Graph Embeddings

MACC Conference

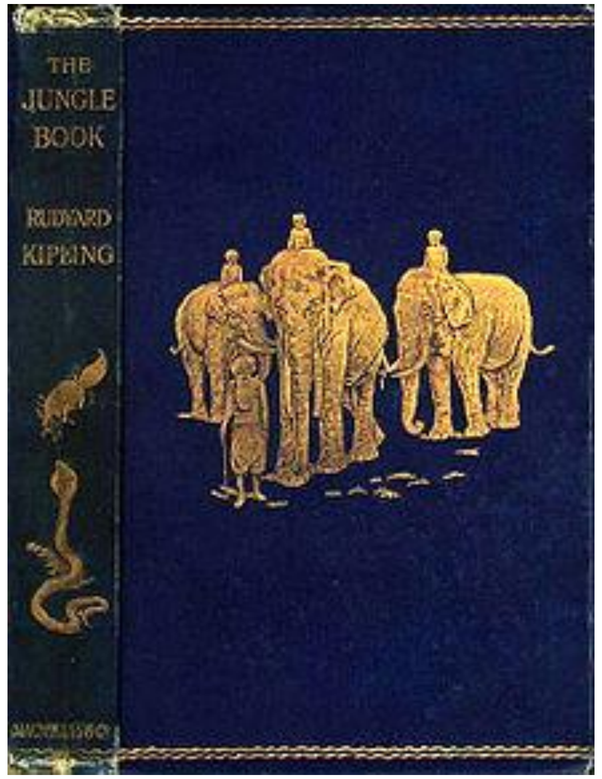
November 5th, 2020

midwest architecture community collaboration

Dan McCreary
Distinguished Engineer

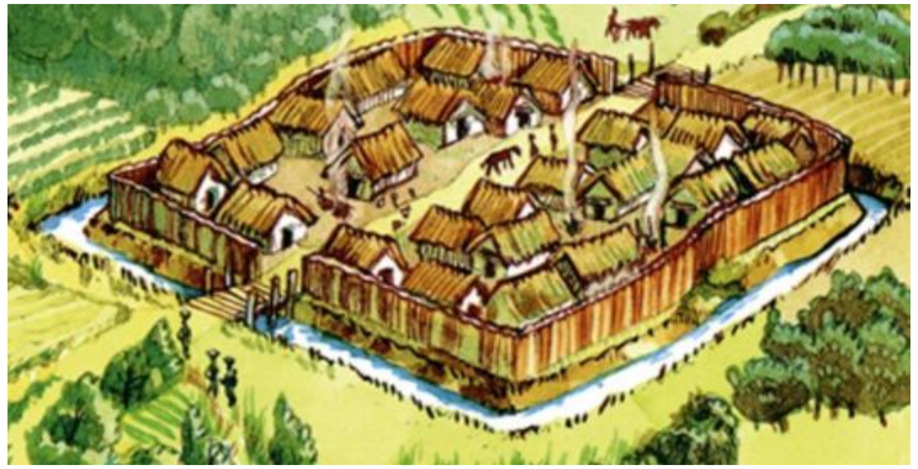
OPTUM® Advanced Technology Collaborative

Storytelling – The Setting



The Jungle Book
by Rudyard Kipling

Mowgli



Walled Village

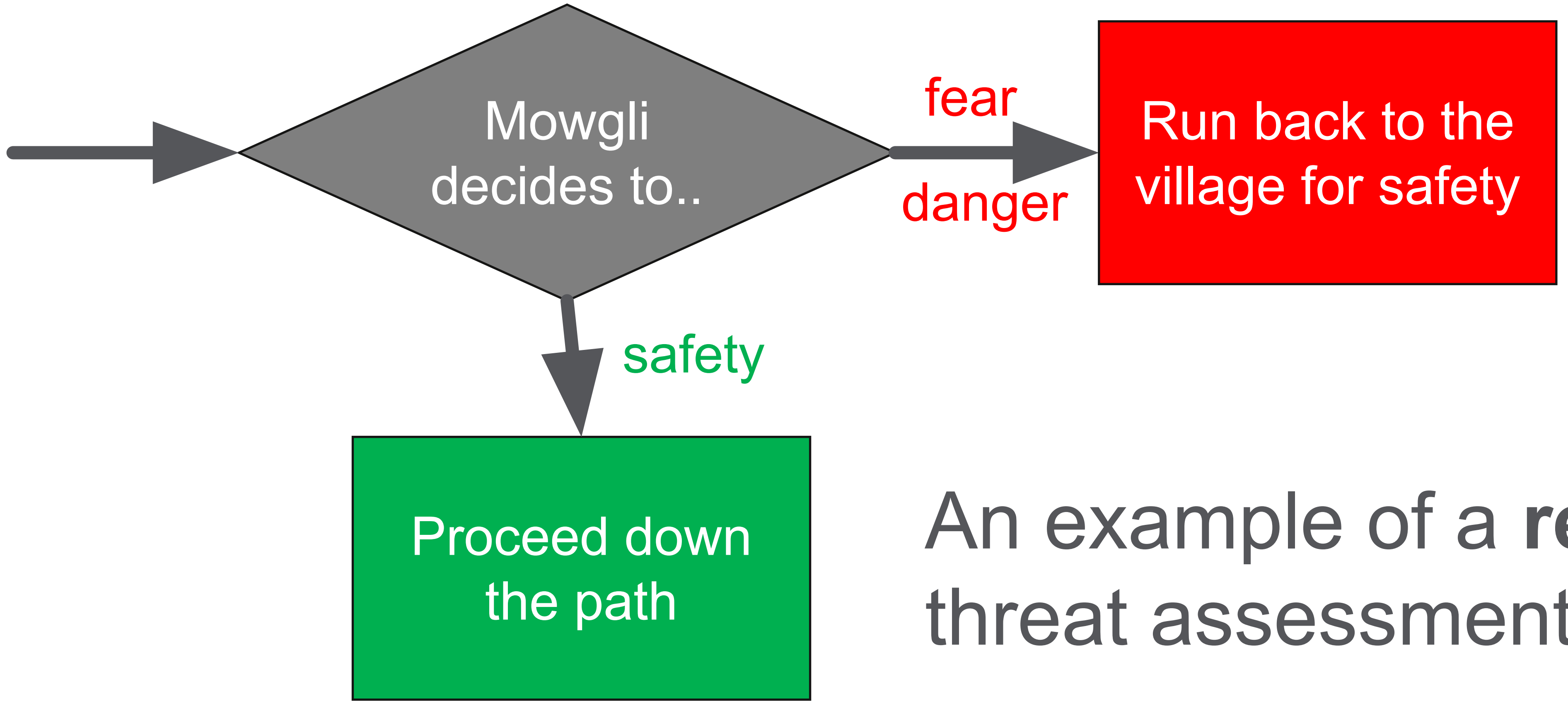
Mowgli's Kitten

Path Near Village

While Walking Down the Path…

Mowgli sees a …

…which is very **similar** to Mowgli's Kitten



Think
Parallel
Processing

An example of a **real-time** existential threat assessment.

OPTUM®

Acme E-Commerce Inc.

A shopper is looking for a baby shower gift

Your product **recommendation** engines suggests this lovely flame thrower!

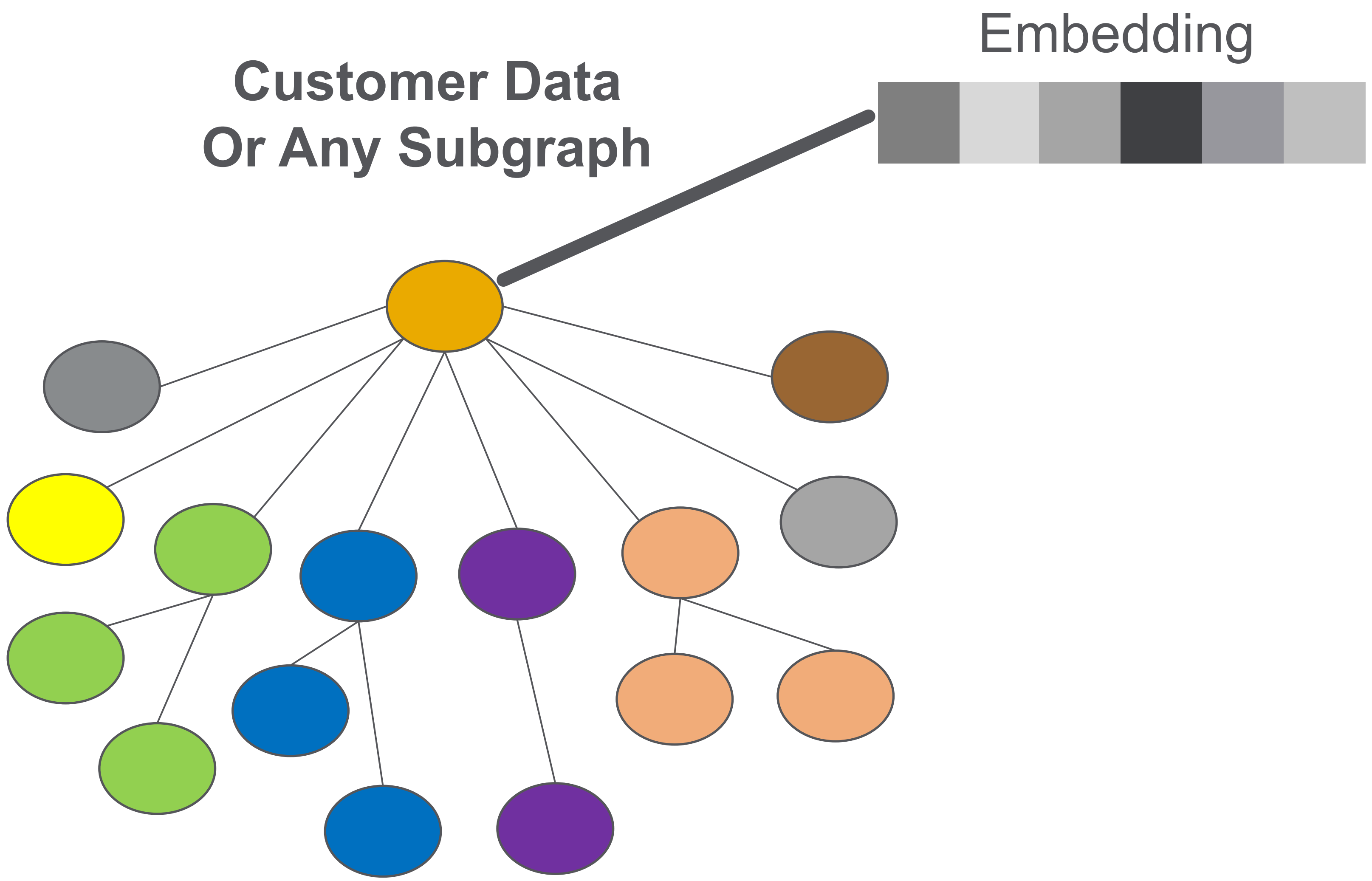
Real-time **similarity** is an existential threat for companies that are not recommending the right products!

Agenda

1. What are graph **embeddings**?
2. What do we mean by **real-time**?
3. What is **Serial** vs. **Parallel** computation?
4. What do we mean by “**similar**”?
5. How do we **use** embeddings in applications?
6. How do we **create** and maintain embeddings?
7. Why are graph embeddings **universal**?
8. How can systems **integrate** them?

