Lecture 4:
Data Integration and Cleaning

CMPT 733, SPRING 2017

JIANNAN WANG

Outline

Motivation

- Asking “Why” before you learn something

Data Integration and Cleaning

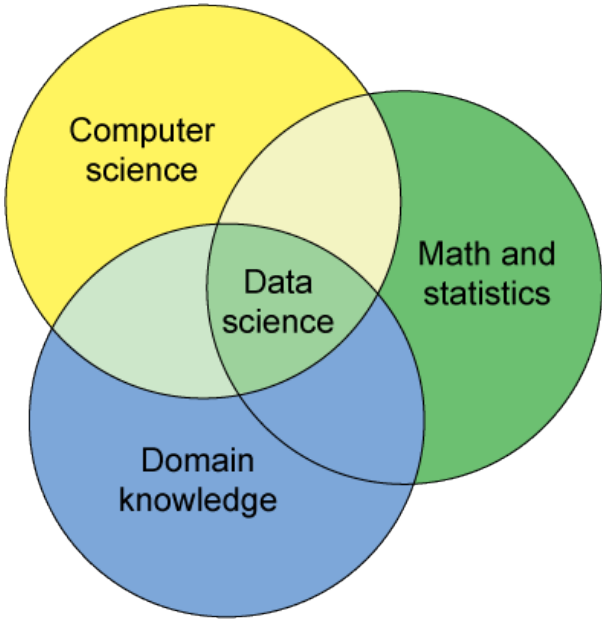
- Getting the big picture

Entity Resolution

- Learning how to solve a particular problem

Want to become a data scientist?

One definition



Making it More Specific

Domain Knowledge

- Use domain knowledge to ask questions and find related data
- E.g., Which customers are likely to leave your company?

No domain knowledge? Then, you have to have strong teamwork and communication skills

Math/Statistics

- Use math/statistics knowledge to come up with a solution
- E.g., Design an algorithm that can make the prediction based on enterprise data

Want to know more statistics? Read CH 5-7 in "Data Science from Scratch"

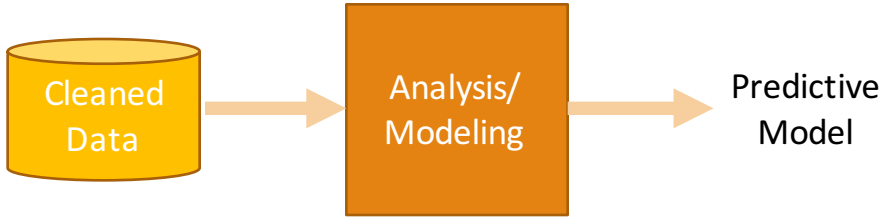
Computer Science

- Use programming skills to build a data-processing pipeline
- E.g., The pipeline takes enterprise data as input, and outputs the prediction

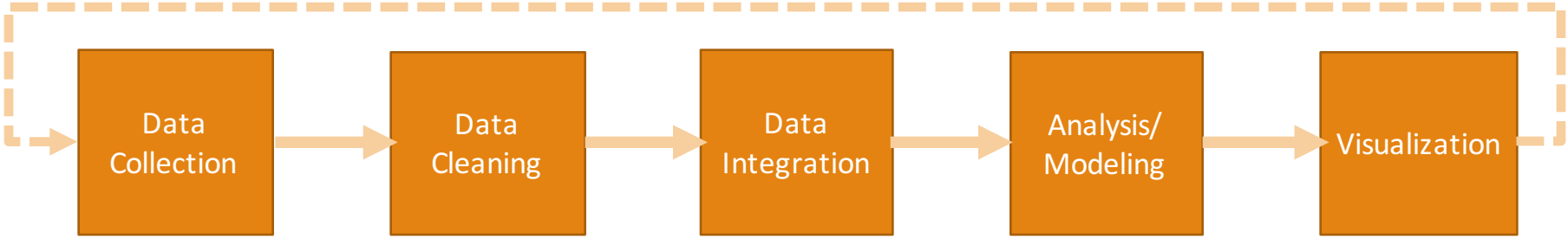
What is a data-processing pipeline?

Data-processing Pipeline

What you think you do?



What you really do?



