EECS4415: Big Data Systems



Data Ingestion and Data Quality

Big Data Technology & Analytics

Data Ingestion ETL, Distcp, Kafka, OpenRefine, 	Query & Exploration SQL, Search, Cypher,	
	Stream Processing Platforms Storm, Spark,	Data
	Batch Processing Platforms MapReduce, SparkSQL, BigQuery, Hive, Cypher,	Serving Bl, Cubes, RDBMS, Key-
	Data Definition SQL DDL, Avro, Protobuf, CSV	value Stores, Tableau,
	Storage Systems HDFS, RDBMS, Column Stores, Graph Databases	
Computer P	latforms	

Distributed Commodity, Clustered High-Performance, Single Node

Data Ingestion

Analytics solutions start with data ingestion

Data integration challenges: volume (many similar integrations) variety (many different integrations) velocity (batch v.s real-time) (or all of the above)

Needs of Data Analytics



* A theory in psychology proposed by Abraham **Maslow** in 1943. Needs lower down in the hierarchy must be satisfied before individuals can attend to needs higher up.

Challenge: Many Sources



Challenge: Many Schemas



Different structures Different manifest data types Different literal for same data type Different Keys

Challenge: Failures



Failures when producers push data Failures storing received data Failures while transferring data over network

Data Bundle Semantics Examples

Full dumps	All data is provided at once and replaces all previous data.
Incremental	Increments are provided in any order, data interval in an increment is provided as metadata. On an hourly, daily or weekly cadence.
Append	Always appended at the end of the dataset. Order is assumed to be correct.
Stream	Stream of incoming data as individual rows or in small batches. On a second, minute or hourly cadence.

ETL: Extract, Transform, Load



Requires transforms to know what reporting, query to enable

ELT: Extract, Load, Transform



- Made possible by more powerful target systems
- Provides more flexibility at later stages than ETL

High Velocity Technologies

Kafka	Kafka is a distributed, partitioned, replicated commit log service. It provides the functionality of a messaging system, but with a unique design.
Kinesis	Amazon Kinesis is a fully managed, cloud-based service for real-time data processing over large, distributed data streams. Amazon Kinesis can continuously capture and store terabytes of data per hour from hundreds of thousands of sources.
S4	S4 is a general-purpose, distributed, scalable, fault-tolerant, pluggable platform that allows programmers to easily develop applications for processing continuous unbounded streams of data.
Storm	Apache Storm is a distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.
Samza	Apache Samza is a distributed stream processing framework. It uses <u>Apache Kafka</u> for messaging, and <u>Apache Hadoop YARN</u> to provide fault tolerance, processor isolation, security, and resource management.

Kafka Approach



Scalability	Allows many producers and consumers. Partitions are the unit of scale.
Schema Variety	Does not really solve this.
Network Bottlenecks	Can handle some variability without losing messages.
Consumer Bottlenecks	Kafka acts as a buffer allowing producers and consumers to work at different speeds.
Bursts	Handles buffering of messages between producers and consumers.
Reliability, Fault Tolerance	Allows reading of messages if a consumer fails.





- Data is produced in Bursts
- Consumers can only consume at a certain rate
- Dropping data can be a problem



Data Quality

Bad Data

- Misspellings {Montgromery street}
- Outliers {1,2,4,2,4,123,3,4}
- Incorrect values {invalid zip, neg number}
- Missing values {6,7,4,3,,4,5,6}
- Incorrect values {94025, -345,96066,...}

So, What is Data Quality?

It depends!

- Entry Quality: Is the record entered correctly?
- Process Quality: Was the record correct throughout the system?
- Identity Quality: Are similar entities resolved to be the same?
- Integration Quality: Is the data sufficiently integrated to answer relevant questions?
- Usage Quality: Is the data being used correctly?
 - Age Quality: Is the data new-enough to be trusted?

Data Quality as a Hierarchy



Data Quality Framework

Quality Evaluation Framework

Observation

- It's too expensive to clean all the data every way
- How do we decide what to clean?