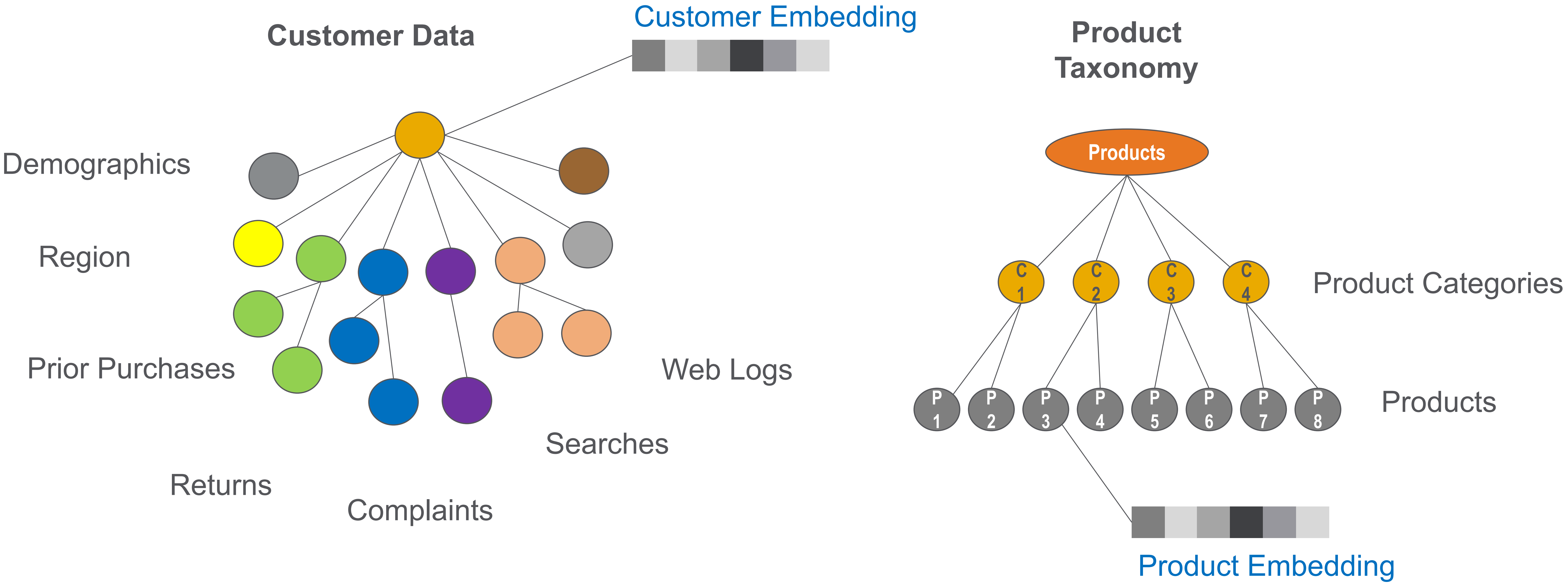
OPTUM®

What is a Graph Embedding?



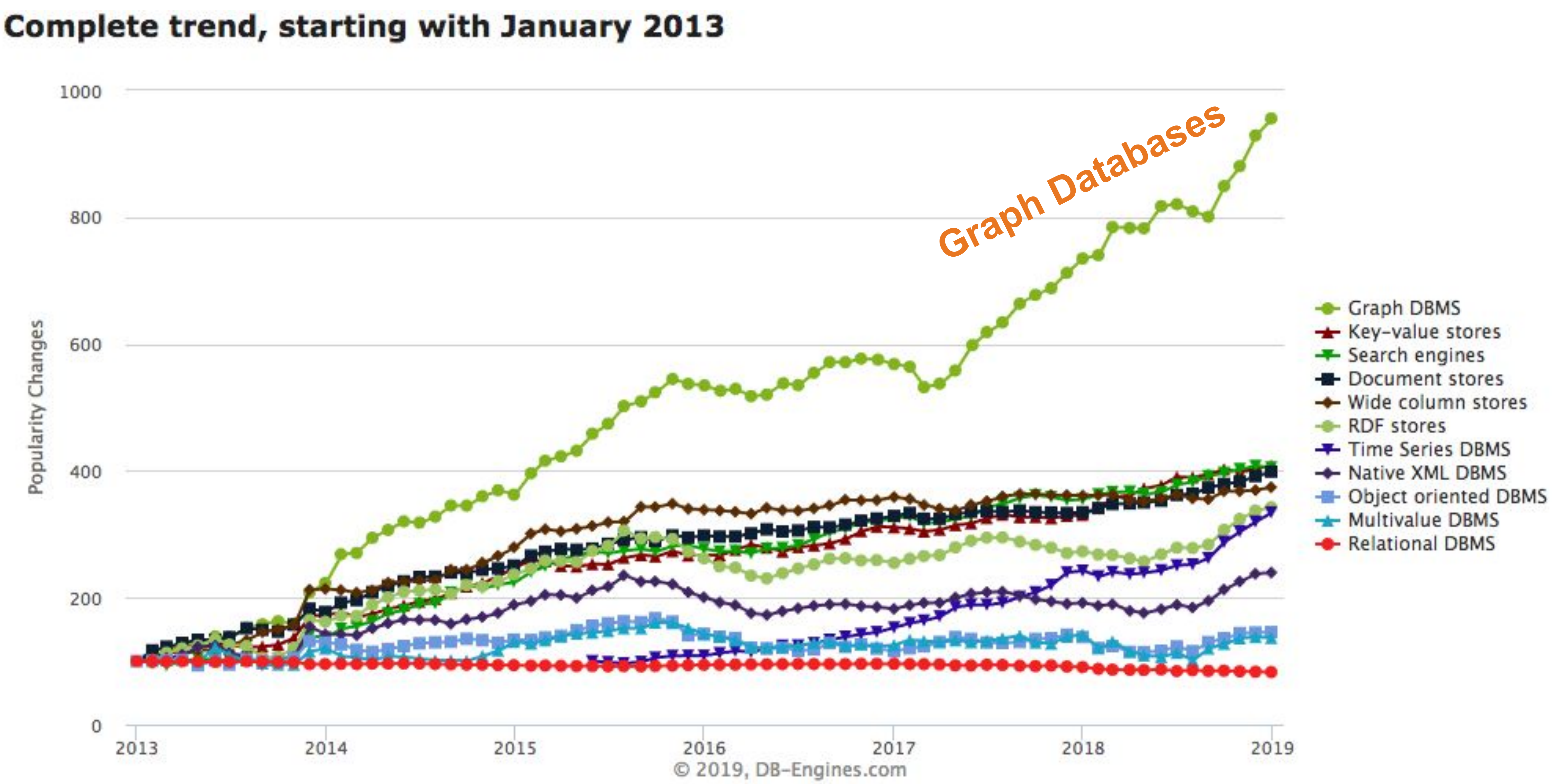
- A data structure used to create fast similarity calculations
- Usually stored as a fixed-length vector of scalers
- Optimized for fast parallel comparison

Product Recommendation



Given a **context**, what similar customers buy what similar products?

Graph Databases are HOT!

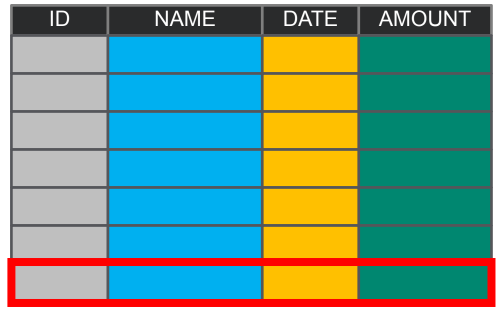


https://db-engines.com/en/ranking_categories

OPTUM®

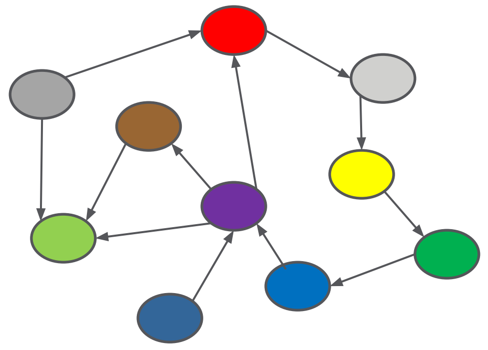
Relational vs. Graph

Relational (row store)



1. Atomic unit of storage is a **row** of a table and data is appended to a table one row at a time
2. All **columns** within a table must have the same structure and no variations within a table are allowed
3. Table structures are **fixed** after design – all rows have the same structure
4. Relationship traversal is done via JOINs at runtime using log(N) **search's** calculated at **query** time
5. Difficult to distribute over a cluster of 100+ nodes

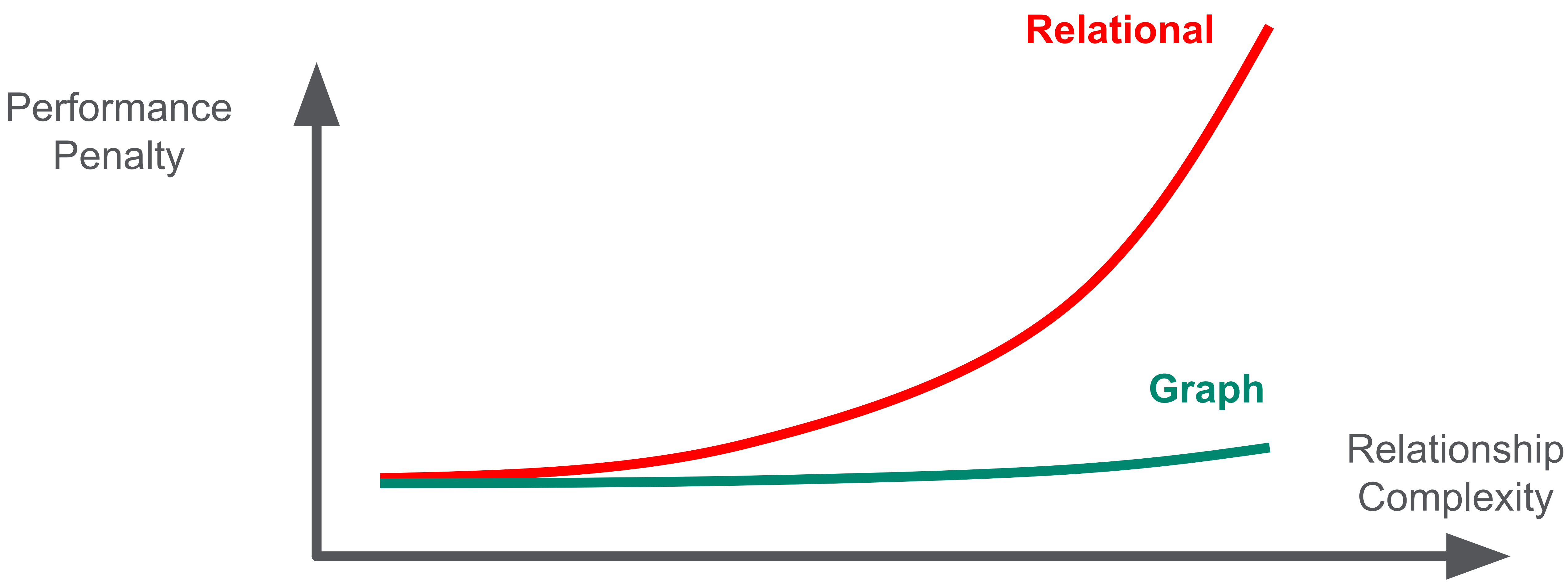
Graph



1. Atomic unit of storage are **nodes** and **edges**
2. Each node and edge may have independent properties that are determined at run time (schema agnostic)
3. Joins between nodes and edges are computed at **load** time and are stored as memory pointers
4. Relationships traversal is done using pointer hopping – each core can **evaluate 2M hops per second**
5. Distributed graph products are new

OPTUM®

Low Complexity Penalty



- Complex systems are easy to model with graph databases
- The world is complex
- Highly normalize models are a precise representations of the world
- We still need to compare things, even if they are complex and have many relationships

OPTUM®

Comparing Two Colors

Gold = rgb(255, 215, 0)

Orange = rgb(255,165, 0)

Features

Label	Red	Green	Blue
Gold	255	215	0
Red	255	165	0
Differences	0	60	0

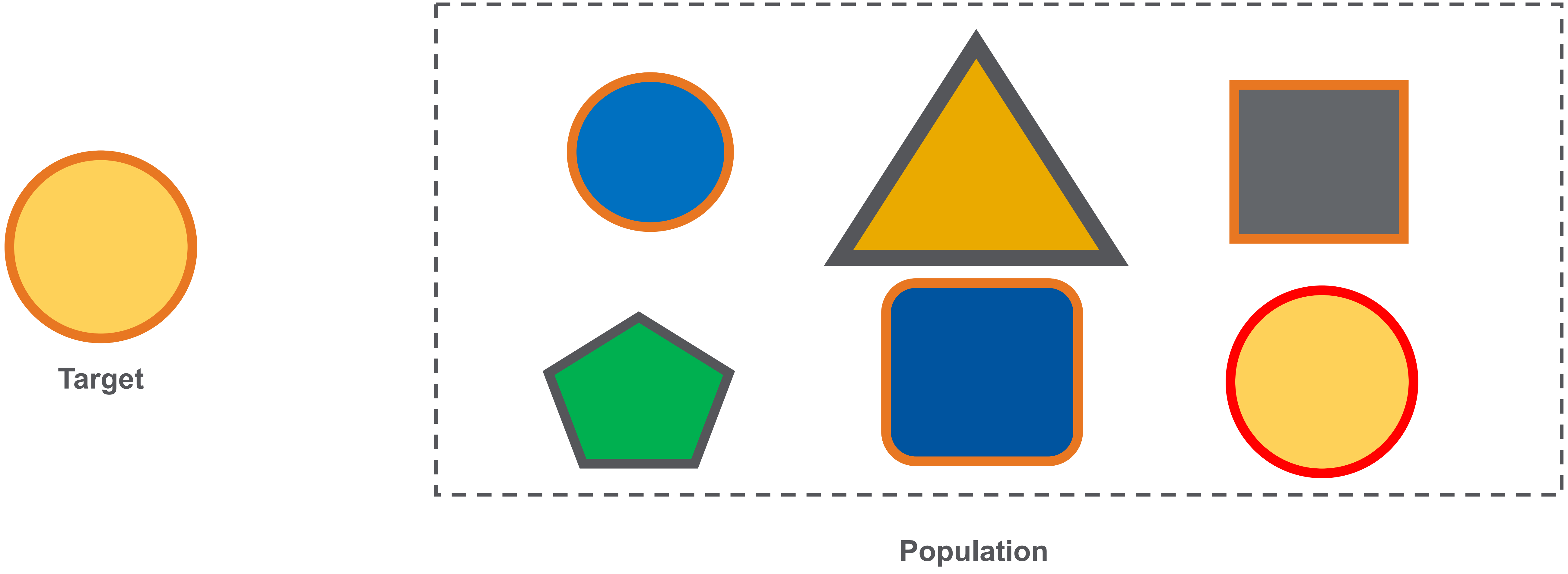
$\sum$ Differences => 0 + 60 + 0 = **60**