Example: Assignment Mark Prediction

Data Collection

- Collect background information

Data Cleaning

- Missing values, Inconsistent values

Data integration

- Integrate with CourSys to get assignment marks

Modeling

- Build a linear regression model

Visualization

- Present results to non-technical persons

B	C
Country	**Background**
brazil	c++, computer graphics
Canada	
hong kong, canadian	
	ee, cisco, more hands-on
indian	cs, four years, oracle,
	enginerer
	sfu cs, eco, 10 years experience finance
indian	at&t, sql
canadian	
waterloo	cs
toronto	astrophysics,

Outline

Motivation

- Asking “Why” before you learn something

Data Cleaning and Integration

- Getting the big picture

Dirty Data Problems

From Stanford Course:

1) Parsing text into fields (separator issues)
2) Missing required field (e.g. key field)
3) Different representations (iphone 2 vs iphone 2nd generation)
4) Fields too long (get truncated)
5) Formatting issues – especially dates
6) Licensing issues/Privacy/ keep you from using the data as you would like?

Data Cleaning Tools

Python

- Missing Data (Pandas)
- Deduplication (Dedup)

OpenRefine

- Open-source Software (http://openrefine.org)
- OpenRefine as a Service (RefinePro)

Data Wrangler

- The Stanford/Berkeley Wrangler research project
- Commercialized (Trifacta)

Data Integration Problem

Data Source 1 (from CourSys)

First Name	Last Name	Mark
Michael	Jordan	50
Kobe	Bryant	48

Data Source 2 (from survey)

Name	Background
Mike Jordan	C++, CS, 4 years
Kobe Bryant	Business, 2 years

Data Integration???

Integrated Data

Name	Mark	Background
Michael Jordan	50	C++, CS, 4 years
Kobe Bryant	48	Business, 2 years

Data Integration: Three Steps

Schema Mapping

- Creating a global schema
- Mapping local schemas to the global schema

Entity Resolution

- You will learn this in detail later

Data Fusion

- Resolving conflicts based on some confidence scores

Want to know more?

- Anhai Doan, Alon Y. Halevy, Zachary Ives. Principles of Data Integration. Morgan Kaufmann Publishers, 2012.

Outline

Motivation

- Asking “Why” before you learn something

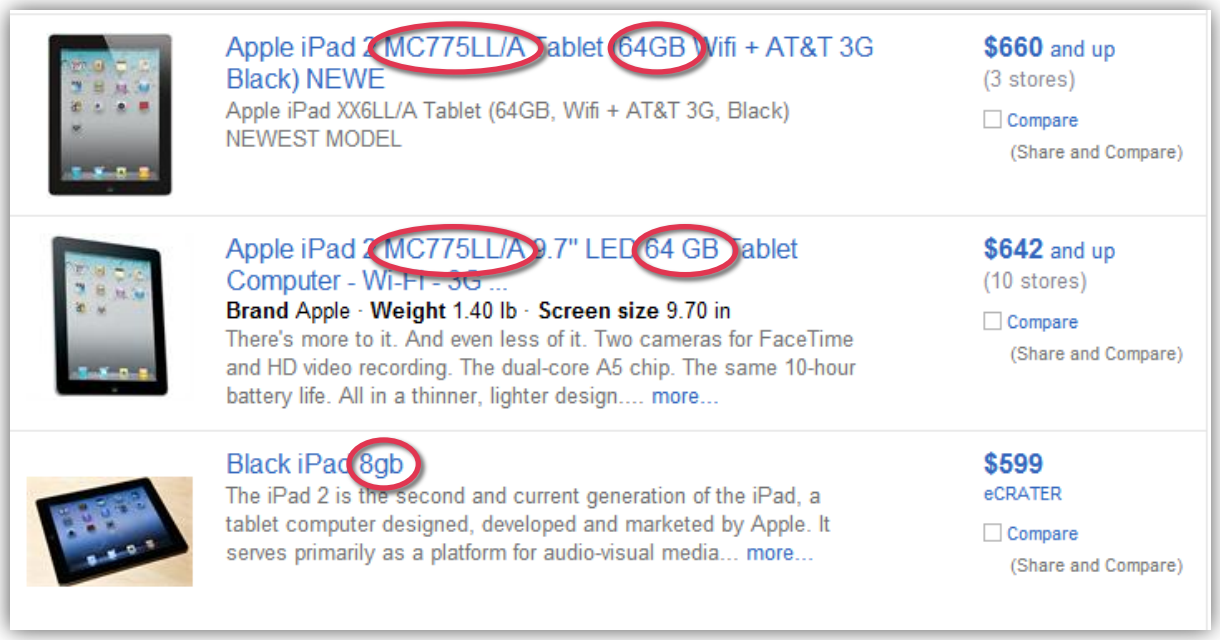
Data Integration and Cleaning

- Getting the big picture

Entity Resolution

- Learning how to solve a particular problem

Entity Resolution



Output of Entity Resolution

ID	Product Name	Price
r_1	iPad Two 16GB WiFi White	$490
r_2	iPad 2nd generation 16GB WiFi White	$469
r_3	iPhone 4th generation White 16GB	$545
r_4	Apple iPhone 3rd generation Black 16GB	$375
r_5	Apple iPhone 4 16GB White	$520

