01

Overview

1. Big data:
    - The three Vs definition.
    - Why it's relevant now.
2. High volume data:
    - Why it's a problem.
    - Cluster computing.
    - Distributed file systems.
3. MapReduce:
    - Cluster architecture.
    - How it works.
    - Examples.
4. Batch data frameworks:
    - Mahout.
    - TensorFlow.
    - Spark.

1. Data stream processing:
   - The high volume data problem.
   - Data stream frameworks.
   - The lambda architecture.
2. Data stream analytics:
   - Summarisation.
   - Truncation.
   - Modification.
   - Examples.
3. Data law and ethics:
   - Legal vs. ethical restrictions.
   - Legal principles of data protection.
   - Ethical uses of data.
   - Examples.

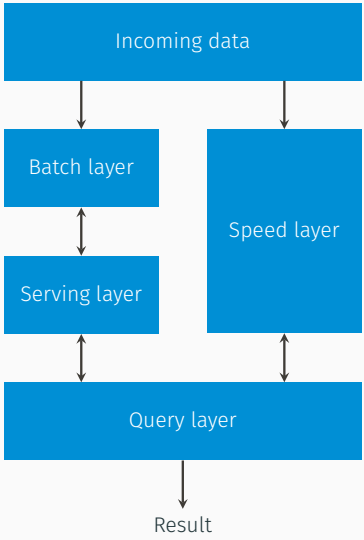
Data stream processing

- Many real world systems produce data in a streamed fashion:
  - Applications produce metrics, *e.g.* CPU usage, memory usage, disk I/O.
  - Sensors produce measurements, *e.g.* temperature, location, health reports.
  - Users produce content, *e.g.* clicks, emails, tweets.
- When the rate at data is produced becomes large, conventional processing techniques begin to fail:
  - If data is produced at a faster rate than it can be stored and indexed, then we cannot access data in a timely manner.
  - If a data packet is delayed in some way (*e.g.* through unlucky routing), it's possible that it can arrive out of order (*e.g.* A, C, B), affecting the outcome of dependent calculations.
- This is the *high velocity* subproblem of big data.

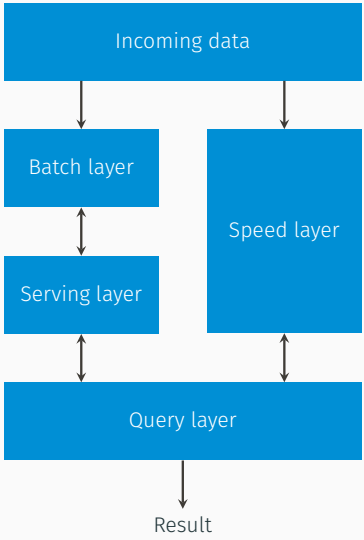
- Batch data processing frameworks (*e.g.* Hadoop) are not usually good solutions for streamed data processing because of the assumptions they make about the data being processed, *e.g.*
  - Data is processed in batch and read in large chunks, *i.e.* high throughput more important than low latency.
  - Data is written once and read many times, *i.e.* data is usually stored in the "correct" format for later processing and is subsequently appended to if necessary.
- Data stream processing has different requirements, *e.g.*
  - Data must processed as it arrives, *i.e.* low latency is more important than high throughput.
  - Data may arrive out of order, *i.e.* the data may not arrive in the order in which it should be processed and may need to be updated at a later point.

- A number of frameworks have been developed for processing and/or analysing data streams, *e.g.*
  - Apache Flink.
  - Apache Samza.
  - Apache Spark.
  - Apache Storm.
  - Oryx (uses Spark).
- The different frameworks tend to have different design goal priorities, *e.g.* fault tolerance, latency, read/write throughput, scalability.
- Some frameworks offer stream processing only, while others support stream and batch data processing.