Sum

Non-normalized **Explainable**
Objective Measure of **Similarity**

OPTUM®

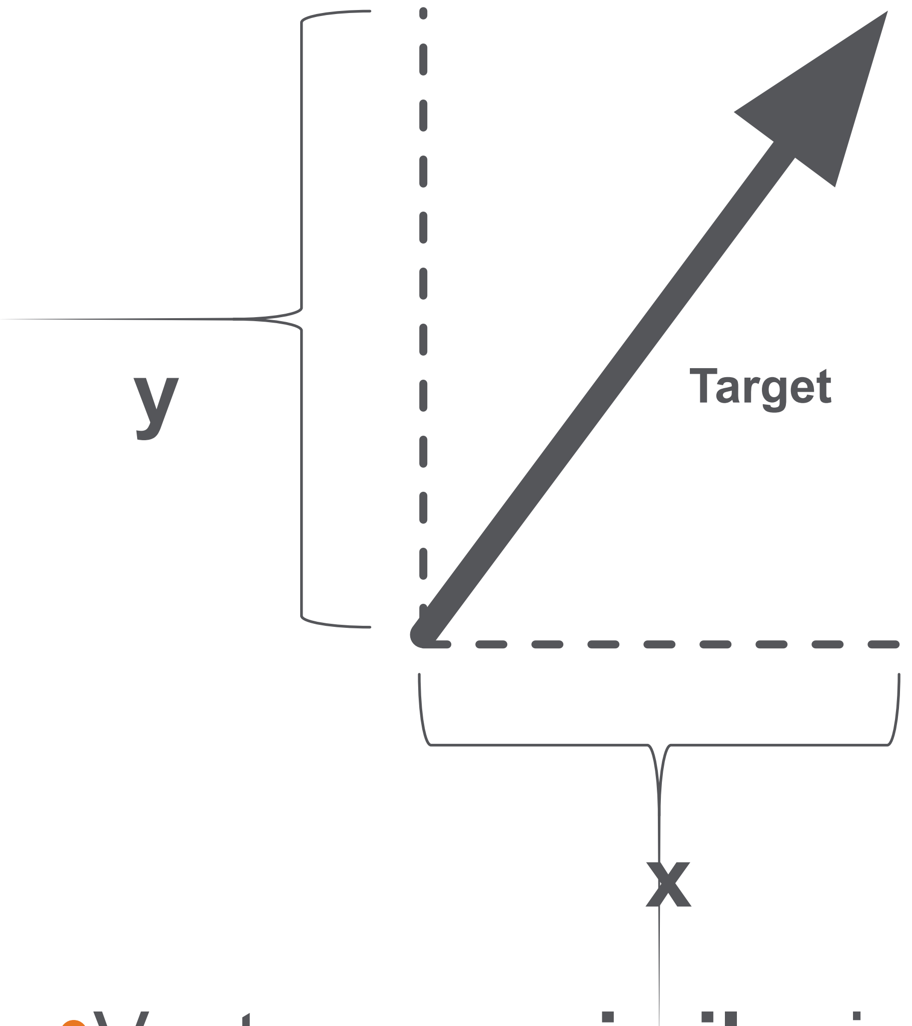
Property-based Similarity Example

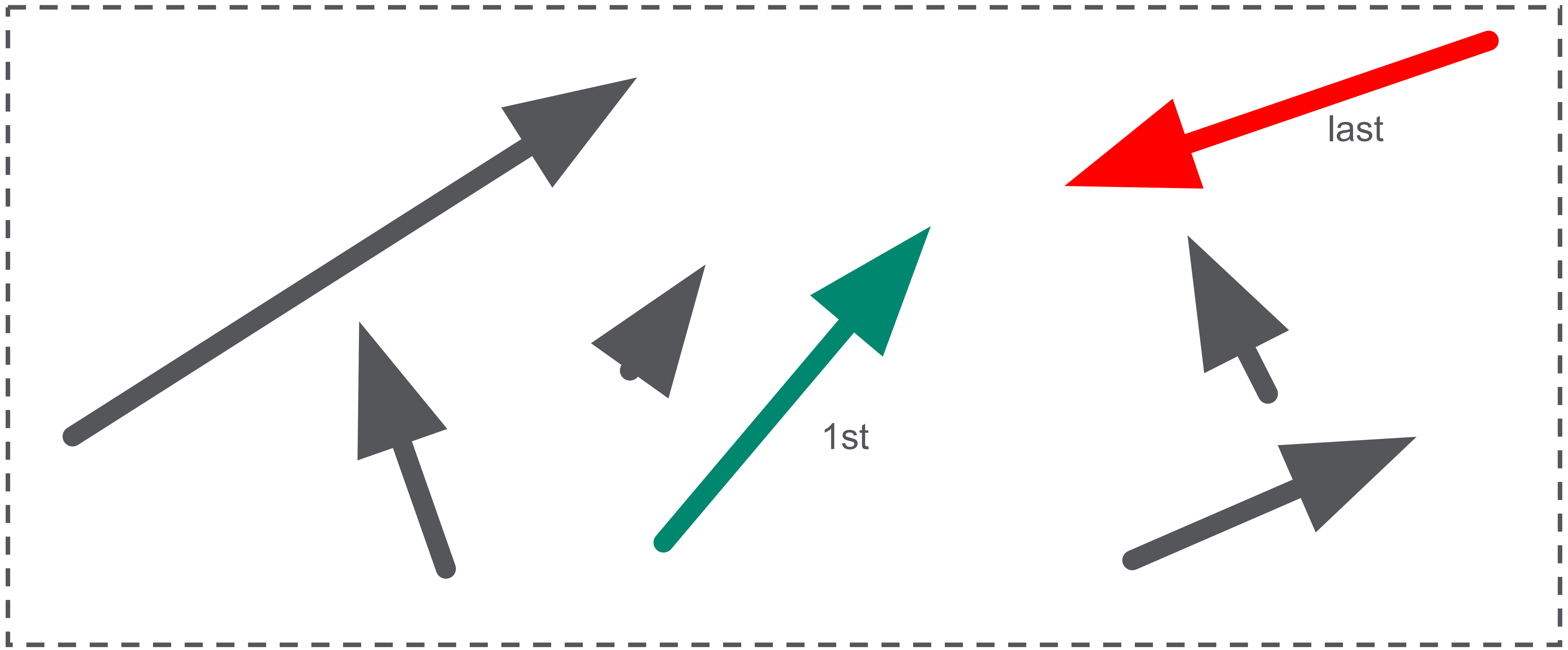


- Used to find the most similar items in a graph by comparing **properties and structure**
- Ideal when you a can compare individual features of an item **numerically**
- Algorithms return a **ranking** of similarity between a target and a population based on the counts and weights of properties that are similar

OPTUM®

Vector Similarity





Population

- Vectors are **similar** in two-dimensional space if they have the same length and direction
- Compare all the “x” lengths and the “y” lengths and **rank** by the sum of the totals of the difference

Vector Representation

$$\begin{pmatrix} x=8 \\ y=10 \end{pmatrix} \qquad \begin{pmatrix} 16 \\ 16 \end{pmatrix} \begin{pmatrix} 8 \\ 6 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \end{pmatrix} \begin{pmatrix} 7 \\ 9 \end{pmatrix} \begin{pmatrix} -4 \\ 6 \end{pmatrix} \begin{pmatrix} -4 \\ 6 \end{pmatrix} \begin{pmatrix} -12 \\ -8 \end{pmatrix}$$

Target Vector

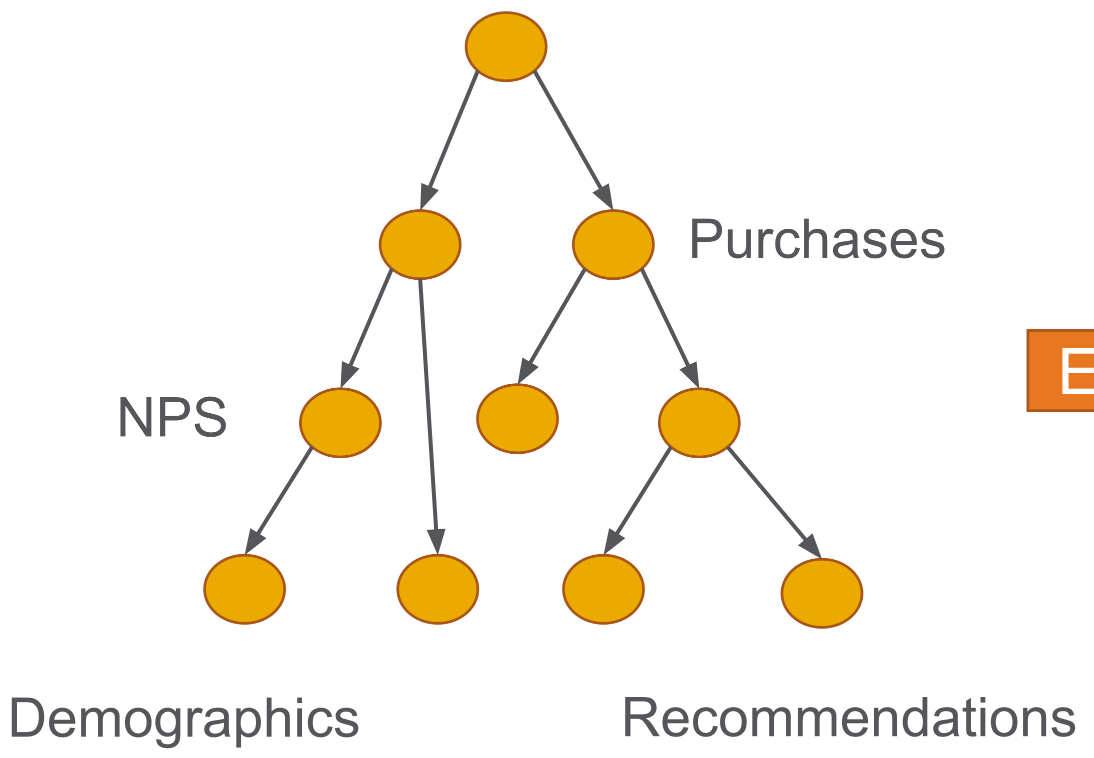
Population Vectors

- Each item can be represented by a series of “feature” vectors
- The numbers are scalers

OPTUM®

Graph to Vector

Target
Customer Graph



Extract

Key Features

Gender_Code
Age
Rural_Urban_Score
LTR_Score
Lifetime_Value
CohortCode1
CohortCode2
ProductGroup1
ProductGroup2
ProductGroup3

…200…features

Convert

Vector Representation

1
47
4
8
9
2
8
18
13
67

…200…integers

Features should represent both the **properties** and **structure** of your customer's graph

OPTUM®

Example: Customer Similarity

Target Customer Vector

$$\begin{bmatrix} 0 \\ 47 \\ 4 \\ 8 \\ 9 \\ 2 \\ 8 \\ 18 \\ 13 \\ 67 \\ \dots \\ \dots \\ \dots \end{bmatrix}$$