We need **a framework** that helps to:

- Determine what issues might occur in the data
- Weight the criticality of the issues
- Profile the data to score quality

The framework allows:

- to approach quality as an ever-increasing standard
- To prioritize data cleaning activities

Quality Issues for Single Streams

Definitional

Constants
Definition Mismatches
Filler Containing Data
Inconsistent Cases
Inconsistent Data Types
Inconsistent Null Rules

Incorrect

Invalid Keys

Invalid Values

Miscellaneous

Missing Values

Orphans

Out of Range

Too-Soon-to-Tell

_	
	Pattern Exceptions
	Potential Constants
	Potential Defaults
	Potential Duplicates
	Potential Invalids
	Potential Redundant Values
ſ	Potential Unused Fields
	Rule Exceptions
	Unused Fields

Weighting Issues

Weight Factor	Issue Type
1	Constants
2	Definition Mismatches
2	Filler Containing Data
1	Inconsistent Cases
2	Inconsistent Data Types
3	Inconsistent Null Rules
5	Invalid Keys
5	Invalid Values

- Issues should be weighted in the context of our system
- Invalid or missing data is most problematic
- Potentially issues can be weighted lower

Assessing Quality

Weight	Issue Type	Disco vered	Possible
1	Constants	1	59
2	Definition Mismatches	4	59
2	Filler Containing Data	1	59
1	Inconsistent Cases	3	59
2	Inconsistent Data Types	15	59
3	Inconsistent Null Rules	6	59
5	Invalid Keys	1	3
5	Invalid Values	5	59

Any issue has a possible maximum

- This can be cardinality of
 - Rows
 - Keys
 - etc.

Quality Score

Raw Score	91.3%		
Weighted Score	90.4%		

Missing Values

Nuclear Option: Listwise Removal

- If a record has a missing value → Ignore the entire record in analyses
- Softer Option: Pairwise Removal
 - If a record has a missing value → Ignore the record <u>iff</u> the missing value is in the analyzed set
- Infer Substitute Values
 - Statistical (Interpolation): predict new data points in the range of a set (Simple linear, Splines, Kriging)
 - Model-based: Impute missing values based on a model (EM algorithm, etc.)

Entity Resolution/ Record Linkage

Entity Resolution/Record Lineage



What is it?

The task of finding records in a data set that refer to the same entity across different data sources

Problems

- Attributes/records not matching 100%
- How to apply to different/ changing data sets
- Missing values
- Data quality, errors etc.
- Semantic relations

Core Problems (to address)

- Normalize data
- Link records on a table
- Link records cross tables



Customer (source 1)

CID	Name	Street	City	Sex
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Client (source 2)

Cno	LastName	FirstName	Gender	Address	Phone/Fax
24	Smith	Christoph	М	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Customers (integrated target with cleaned data)

No	LName	FName	Gender	Street	City	State	ZIP	Phone	Fax	CID	Cno
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503- 5998	444-555- 6666		11	493
2	Smith	Christian	М	2 Hurley Place	South Fork	MN	48503- 5998			24	
3	Smith	Christoph	М	23 Harley Street	Chicago	IL	60633- 2394	333-222- 6542	333-222- 6599		24

Types of Problems



Single-source Cleaning

Single Source Problems

Scope/Prob	lem	Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city="Liipzig"	usually typos, phonetic errors
	Cryptic values,	experience="B";	
	Abbreviations	occupation="DB Prog."	
	Embedded values	name="J. Smith 12.02.70 New York"	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city="Germany"	
Record	Violated attribute dependencies	city="Redmond", zip=77777	city and zip code should correspond
Record	Word	name ₁ = "J. Smith", name ₂ ="Miller P."	usually in a free-form field
type	transpositions		-
	Duplicated records	emp ₁ =(name="John Smith",); emp ₂ =(name="J. Smith",)	same employee represented twice due to some data entry errors
	Contradicting	emp ₁ =(name="John Smith", bdate=12.02.70);	the same real world entity is described by
	records	emp ₂ =(name="John Smith", bdate=12.12.70)	different values
Source	Wrong references	emp=(name="John Smith", deptno=17)	referenced department (17) is defined but wrong

Scope/Probl	em	Dirty Data	Reasons/Remarks
Attribute	Illegal values	bdate=30.13.70	values outside of domain range
Record	Violated attribute	age=22, bdate=12.02.70	age = (current date - birth date)
	dependencies		should hold
Record	Uniqueness	emp ₁ =(name="John Smith", SSN="123456")	uniqueness for SSN (social security
type	violation	emp ₂ =(name="Peter Miller", SSN="123456")	number) violated
Source	Referential	emp=(name="John Smith", deptno=127)	referenced department (127) not defined
	integrity violation		