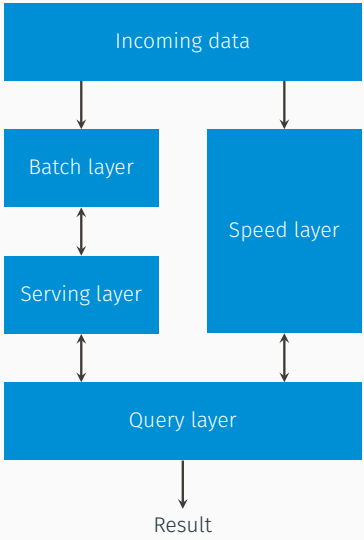
- The *lambda architecture* combines both batch and stream data processing in one system.
- It offers the best of both worlds:
  - High throughput and fault-tolerant processing of historical data, via a batch processing layer.
  - Low latency and in-order data processing of recent data, via a speed processing layer.
- Both Apache Flink and Oryx support lambda architecture configurations.



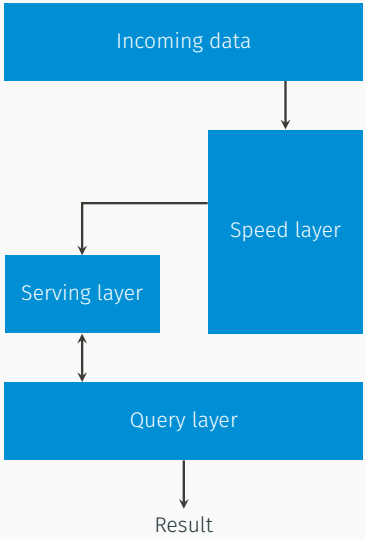
- The architecture consists of a number of elements:
  1. Incoming data: this is sent to both the batch and speed layers.
  2. Batch layer: high throughput, fault tolerant data storage (*e.g.* HDFS) and processing (*e.g.* MapReduce).
  3. Serving layer: a low latency cache for regularly accessed historical data.
  4. Speed layer: low latency processing and cache for recent data.
  5. Query layer: queries are answered by merging results from the speed (recent) and serving (historical) layers.



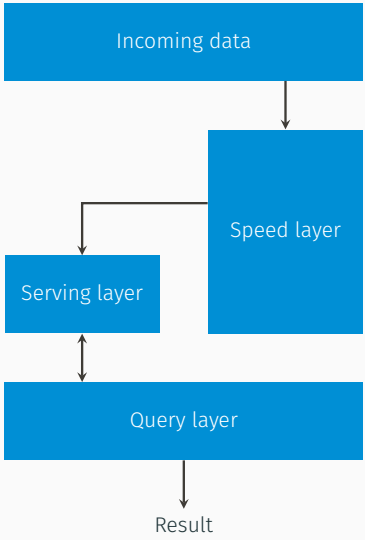
- While the lambda architecture has many benefits, it has one big drawback: processing tasks (*e.g.* analytics) that run on the batch layer must also be run on the speed layer.
- Generally, this means that tasks must be written twice: once for the batch framework (*e.g.* MapReduce) and once for the stream framework (*e.g.* Storm).
- This creates dependency issues that can delay the development of new features and complicate the maintenance of existing ones.



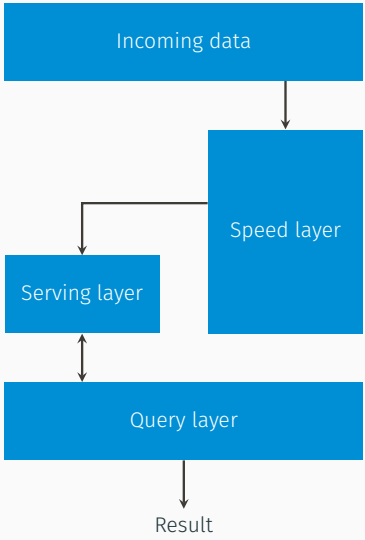
- The *kappa architecture* is a simplification of the lambda architecture.
- It eliminates the need for a batch layer by persisting historic data to a log store (*e.g.* Kafka), utilising the speed layer for all data processing tasks and the serving layer to answer all queries.
- As batch data processing is no longer required, *any* streaming framework can be used in the speed layer.