Cosine similarity

$$\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Sample Population Vectors

$$\begin{bmatrix} 0 \\ 43 \\ 2 \\ 13 \\ 7 \\ 1 \\ 8 \\ 18 \\ 13 \\ 67 \\ \dots \\ \dots \\ \dots \end{bmatrix} \quad \begin{bmatrix} 1 \\ 45 \\ 4 \\ 8 \\ 9 \\ 3 \\ 8 \\ 15 \\ 13 \\ 45 \\ \dots \\ \dots \\ \dots \end{bmatrix} \quad \begin{bmatrix} 1 \\ 22 \\ 4 \\ 3 \\ 9 \\ 2 \\ 8 \\ 18 \\ 12 \\ 32 \\ \dots \\ \dots \\ \dots \end{bmatrix}$$

Similarity Score: .80 .85 .92

- Graph **Similarity** algorithms allows graph databases to quickly compare current items with many other items
- For any given customer, we can use a fast, in-memory similarity algorithm to compare the key features of any patients to a larger population
- Cohort codes can be **pre-calculated** to quickly narrow the sample population to a smaller group
- This calculation is considered a “embarrassingly parallel” query and could be accelerated by adding more nodes to a cluster
- Specialized graph hardware such as GPUs and FPGAs can dramatically accelerate these calculations

The Curse of Dimensionality

…various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.

https://en.wikipedia.org/wiki/Curse_of_dimensionality

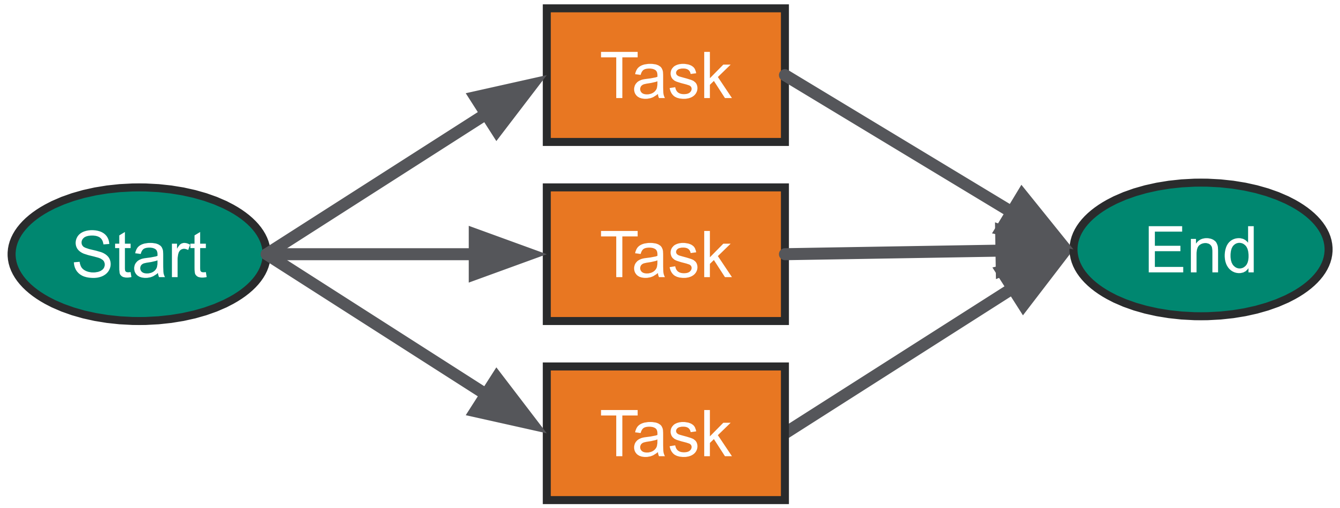
https://en.wikipedia.org/wiki/Dimensionality_reduction

Serial vs. Parallel Graph Algorithms

Serial Graph Algorithms

- One task cannot begin before the prior task is complete
- Task order is important
- Serial Algorithms cannot easily be optimized on FPGAs

Parallel Graph Algorithms



- Many tasks can be done independently
- Task order in not relevent
- Tasks can usually be done faster on FPGAs

Does the Human Brain do Serial or Parallel Computation?

- The following slide photos of two people
- One is a famous actor
- The other is a synthetically generated image of a person (generated by a GAN)
- Shout out “**Left**” or “**Right**” as soon as you can tell which is the **famous actor**

OPTUM®

What just happened

1. The visual cortex received the images as electrical signals from your eyes
2. Your brain identified key **features** of each face from the images - in **parallel**
3. Your brain sent these features as electrical signals to your memories of people's faces
4. Your brain compared these features to **every memory you have of a person's face** – in **parallel**
5. Your brain sent their recognition scores to a control center of your brain
6. Your brain's speech center vocalized the word "right" – in **series**

Answer: The human brain, comprised of around 84B neurons, does **both** parallel and series calculations

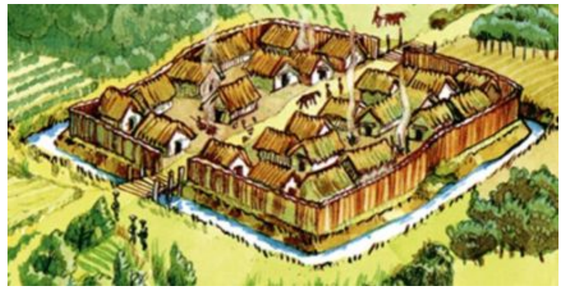
Two Key Questions:

1. How does the brain know to pay **attention** to specific features of a face?
2. What portions of real-time recommendation systems can be done cost effectively in **parallel** at low cost?

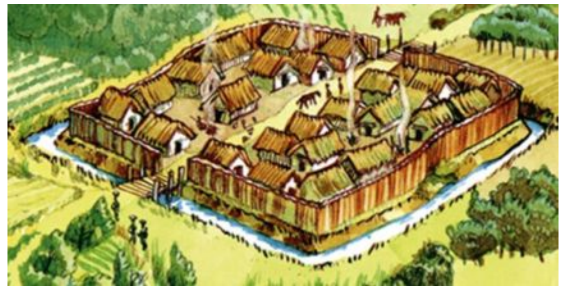
The Neighboring Village Problem

Mowgli

Sees a new person on the path



My Village



Other Village

- Is this person from my village or a nearby village?
- How similar is this face to someone I know in my village?
- Is this person a threat?

OPTUM®

The Product Recommendation Challenge

1. A customer comes to your web site
2. You have information on their prior purchases, and they view new products
3. How **quickly** can you generate a real-time recommendation based on 10 products and 1 million product reviews?
4. How much **detail** can you consider and return the best match in 100 milliseconds?

How quickly can we compare a given customer and their purchases to 1 million other customers and their purchases?

What comparison tasks can be done in parallel?

Don't Delay a Calculation Over 200 milliseconds (2/10th of a second)

- 47 percent of visitors **expect** a website to load in less than 2 seconds
- 40 percent of visitors will **leave** the website if the loading process takes more than 3 seconds.

According to Kissmetrics http://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/

OPTUM®

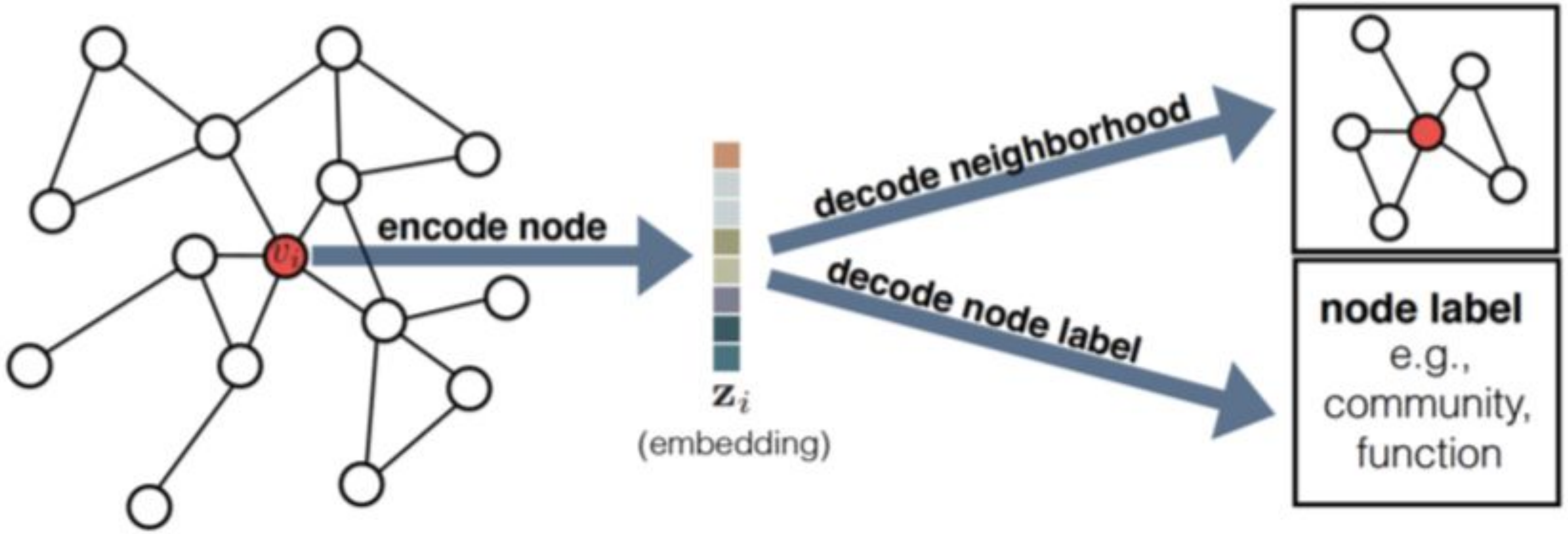
The Rise of Automatic Graph Feature Engineering

Recent years have seen a surge in approaches that ***automatically*** *learn to encode graph structure into low-dimensional* ***embeddings****.*

The central problem in machine learning on graphs is finding a way to incorporate information about the ***structure*** *of the graph into the machine learning model.*

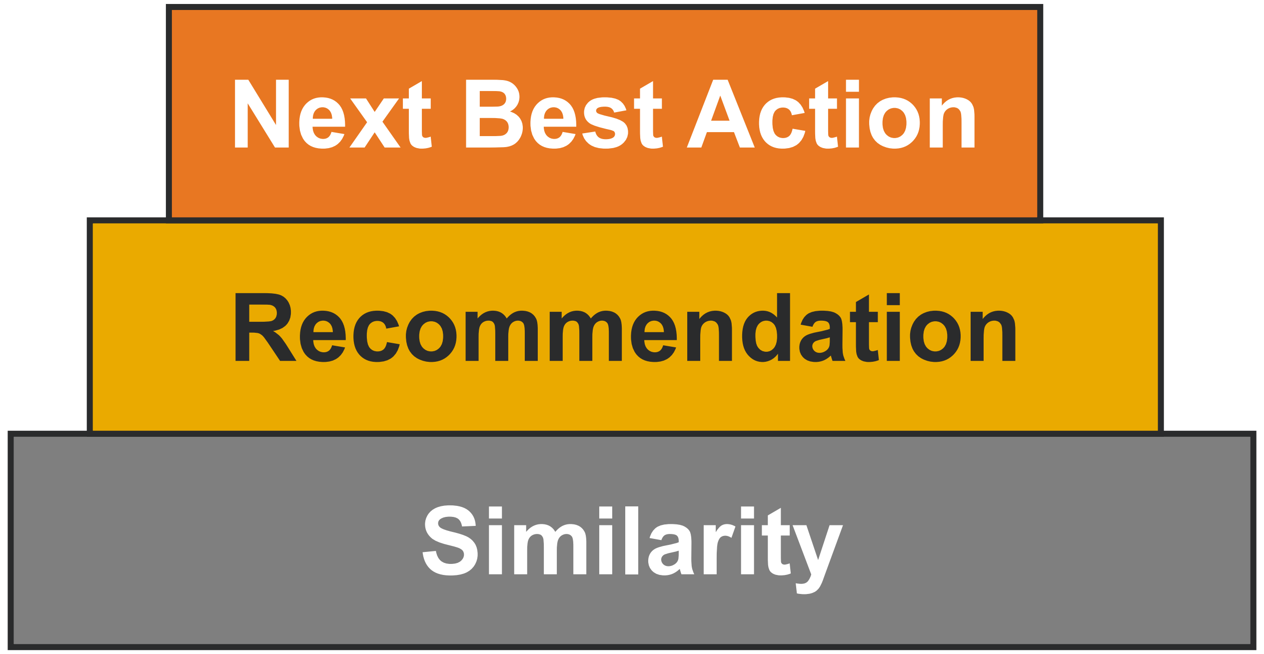
From Representation Learning on Graphs: Methods and Applications by Hamilton et. El.