$(r_1, r_2), (r_3, r_5)$

Entity Resolution Techniques

Similarity-based

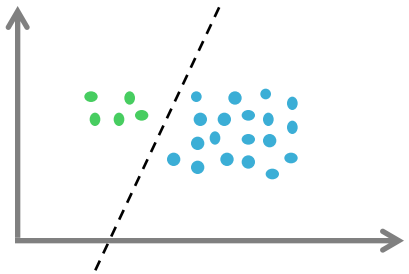
- Similarity Function (e.g., $\text{Jaccard}(r, s) = |\frac{r \cap s}{r \cup s}|$)
- Threshold (e.g., 0.8)

Jaccard(r1, r2) = 0.9 ≥ 0.8 Matching

Jaccard(r4, r8) = 0.1 < 0.8 Non-matching

Learning-based

- Represent a pair of records as a feature vector



Similarity-based

Suppose the similarity function is Jaccard.

Problem Definition

Given a table T and a threshold θ, the problem aims to find all record pairs $(r, s) \in T \times T$ such that $\mathrm{Jaccard}(\mathrm{r}, \mathrm{s}) \geq \theta$

The naïve solution needs n^2 comparisons

Filtering-and-Verification

Step 1. Filtering

- Removing obviously dissimilar pairs

Step 2. Verification

- Computing Jaccard similarity only for the survived pairs

How Does Filtering Work?

What are "obviously dissimilar pairs"?

- Two records are obviously dissimilar if they do not share any word.
- In this case, their Jaccard similarity is zero, thus they will not be returned as a result and can be safely filtered.

How can we efficiently return the record pairs that share at least one word?

- To help you understand the solution, let's first consider a simplified version of the problem, which assumes that each record only contains one word

A simplified version

Suppose each record has only one word. Write a SQL query to do the filtering.

r_1	Apple
r_2	Apple
r_3	Banana
r_4	Orange
r_5	Banana

```
SELECT T1.id, T2.id
FROM Table T1, Table T2
WHERE T1.word = T2.word and T1.id < T2.id
```

Does it require n^2 **comparisons ?**

Output: (r1, r2), (r3, r5)

A general case

Suppose each record can have multiple words.

r_1	Apple, Orange
r_2	Apple
r_3	Banana
r_4	Orange, Apple
r_5	Banana

Flatten →

r_1	Apple
r_1	Orange
r_2	Apple
r_3	Banana
r_4	Orange
r_4	Apple
r_5	Banana

1. This new table can be thought of as the **inverted index** of the old table.
2. **Run the previous SQL on this new table** and remove redundant pairs.

Not satisfied with efficiency?

Exploring stronger filter conditions

- Filter the record pairs that share zero token
- Filter the record pairs that share one token
- ....
- Filter the record pairs that share k tokens

Challenges

- How to develop efficient filter algorithms for these stronger conditions?

Jiannan Wang, Guoliang Li, Jianhua Feng.
Can We Beat The Prefix Filtering? An Adaptive Framework for Similarity Join and Search.
SIGMOD 2012:85-96.

Not satisfied with result quality?

TF-IDF

- Use weighted Jaccard: $\mathbf{WJaccard}(r, s) = \frac{wt(r \cap s)}{wt(r \cup s)}$

Learning-based

- Model entity resolution as a classification problem
- How to generate feature vectors?

M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In KDD, pages 39–48, 2003

Crowdsourcing

- Build a hybrid human-machine system (like Iron Man) for entity resolution

Summary

Data-processing Pipeline

- Data Collection → Data Cleaning → Data Integration → Modeling → Visualization

Data Cleaning

- Dirty Data Problems, Data-cleaning tools

Data Integration

- Schema Mapping, Entity Resolution, Data Fusion

Entity Resolution

- Filtering-and-Verification Framework (avoid n^2 pair comparisons)
- How to further improve efficiency and result quality

Assignment 4: Entity Resolution

Part 1. Similarity Joins (required)

Part 2. Where To Go From Here (optional)

Deadline: 11:59pm, Feb 5th
http://tiny.cc/cmpt733-a4

About Final Project

My thoughts

- You have to integrate data from at least two data sources
- You have to come up with some interesting questions to investigate
- You have to give an excellent talk on your project proposal
- …

You will get a detailed project instruction in two weeks