- The components of the kappa architecture differ to those of the lambda architecture:
  1. Incoming data: a log store (*e.g.* Kafka) that can relay live data and replay historic data (as requested) to the speed layer.
  2. Speed layer: low latency data processing functionality, relays all processed data to the serving layer.
  3. Serving layer: a low latency cache for regularly accessed historical data and recently processed live data.
  4. Query layer: queries are answered by the serving layer only.



- As the speed layer is responsible for all data processing tasks, there is no duplication of responsibility as with the lambda architecture.
- However, this comes at the cost of having to replay and reprocess data in cases where it is not present in the serving layer.
- While this increases latency on cache misses, it is possible to tune the system so that the probability of a significant delay is reduced to an acceptable level.



Data stream analytics

- Data analysis tends to be more complicated to implement on data streams.
- For instance, if we want to detect anomalies on a live data stream using linear regression, then we will need to store the training data in-memory in order to ensure low latency.
  - If the data rate is very high, then we will have to store lots of data in-memory.
  - If the number of streams is very large, then we will have to store lots of data in-memory.
  - Eventually, there will come a point where the data rate or number of parallel streams is so large that there will not be enough memory available.
- In general, there are two solutions to these problems:
  1. Use a lambda architecture, at the cost of higher latency.
  2. Modify algorithms so they can better deal with streams of data.

- There are typically three ways to modify an algorithm:
  1. Summarise the data.
  2. Truncate the data.
  3. Use a specialised streaming algorithm (only applicable in some cases).
- Summarisation reduces the amount of data to process by discarding the most redundant parts, *e.g.*
  - Sampling the data, so that there are fewer points to be analysed.
  - Filtering the data, *e.g.* computing a rolling mean.
  - Using dimensionality reduction techniques, *e.g.* principal component analysis.
- Data summarisation is lossy: we discard potentially useful information in the process and so the accuracy of our analysis may suffer.

- Truncation involves applying a window function to the data, which discards all data before a certain point.
- Data truncation is also lossy, but only the oldest data is lost — this may more acceptable than summarisation in some cases.
- Some algorithms can be adapted to operate on streaming data more efficiently, with no loss of accuracy, *e.g.*
  - Miller updating linear regression.
  - Sequential *K*-means clustering.
- However, the type of specialisation generally depends on the algorithm (*i.e.* there are few generic solutions) and some algorithms cannot be specialised.

- Usually, we compute the mean of the sample $X = \{x_1, x_2, \ldots, x_n\}$ as

$$\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i, \tag{12.1}$$

  i.e. we sum up the historical values of X and divide by their total count.

- If we were analysing data streams using Equation 12.1, we would have to hold all n data points in-memory, per stream.
- If each data point had a 64 bit memory footprint (*e.g.* Java doubles), and data arrived at one sample per second, then we would require approximately 20 MB of memory to store a month of data, per stream.
- On commodity hardware, with 16 GB of memory, we would be limited to analysing 800 streams, at most!

- However, we can rewrite Equation 12.1 as follows:

$$\begin{aligned}
\bar{x}_n &= \frac{1}{n}\sum_{i=1}^{n-1} x_i + \frac{x_n}{n} \\
&= \frac{1}{n}\left(\frac{n-1}{n-1}\right)\sum_{i=1}^{n-1} x_i + \frac{x_n}{n} \\
&= \frac{n-1}{n}\left(\frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right) + \frac{x_n}{n} \\
&= \frac{n-1}{n}\bar{x}_{n-1} + \frac{x_n}{n}. \qquad (12.2)
\end{aligned}$$

- Equation 12.2 depends on n, $\bar{x}_{n+1}$ and x_n only!