Example of Graph Embedding – Encode and Decode



From Representation Learning on Graphs: Methods and Applications by Hamilton et. El.

Why Are Similarity Calculations Critical?



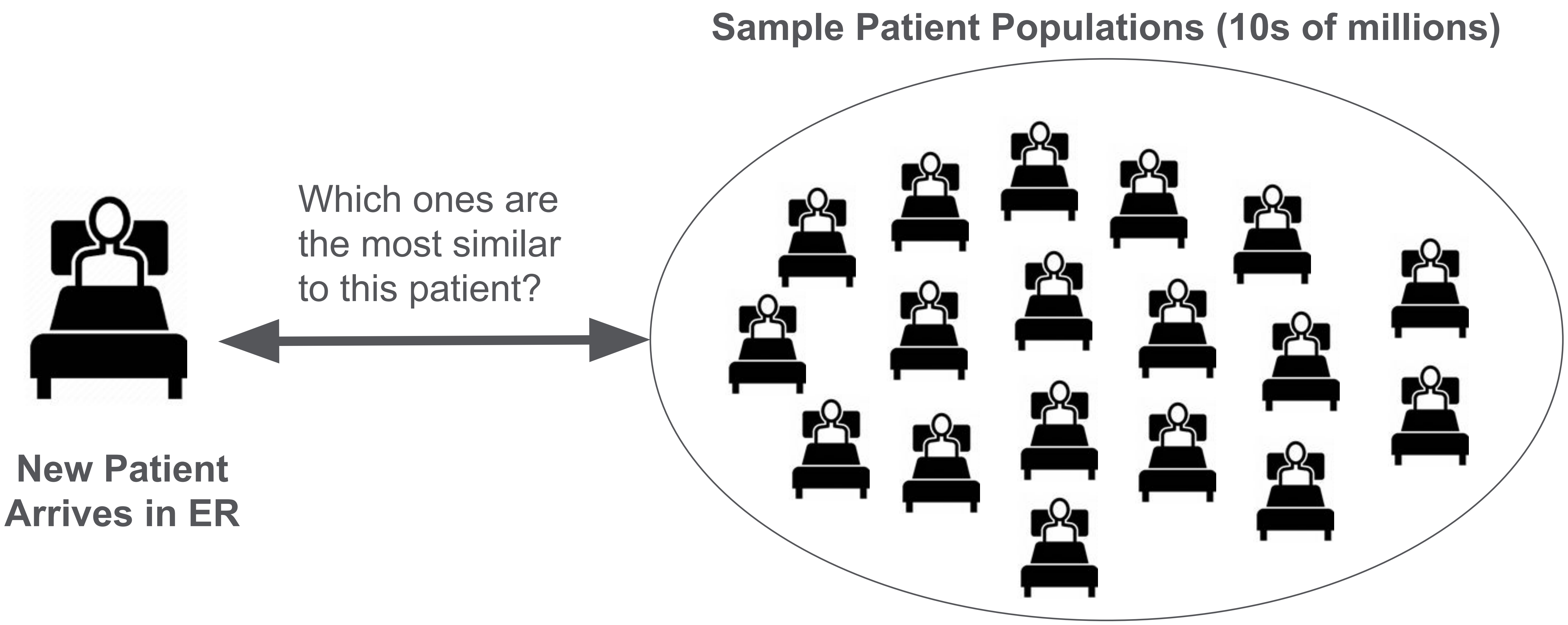
- Similarity is at the foundation of recommendation engines
- Recommendation engines power sites like:
  - Google – recommend a document
  - NetFlix™ – recommend a movie
  - Amazon - recommend a product
  - Pintrest™ – recommend an interest
  - Healthcare – recommend a care path

Recommendations must take into account many factors including recent searches

To be useful in interactive web sites we set a goal of response times of under 200 milliseconds

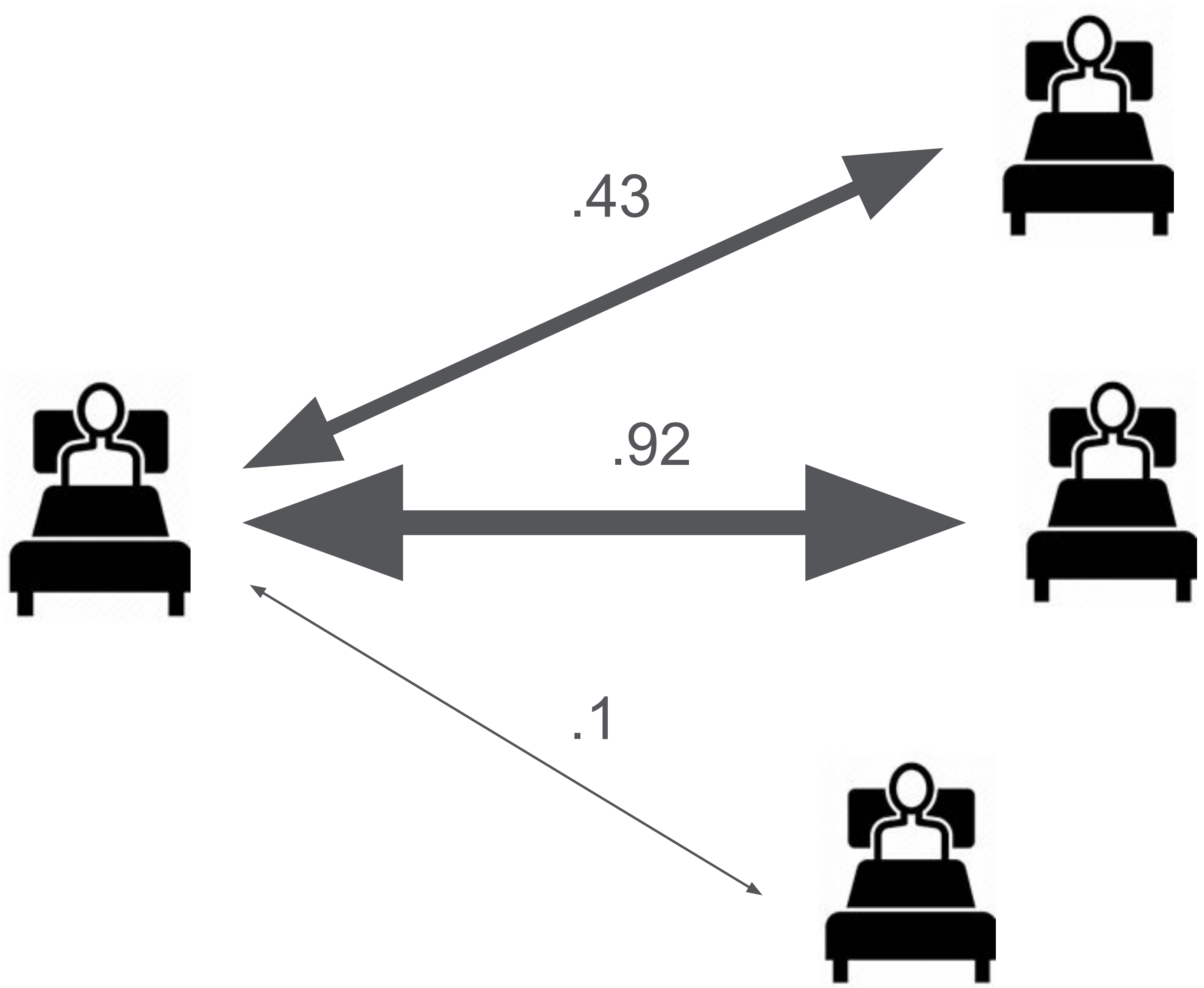
OPTUM®

Real-Time Patient Similarity



- Given a new patient that arrives a clinical setting, how can we **quickly** find the most similar patients?
- **Assumption:** we have 10M clinical records of our population of 235 million members
- Can we find the **100 most similar patients in under 200 milliseconds**?

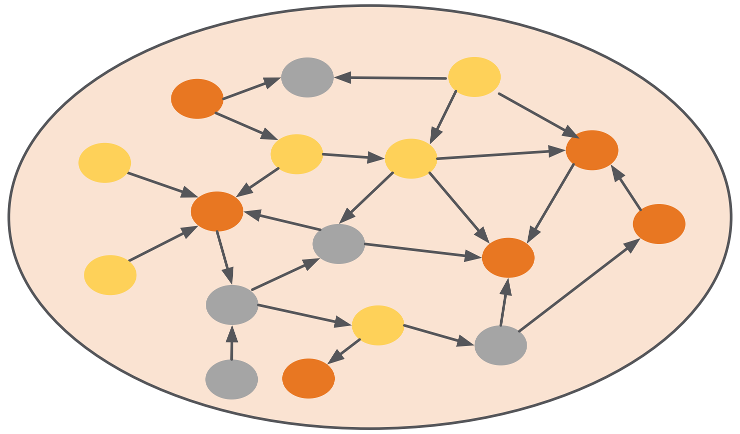
Similarity Score – A scaled measure of "alikeness" for a context



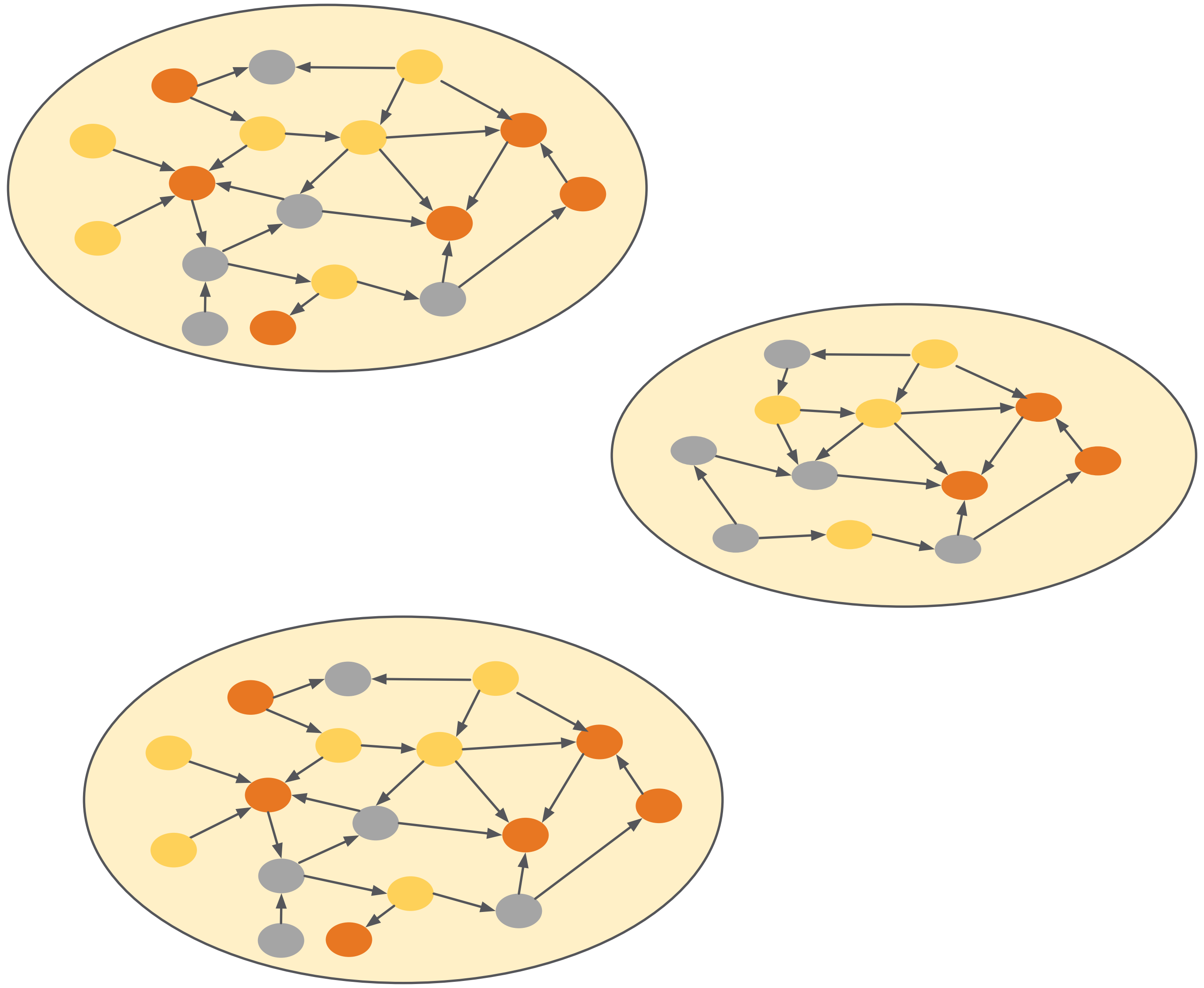
- A single **scaled** dimension of comparison for a given setting or context
- Comparing a patient to itself would have a similarity score of 1.0
- Patients that have few common characteristics would have a score of 0.1