Deterministic vs. Fuzzy Methods

Deterministic

rules + record level match

Fuzzy

 Match similar values and records and attempt to determine if *match*, *no match* or *possible match*

Deterministic Approach

Deterministic: rules + attribute level match

Data Set	#	SSN	Name	DOB	Sex	ZIP
	1	000956723	Smith, William	1973/01/02	Male	94701
Sot A	2	000956723	Smith, William	1973/01/02	Male	94703
Sel A	3	000005555	Jones, Robert	1942/08/14	Male	94701
	4	123001234	Sue, Mary	1972/11/19	Female	94109
Set P	1	000005555	Jones, Bob	1942/08/14		
Sel D	2		Smith, Bill	1973/01/02	Male	94701

if a.SSN == b.SSN => match if (missing(a.SSN) or missing(b.SSN)) => if ((a.dob == b.dob) and (a.zip == b.zip) and (a.sex == b.sex)) => match

Fuzzy Approach

Fuzzy: similarity of attribute + distance of record + match thresholds + learning



match,

Fuzzy Approach

Recall vs Precision

Fuzzy + rules => practice

"Generate rules" by using machine learning



Similarity Metrics

- Levenshtein Distance (Edit Distance)
- Jaro Distance
- Jaro-Winkler Distance

Levenshtein Distance/ Edit Dist.

- Minimum number of single character edits:
 - Insert
 - Delete
 - Substitute

Stokcholm -> Stockholm sanfransicso-> san francisco sanfransicso-> San Francisco {Distance = 2} {Distance = 4} {Distance = 6}

Levenshtein

Steps 1 and 2

		G	U	Μ	B	0
	0	1	2	3	4	5
G	1					
Α	2					
М	3					
В	4					
0	5					
L	6					

Step	Description
1	Set n to be the length of s.
	Set m to be the length of t.
	If $n = 0$, return m and exit.
	If $m = 0$, return n and exit.
	Construct a matrix containing 0m rows and 0n columns.
2	Initialize the first row to 0n.
	Initialize the first column to 0m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0.
	If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of:
	a. The cell immediately above plus 1: d[i-1,j] + 1.
	b. The cell immediately to the left plus 1: d[i,j-1] + 1.
	c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Steps 3 to 6 When i = 1

	0	G 1	U 2	М 3	В 4	0 5	
G	1	0					
Α	2	1					
М	3	2					
В	4	3					
0	5	4					
L	6	5					

Steps 3 to 6 When i = 2

		G	U	M	B	0
	0	1	2	3	4	5
G	1	0	1			
Α	2	1	1			
Μ	3	2	2			
B	4	3	3			
0	5	4	4			
L	6	5	5			

Steps 3 to 6 When i = 3

		G	U	M	B	0
	0	1	2	3	4	5
G	1	0	1	2		
Α	2	1	1	2		
M	3	2	2	1		
В	4	3	3	2		
0	5	4	4	3		
L	6	5	5	4		

Steps 3 to 6 When i = 4

		G	U	M	B	0
	0	1	2	3	4	5
G	1	0	1	2	3	
Α	2	1	1	2	3	
M	3	2	2	1	2	
В	4	3	3	2	1	
0	5	4	4	3	2	
L	6	5	5	4	3	

Steps 3 to 6 When i = 5

		G	U	M	В	0	
	0	1	2	3	4	5	
G	1	0	1	2	3	4	
Α	2	1	1	2	3	4	
Μ	3	2	2	1	2	3	
В	4	3	3	2	1	2	
0	5	4	4	3	2	1	
L	6	5	5	4		2	
							-

distance

Other Algorithms

- Hamming Distance
- Damerau-Levenshtein Distance
- Longest Common Substring Distance
- Cosine Distance
- Jaccard Distance
- Jaro Distance
- N-grams

Multi-source Cleaning

Summary of problem

- Assume you want to link records across sources that do not share a common record key
 - These records may have data quality issues
- How would you create such linkage using fuzzy and deterministic methods?