- Equation 12.2 depends on n, $\bar{x}_{n-1}$ and x_n only!
- This means we only need to keep track of *three* pieces of data, instead of n pieces of data.
  - No matter how large n becomes, we won't require any additional memory.
  - With 16 GB of memory, we can now analyse approximately 5.7×10^9 streams!

```
class StreamingMean():

  def __init__(self):
    self.n = 0
    self.mean = None

  def compute(self, x):
    # Catch the edge case on the first call
    if self.n == 0:
      self.n = 1
      self.mean = x
    else:
      self.n = self.n + 1
      self.mean = ((self.n - 1) * self.mean + x) / self.n

    return self.mean
```

- This kind of approach works for the streamed computation of other kinds of statistics too, *e.g.* standard deviation / variance, Pearson correlation.
- However, not every calculation has a streamed equivalent — in cases where modified algorithms aren't available (*e.g.* median, IQR), we must rely on summarisation or truncation.

```
class StreamingMean():

  def __init__(self):
    self.n = 0
    self.mean = None

  def compute(self, x):
    # Catch the edge case on the first call
    if self.n == 0:
      self.n = 1
      self.mean = x
    else:
      self.n = self.n + 1
      self.mean = ((self.n - 1) * self.mean + x) / self.n

    return self.mean
```

Data law and ethics

- Data analytics is a powerful tool that can extract deep insights from relatively small amounts of data, and so should be used in a responsible manner.
- Broadly speaking, there are two categories of restrictions on the use of data analytics:
  1. Legal restrictions: imposed by law, depending on the jurisdiction in which data is collected and/or processed.
  2. Ethical restrictions: respecting the privacy and dignity of others, beyond what is required by law.
- Generally, the issues in these categories are summarised by three questions:
  1. *Who* should be granted access to data?
  2. *What* should they be granted access to?
  3. *How* should they be allowed to use it?

- The *General Data Protection Regulation* (GDPR) is an EU regulation on data protection, due to be implemented on 25 May 2018.
- It aims to modernise and unify personal data collection and processing rules for the benefit of both citizens and businesses.
- Citizens' rights include:
  - Easier access to personal data and straightforward (*i.e.* human readable) information on how data is used.
  - The right to erasure, *i.e.* the right to have your personal data erased on request, when there is no legal reason to retain it.
  - The right to know when personal data has been hacked.
  - The right to data portability, *i.e.* the right to transfer your data to a different service provider easily.

- The GDPR also ensures that businesses must adhere to a set of regulations:
  - By implementing a single set of rules that are applied across the EU, reducing the number of individual regulations that must be adhered to, and applying the same set of rules to EU and non-EU companies.
  - By requiring that data protection safeguards are built in to products “by design and by default”.
  - By enforcing privacy-friendly techniques (*e.g.* anonymisation and encryption).
  - By providing a “one stop shop” authority for data protection issues.
- These regulations benefit existing compliant businesses over (usually non-EU) non-compliant businesses.
- Exemptions are made for SMEs where the rights and freedom of the individuals whose data is being processed is unlikely to be affected.

1. Lawfulness, fairness and transparency:
    - Data must be processed lawfully, fairly and in a transparent manner.
2. Purpose:
    - Data must be collected for specified, explicit and legitimate purposes.
    - You must be informed of, and consent to, the purpose for which your data is being collected *in plain language*.
    - You can withdraw your consent at any time.
    - Data cannot be further processed beyond the purpose for which it was collected, although exemptions are made in cases where the public interest is served, *e.g.* archiving, statistics, scientific studies and historical research.
3. Minimisation:
    - Collected data must be relevant to the stated purpose.
    - Data collection must be limited to what is necessary for achieving the purpose.

4. Accuracy:
    - Collected data must be accurate and, where necessary, kept up to date.
    - Inaccurate data must either be corrected or erased, without delay.
5. Identification:
    - Data must kept in a form which permits identification of its subjects for no longer than is necessary for the purpose for which it is processed.
    - Exemptions are made in cases where further processing is in the public interest – as with the Purpose principle – but under the additional restriction the rights and freedoms of the data subject are safeguarded.
6. Security:
    - Data must be processed in a secure manner.
    - Unauthorised or unlawful processing of the data must be prevented.
    - Data must also be protected against accidental loss, destruction or damage.