Graph Representation of Patients – Includes Structure

Target Patient



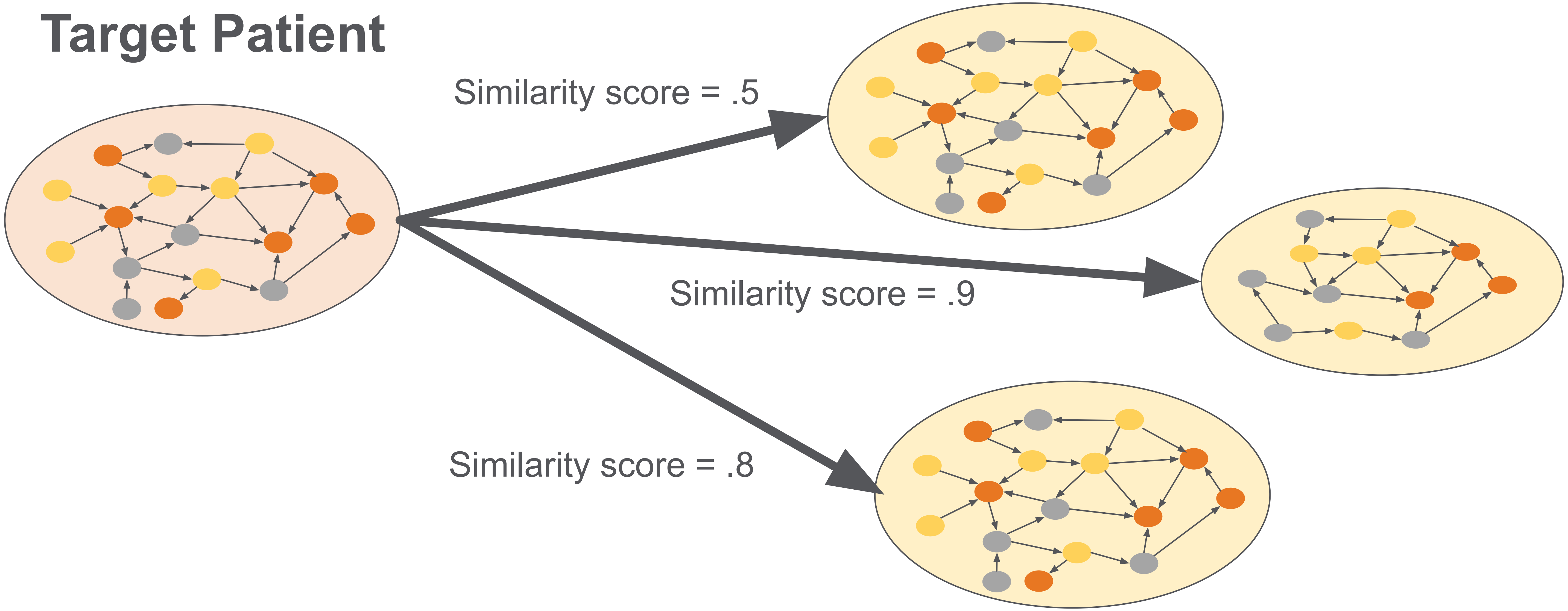
Sample Patient Population (10M)



Graph Representation of Patients – Includes Structure

Sample Patient Population (10M)

Target Patient



How can I **quickly** compare these graphs and find the most similar patients?

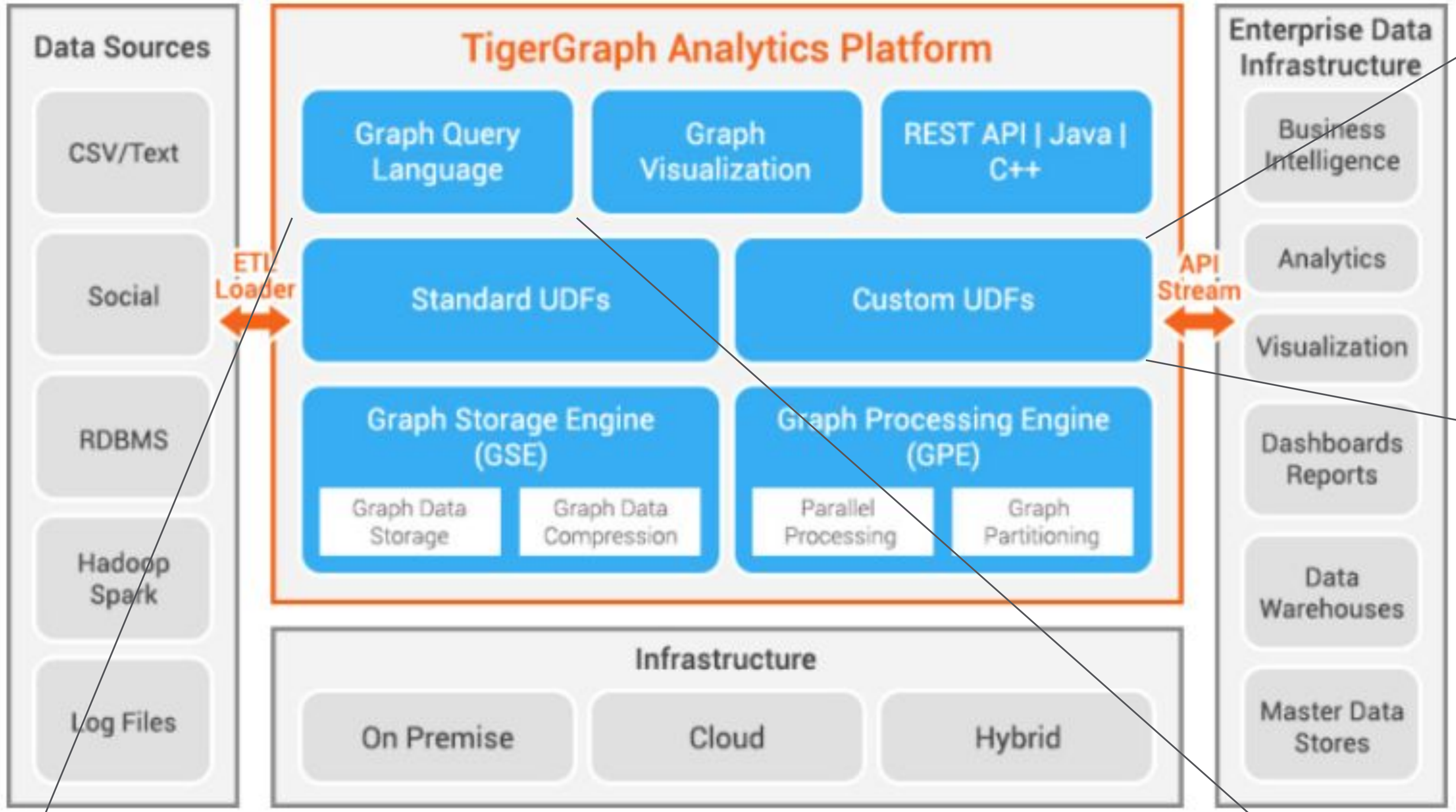
Xilinx FPGA Implementation

Kumar Deepak
Distinguished Engineer
Xilinx

- The more items to compare, the longer it takes to fine the top 100 most similar items
- FPGA = Field Programable Gate Array
- 30B transistors that can be rewired in 2 seconds using “C” langauge
- Ideal for parallel computation
- Benchmarked FPGA implementation of cosine similarity
- Open-source versions coming soon

Note: AMD (90B) recently announced it will acquire Xilinx (30B)

Integration with TigerGraph



<TG Install Dir>/dev/gdk/gsql/src/QueryUdf/ExprFunctions.hpp:
inline string udf_open_alveo(int mode) { … }
inline bool udf_close_alveo(int mode) { … }
inline bool udf_write_device(int mode) { … }
inline ListAccum<testResults>
udf_cosinesim_ss_alveo(ListAccum<int64_t>& patient_vector, uint64_t topK) { … }

libxilinxsimilarity.so (code to manage client requests)

Xilinx Runtime (PCIE driver and card management)

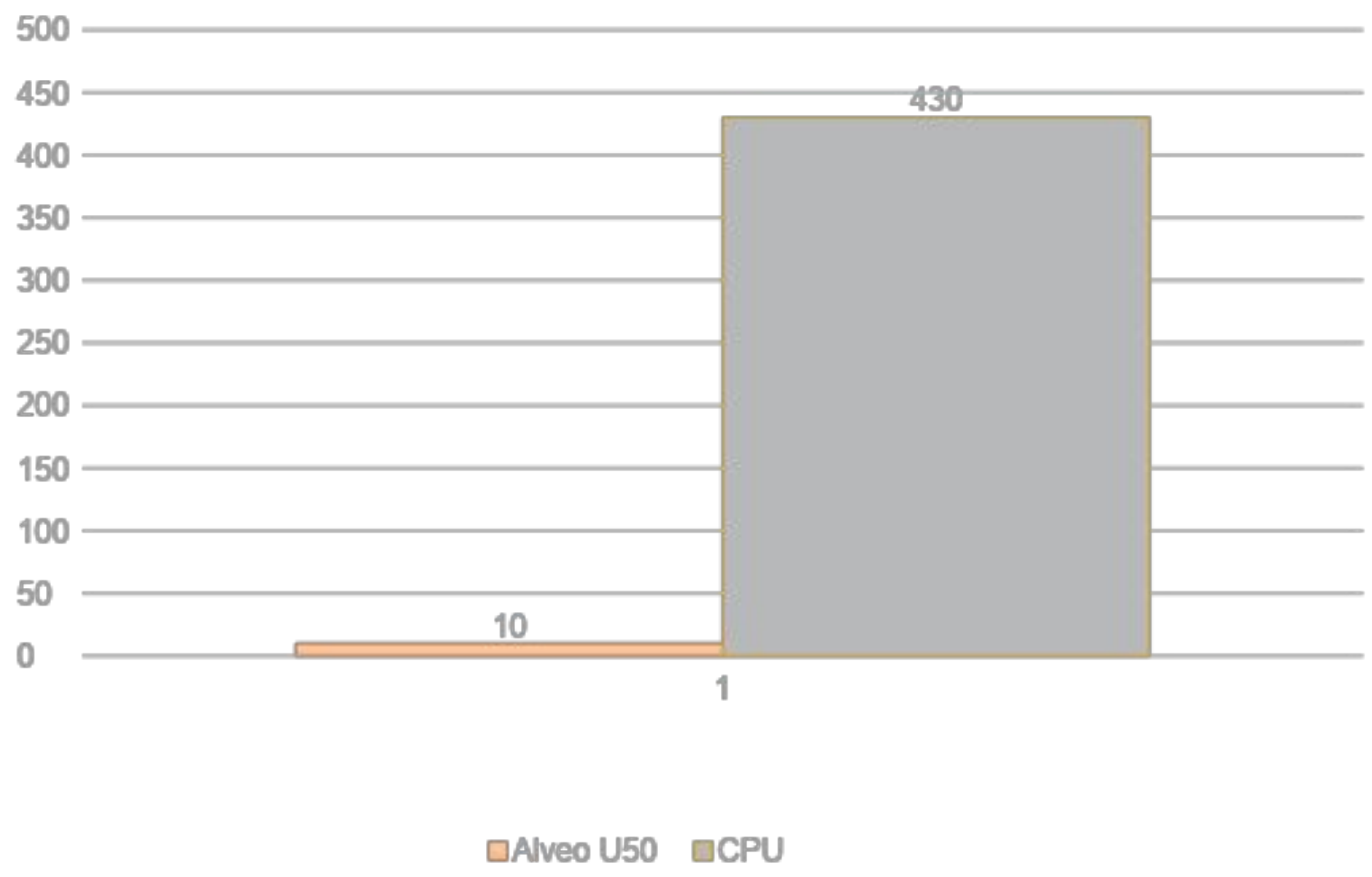
similarity.gsql:

open_alveo() { …; udf_open_alveo(1); …}
load_alveo() {…; udf_write_device(1); …}
close_alveo() { …; udf_close_alveo(1); …}
cosinesim_ss_alveo(newPatient, topK) { …; udf_cosinesim_ss_alveo(*newPatient, topK); …}*