Multi Source Approach



Fuzzy Methods, Multi-records

- Defining similarity is application specific
- Assume you have no common, unique key
- What can you do to identify similar record in different (or the same) source
- Example:
 - Cosine similarity for numeric /continuous attributes
 - Edit distance for strings
 - Jaccard for binary or categorical values

Principle (No unique key)



We map columns: a2<->b5 a3<->b3 a5<->b6

Compare record 1 from table 1, with record 100 from table 5. Are they representing the same entity? Are the attributes continuous, categorical, string, etc.?

Continuous Attributes

For **continuous attributes** we can calculate cosine distance



```
Let d1 = 20 30 40

Let d2 = 10 30 50

Cosine Similarity (d1, d2) = dot(d1, d2) / ||d1|| ||d2||dot(d1, d2)

||d1|| = sqrt((20)^2 + (30)^2 + (40)^2) = 53.8516480713

||d2|| = sqrt((10)^2 + (30)^2 + (50)^2) = 59.160797831

Cosine Similarity (d1, d2) = 3100 / (53.8516480713) * (59.160797831

= 3100 / 3185.90646441

= 0.97303547189
```

String: calculate an {edit} distance.

Categorical: bit vectors (0/1) for *different/same*

For sets of binary or categorical

The **Jaccard distance**, which measures *dis*similarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union:

$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

Example Record Similarity



score = $w_1^*(cd(cont(r_1,r_5)))+w_2^*(ed(r_1.a_n,ry.by))+w_3^*(compare(r_1.a_k,r_5.a_l))$ Cosine distance

Ps. this just an illustration, how to model similarity is application specific.

You define score ranges for same, possibly same, not same using thresholds

Scaling Record Linkage



Entity matching can be very computationally intensive. How to scale it?

Understand basic scaling techniques
 Understand the techniques pros and cons

Scale/Efficiency Issues



- Polynomial complexity
- N*M record comparisons
- Expensive comparison operations
- May require multiple passes over the data

Concerns

Precision and Recall

Efficiency

Processing Time





MB of processed data/ cost of infrastructure





Ways to Cope With Complexity



Sliding Window

- Create key from relevant data
- Sort on key
- Compare records in sliding window

Pros: Fewer comparisonCons: Sensitivity to key selection, missed matches



Blocking

- Define a blocking key
- Create blocks
- Compare within blocks

Pros: fewer comparisons **Cons**: missed matches







_

Clustering

- Similarity is transitive
- a->b->c
- Union-find algorithm

Pros: Fewer comparison
Cons: Potentially complex computation



Canopy Clustering

- Records can belong to multiple canopies
- Cluster into canopies using "cheap" algorithm

Pros: Fewer comparisonCons: Sensitivity to key selection, missed matches



- 1. Begin with the set of data points to be clustered.
- 2. Remove a point from the set, beginning a new 'canopy'
- 3. For each point left in the set, assign it to the new canopy if the distance less than the loose distance
- 4. If the distance of the point is additionally less than the tight distance , remove it from the original set
- 5. Repeat from step 2 until there are no more data points in the set to cluster
- 6. Clustered canopies are sub-clustered using an expensive but accurate algorithm

Reduce Comparison Complexity

Decision Tree





Figure 2: The rule set produced by translating the information in the induced decision tree.



- then class=matched-yes with certainty factor 0.933
- if name-edit-distance >= 4.5
 - then class=matched-no with certainty factor 1.0

Figure 3: The rule set produced by simplifying the initial rule set.

Processing Time



Data cleaning Hands-on Activity

Cleaning Tools

- OpenRefine (ex Google Refine)
 - Watch Introduction (6:48)
 - https://www.youtube.com/watch?v=B70J_H_zAWM
 - Install Open Refine (and Java if needed)
 - <u>http://openrefine.org/</u>
 - Download a data set
 - City of Berkeley Employee Salaries 2013 Data from the link: <u>https://data.cityofberkeley.info/Economic-Data/City-of-Berkeley-Employee-Salaries-2013/ifen-52iq</u>
 - Create a new OpenRefine Project and load the data
 - Try to answer a few Questions below

Q1: Who gets the highest base pay in the city?

Q2: Who gets the highest 'total pay + benefits' in the city?

Q3: Group all police related job titles to a 'Police' job title

Q4: Export all information related to 'Police' jobs to a new CSV file