- In the Unites States, the constitution grants all US citizens the right to an “expectation of privacy”.
- *Mosaic theory* is the legal argument that individual actions, analysed collectively, may violate this expectation.
- Generally, it is invoked in cases where multiple search warrants were used in the investigation of a suspect, the results of which were combined at a later point in time.
- In 2012, an argument based on mosaic theory was upheld by the US Supreme Court (United States vs. Jones).
- In this instance, it was found that a GPS trace was unlawfully used to construct a profile of the suspect’s life that violated his expectation of privacy, even though it was obtained on legal grounds.

- It has been suggested that machine learning could be used to analyse independent instances of surveillance data in a mosaic fashion.
- The basic idea is that if the data *could* be combined in *some* way that constitutes a mosaic search, then this would violate the law.
- If the argument holds, it could have serious implications for law enforcement and the legal system:
  - It may not be clear at the time a search warrant is granted whether a machine learning system that is capable of creating such a profile exists.
  - It is possible that such a system *could* exist at some point in the future, at which point the combined data would automatically become a mosaic search.
- While this suggestion has been controversial, it is indicative of the confusion that exists between current laws and evolving technologies.

- Ethical restrictions on the use of data are not legally enforceable and vary in definition from person to person and from culture to culture.
- While there are no enforceable rules on what should or shouldn't be done, the general principle is that *just because you can do something, doesn't mean you should do something*.
- There is often an incentive for service providers to comply as unethical behaviour usually leads to negative publicity.
- However, this hasn't prevented violations in the past and, presumably, will not prevent others in the future.

- Purpose and consent play a key role in the ethical use of data:
  - Consumers may not be comfortable with certain kinds of analysis being performed on their data.
  - If the purpose of the analysis is legitimate, then there is no obligation on the service provider to cease.
- One way around this is allowing consumers to opt out of analyses:
  - Consumers can continue to use the service, without their data being analysed in ways they find objectionable.
  - Typically, only a small number of consumers will choose to opt out, in which the results of the analysis will not be biased significantly.
- Many providers implement this kind of policy, *e.g.* Amazon (personalised ads), Google Maps (location history), Netflix (UX testing).

- In 2002, marketers at the US retail store Target started to investigate whether it was possible to reliably predict whether a customer was pregnant based on their purchasing habits.
- Target wanted to increase their customer base in the maternity products market, but faced stiff competition from other retailers.
- New mothers can be identified through public records and purchasing habits and are usually inundated with special offers very soon after the birth of their child.
- Target decided to try to identify mothers in their second trimester, so that they had a significant time advantage over their competition.

- Statisticians at Target were able to model the situation quite well:
  - By examining changes in purchase behaviour, the model was able to predict whether a given customer was pregnant or not to a high degree of accuracy.
  - 25 products were identified as key indicators of pregnancy, including vitamin supplements, cotton balls, scent-free soap and wash cloths.
  - The model was accurate enough to be able to predict not only that a customer was pregnant, but also when they were due to within a small window of time.
- All of the data collected and the analyses carried out were legal, but was the analysis ethical without the explicit consent of the customers?
  - The privacy of customers was eroded, without their explicit knowledge or consent.
  - In one case, a teenage girl was sent pregnancy coupons, prompting her father to complain to Target only later to find out that his daughter was in fact pregnant.

- Every day, the NSA collects massive amounts of data on US citizens and residents from a variety of sources and uses machine learning to try to detect potential security threats to the United States and its allies.
- Consider the following:
  - There are more than 300 million people in the United States.
  - There are very few genuine terrorist plots in the United States (*i.e.* terrorism is rare).
- However, the NSA is well-funded, employs expert analysts and has access to the best technology.
- How well does their system work? Very few people know.
- How well *can* their system work? Let's do the maths!

- Let's start by making some basic assumptions:
  1. Let's say that the NSA collects, on average, ten pieces of information (*i.e.* features) about each person per day, *e.g.* dialled phone numbers, emails, tweets, websites visited, purchases made.
  2. Let's also assume that there are about ten people in the country actively plotting terrorist activity.
  3. Finally, let's assume that the machine learning algorithms used by the NSA are cutting edge and can produce a model with a false positive rate of 1% and a false negative rate of 0.1%.
- This means that, over the course of a year, the NSA would collect about one hundred billion records ($365 \times 300 \times 10^6 \approx 100 \times 10^9$).