OPTUM®

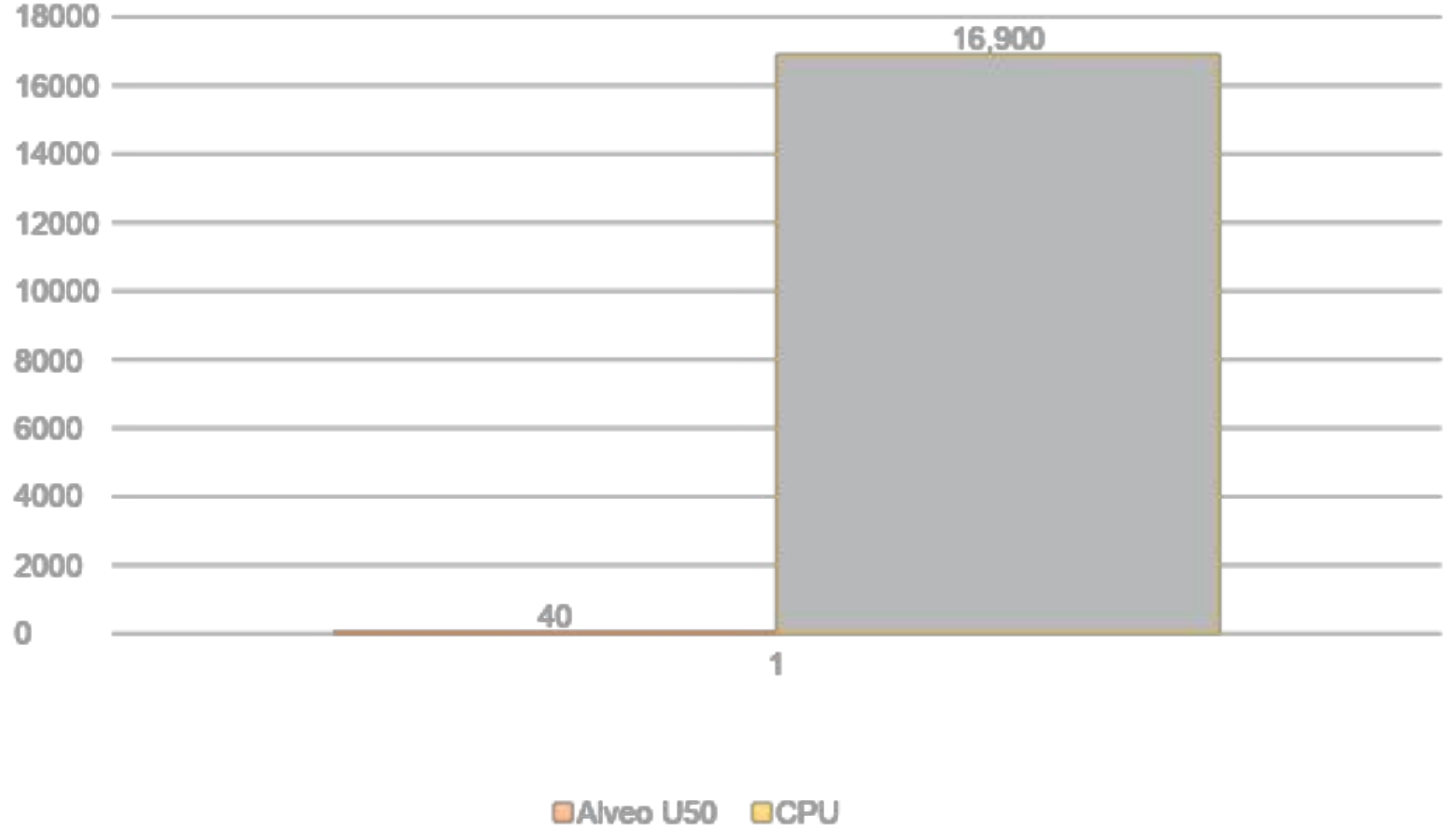
Time (milli-seconds) to get top 100 similar patients

40x faster than CPU



Using one Alveo U50

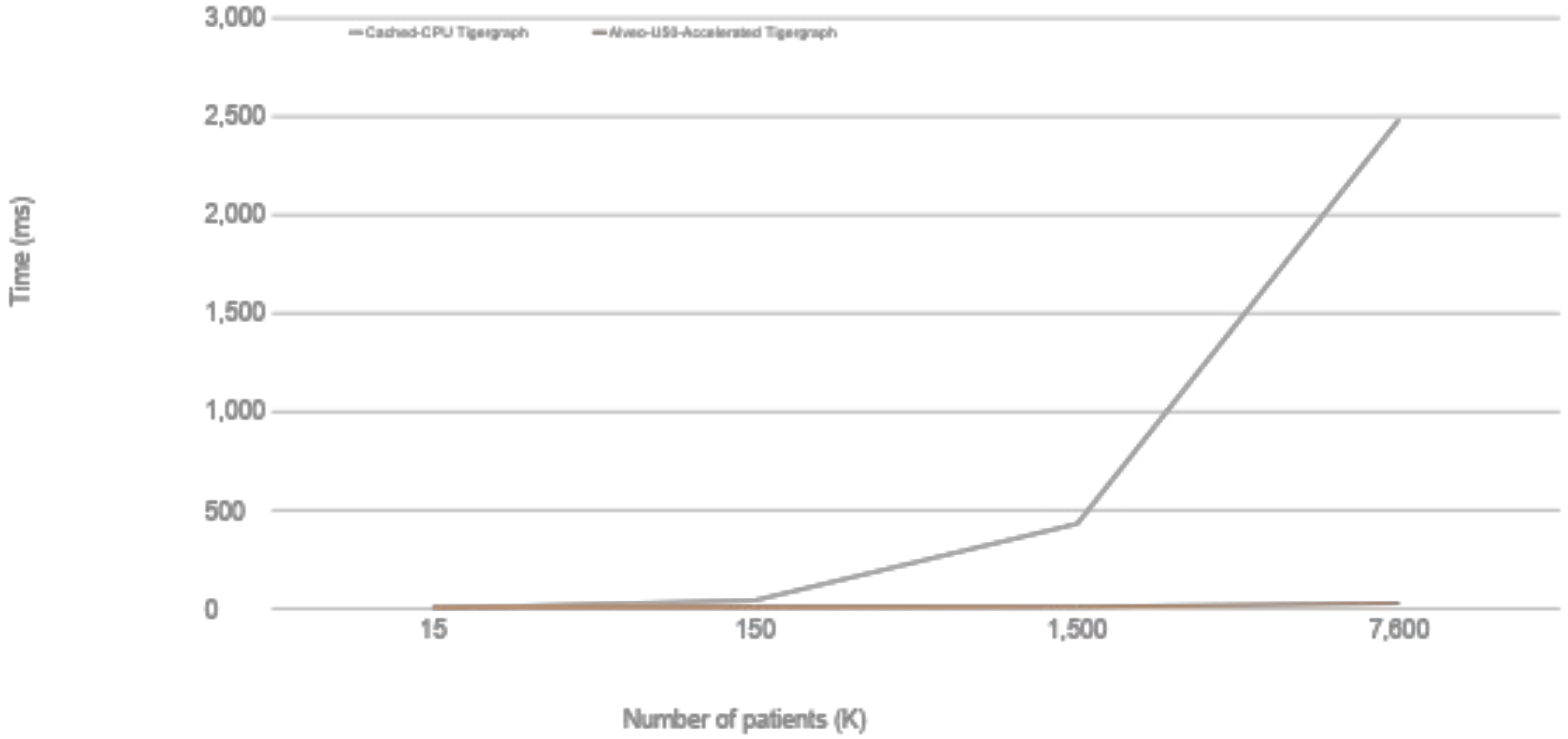
400x faster than CPU



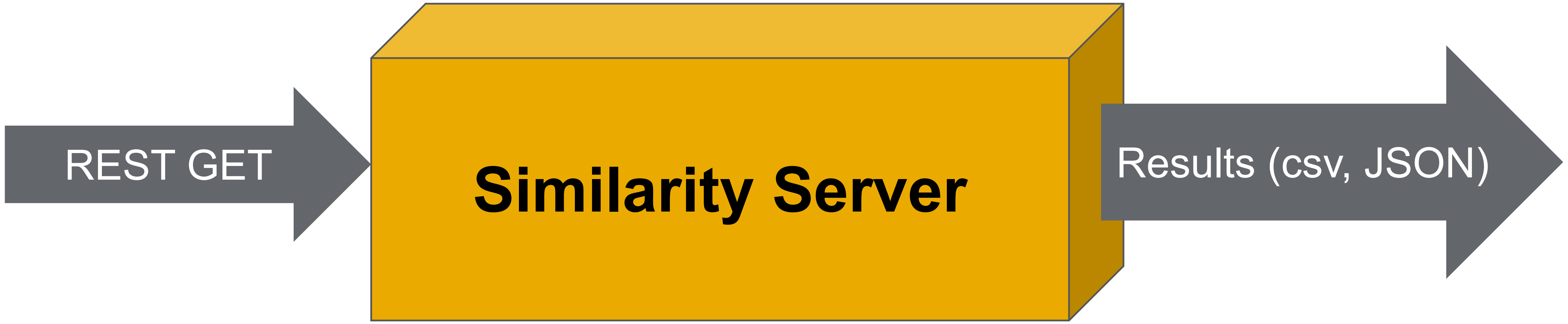
Using 5 Alveo U50’s

OPTUM®

Scaling with number of patients

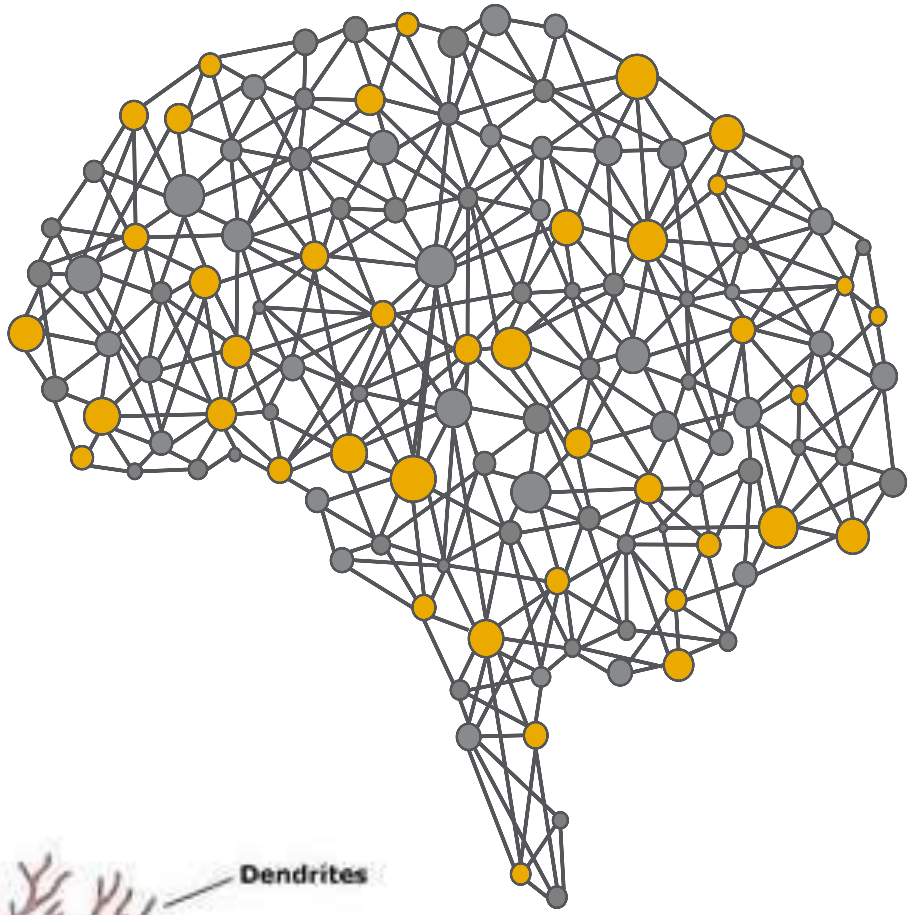


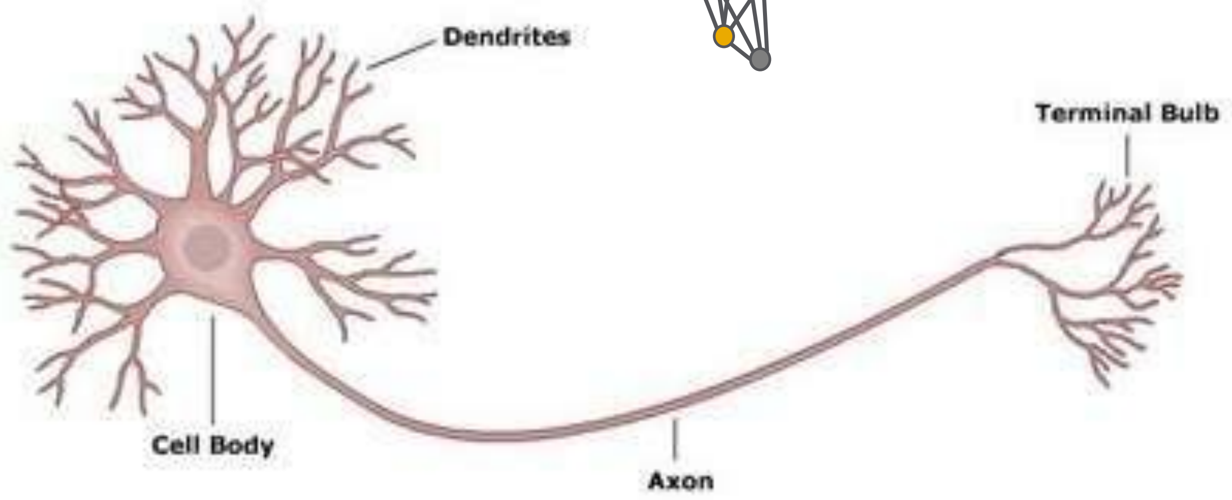
Three General REST Services to Support Similarity