- As the false negative rate is very low (0.1%), it is unlikely that a genuine threat will be misclassified, *i.e.*

$$\begin{aligned} N_{fn} &= 0.001 \times 10 \\ &\approx 0. \end{aligned}$$

- However, the total number of false alarms is given by

$$\begin{aligned} N_{fp} &= 0.01 \times 100 \times 10^9 \\ &= 10^9, \end{aligned}$$

i.e. the system would produce about a billion false alarms every year!

- While the system is capable of discovering the ten genuine terror plots, it is unlikely to be of any practical use:
  - The number of false alarms swamps the number of real events, and so each event must be manually investigated to determine whether it is genuine.
  - However, there would be approximately 2.7 million terror alerts per day, which would be infeasible for law enforcement to investigate.
- Are we being too pessimistic?
  - Let's assume instead a machine learning system with a false alarm rate of 0.0001%, *i.e.* it only finds spurious relationships one time in a million.
  - The new system would generate about one hundred thousand false alarms a year, the equivalent of approximately 270 terror alerts per day.
  - Again, this dwarfs the number of genuine terror plots we are hoping to discover and so manual assessment is required and may again be infeasible.

- The NSA probably collects more information than we have assumed, *e.g.*
  - Data on non-US citizens, which may lead to high volume data problems.
  - A faster update rate than once per day, which may lead to high velocity data problems.
  - A greater number of features per person, which may lead to the curse of dimensionality.
- Finding terrorist plots is like finding a needle in a haystack; however, making the haystack bigger does not necessarily help to find the needle.

- In many situations, service providers have access to *personally identifiable information* (PII), *e.g.* names, addresses, dates of birth, email addresses, credit card numbers, government issued ID numbers.
- Does *every* employee of the service provider *need* access to this data?
  - In some situations (*e.g.* medical records), this is regulated by law.
  - In lots of other situations, it isn't, and there is no obligation on providers to manage access.
- If access restriction is not an option, then anonymisation should be considered, *i.e.* the replacing of PII with unique but unintelligible identifiers, *e.g.* with encryption.
- This way, even if data is copied, stolen or accidentally transferred, sensitive information will not be readily accessible.

- The US ride-sharing company Uber has been accused of numerous data ethics violations over the past few years:
  - Uber developed an internal tool — known as God View — that allowed its employees to track the location of its customers in real time.
  - Historical data was also available to many employees within the company.
- Numerous incidents have been reported:
  - In 2014, a manager tracked the current location of a journalist as she travelled in an Uber car.
  - The same manager also accessed the journalist's historical trip data, without her knowledge or consent.
  - Also in 2014, an executive threatened to reveal sensitive information about another journalist who had written a negative piece about the company.
  - In 2015, Apple threatened to remove Uber's app from their App Store as a result of Uber covertly tracking iPhone users in violation of Apple's privacy policy.

Summary

- Streamed data processing requires an additional layer of problems to be solved:
  - Architecture: pure streams or a lambda/kappa approach?
  - Algorithms: summarisation, truncation or modification?
- Legal and ethical questions:
  - Law sets out the minimal standard to be followed.
  - Ethics require us to take individual responsibility in specific situations.
- Lab work: use Apache Spark to analyse a data stream.
- This week is the last lecture!

1. Ullman et al. *Mining of Massive Data Sets.* Cambridge University Press, 2014. (stanford.io/1qtgAYh)
2. C. Anderson. *Creating a Data-Driven Organization.* O'Reilly, 2015. (oreil.ly/1HSStYo)
3. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (bit.ly/1TtydR4)
4. Bellovin at al. *When enough is enough: Location tracking, mosaic theory, and machine learning.* NYUJL & Liberty, 2013. (bit.ly/2pDsF0m)
5. B. Schneier. *Data Mining for Terrorists.* Schneier on Security, March 2006. (bit.ly/2pDQLuO)