- **Bulk Upload Data:** input - millions of vectors; output – success/failure code
- **Update Vector:** input - vertex ID, 198 integers; output – success/failure code
- **Find Similar:** input - vertex ID, 198 integers; output – 100 vertex IDs (64 bits)

Brain Metaphors and Analogies





- Use cautiously
- The brain has 82B neurons
- Each neuron has 10,000 connections (axons)
- Most axons are on “standby” for future learning
- Enterprise graphs have billions of vertices but typically have a **degree** of 5-6

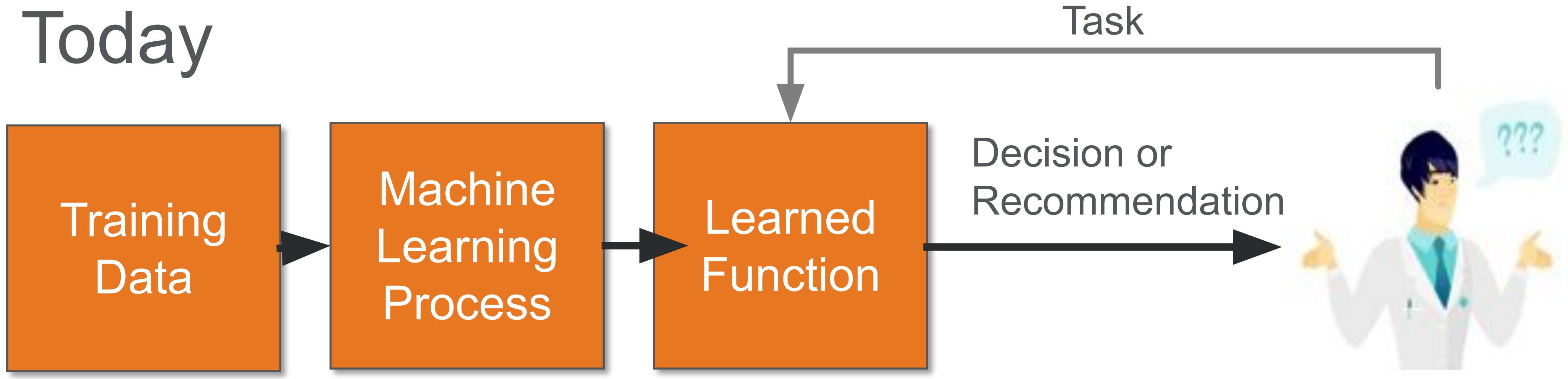
Analogies

- A plane is like a bird
- A submarine is like a fish
- A graph is like a brain

The Deep Learning Bartender

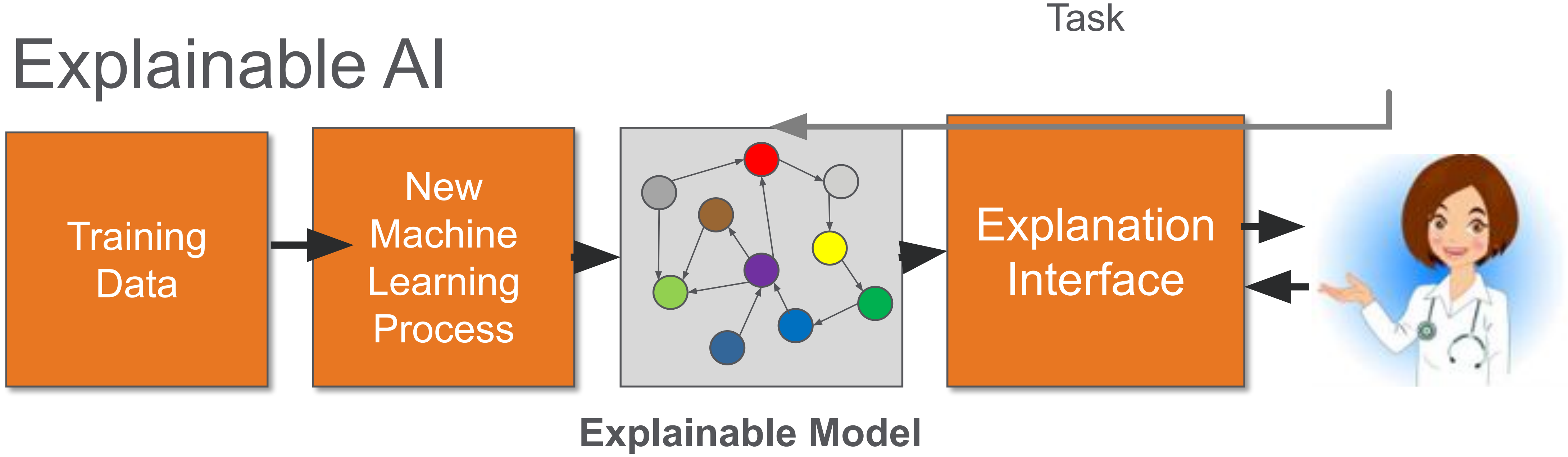
Explainable AI Models Leverage Graphs

Today



- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How to I correct an error?

Explainable AI



- I understand why
- I understand why not
- I know when you succeed
- I know when you fail
- I know when to trust you
- I know why you erred

OPTUM®

https://www.darpa.mil/program/explainable-artificial-intelligence

Three Eras of Computing

1) Procedural Era

Data

Procedural Code (Rules)

Programs

Answers

Explanations (Why)

2) Machine Learning Era

Data

Answers

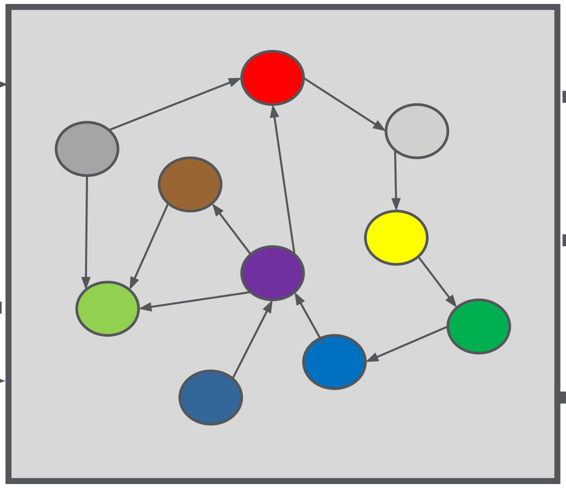
Machine Learning

Rules (10M weights)

3) Graph Era

Data

Machine Learning



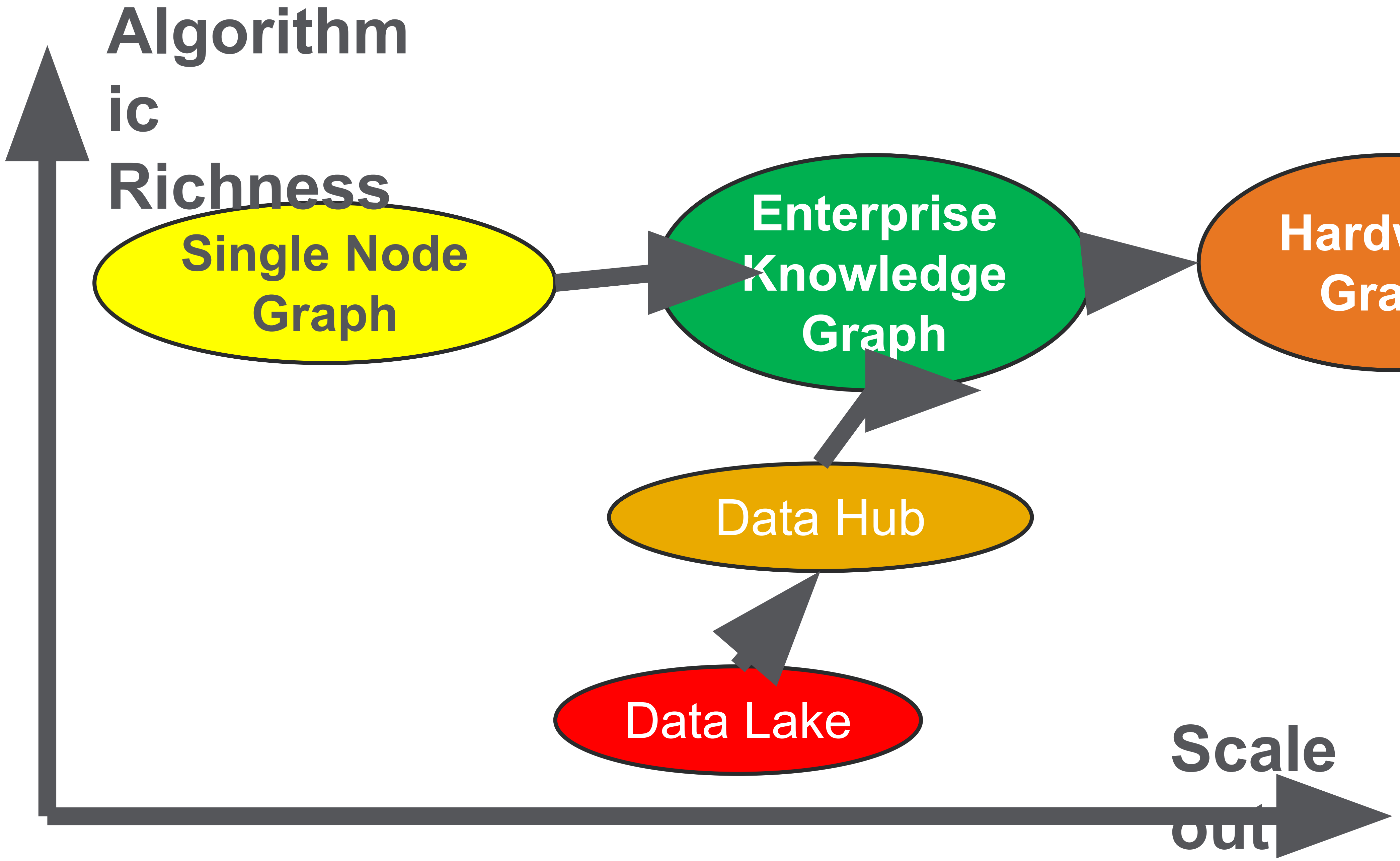
Knowledge

Answers

Explanations (Why)

OPTUM®

Onward to the Hardware Graph!



Related Use Cases

- **Recommendation Engines for Ecommerce and Healthcare**
  - For any person calling in for a recommended provider or senior living facility, can we find similar recommendations in the past?
- **Document Search**
  - Show similar documents to a search result using document embeddings
- **Incident Reporting**
  - When trouble ticket is reported, what are the most similar problems and what were their solutions?
- **Errors in Log Files**
  - When there are are error messages in log files, how can we find similar errors and their solutions?
- **Learning Content**
  - Can we recommend learning content for employees that have similar goals?
- **Schema Mapping**
  - Automate the process of creating data transformation maps for new data to existing schemas

Thank you!

Dan.McCreary@optum.com

Linked in http://www.linkedin.com/in/danmccreary

Medium http://dmccreary.medium.com

@dmccreary