

Building a recommendation engine with Neo4j and Clojure

Mark Needham @markhneedham

Introducing our data set...



meetup.com's recommendations



People in this

Meetup are also in:



Data & Analytics Innovation & Entrepreneurship

1,167 Members



Society of Data Miners

531 Analytics Practitioners



Cassandra London

1,635 Members



London New Tech

5,558 New technologists



Meteor London

1,596 Meteorites



London Ajax User Group

1,407 Software Engineers

ay: Join 190 R Users at "LondonR Meeting (and Workshop)"

Inbox x

LondonR <info@meetup.com> [Unsubscribe](#)

to me

15:40 (15 hours ago)

Events in this message

LondonR Meeting (and Workshop) Mon 30 Nov 2015 14:30 – 22:00 (WET)

[Add to Google Calendar](#)

MONDAY

LondonR Meeting (and Workshop)

LondonR

Monday, November 30, 2015
2:30 PM

Balls Brothers
Minster Court, Mincing Lane, EC3R 7PP
London

190 R Users going, including:



John Van Praag

"R in my main coding language. Aiming to look to gain deeper understanding of the language."

THURSDAY, NOVEMBER 26

6:30 PM

KNIME User Group UK

Anomaly Detection in Predictive Maintenance with KNIME

3 KNIMers going

6:30 PM

Spark London

11th Spark London Meetup - part of Big Data Week London

120 Members going

New Meetup Group: Agile without Borders

Inbox x



Meetup <info@meetup.com> [Unsubscribe](#)
to me

21 Nov (4 days ago)

Meetup

Agile without Borders

Community of Agile practitioners with in interest in developing techniques to scale across geography, time and cultures.

Join us

[Find out more](#)

Organized by



+1

Making recommendations



- ▶ Several different types
 - groups to join
 - topics to follow
 - events to attend
- ▶ As a user of [meetup.com](https://www.meetup.com) trying to find groups to join and events to attend

Making recommendations



- ▶ **Content based filtering**

Recommend items based on what users have liked in the past

- ▶ **Collaborative filtering**

Predict what users like based on the similarity of their behaviors, activities and preferences to others

Content based filtering



1. Collect item characteristics
2. Find similar items
3. Recommend similar items

e.g. similar movie genres

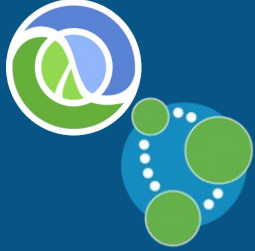
Collaborative filtering



1. Collect user behaviour
2. Find similar users
3. Recommend behaviour taken by similar users

e.g. people with similar musical tastes

Find similar groups to Neo4j



As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

What makes groups similar?



We're about:

Data Mining · New Technology · Web Development · Data Visualization · Data Analytics · Open Source · Cloud Computing · Graph Databases · Big Data · NoSQL · Neo4j · Database Development · Java · Computer programming



We're about:

Open Source · Technology · Web Development · Computer programming · Agile Project Management · Java · Software Development



We're about:

Big Data Analytics · Artificial Intelligence · Computer programming · Big Data · Computer Science · Natural Language Processing · Machine Learning · Data Analytics · Data Visualization · Data Mining · Data Science · Algorithms · Deep Learning · neural networks



We're about:

Data Science · Machine Learning · Predictive Analytics · Data Mining · Big Data · Artificial Intelligence · Statistical Computing · Applied Statistics · Data Analytics · Open Source · Web Analytics · Text Analytics · Natural Language Processing · Hadoop · NoSQL



We're about:

BigData · Intellectual Discussion · Big Data · Debate · New Technology · Data Analytics · Data Visualization · Online Marketing · Database Development · Information Architecture · Information Science · Freedom · Cloud Computing · Business Strategy · Internet Professionals

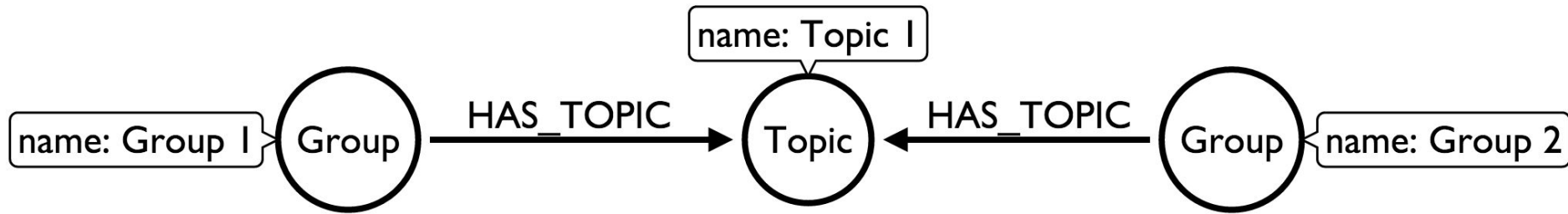


extract

We're about:

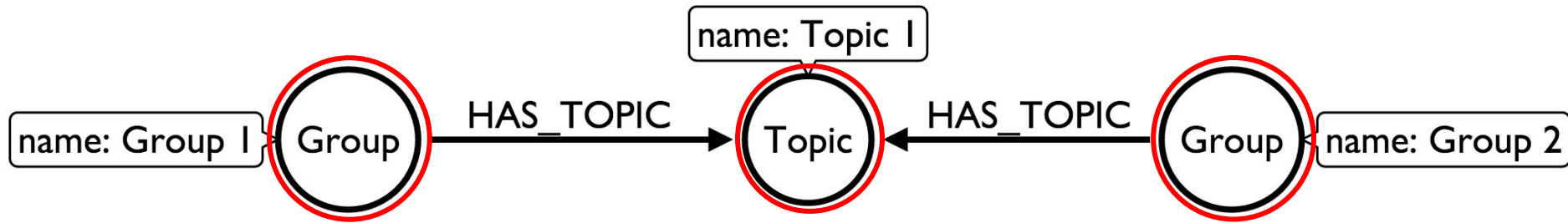
Marketing · Data · Software Development · New Technology · Web Technology · Business Intelligence · Cloud Computing · Business Strategy · Big Data · Machine Learning · Data Analytics · Data Visualization · Data Mining · Data Science · Big Data Analytics

Find similar groups to Neo4j



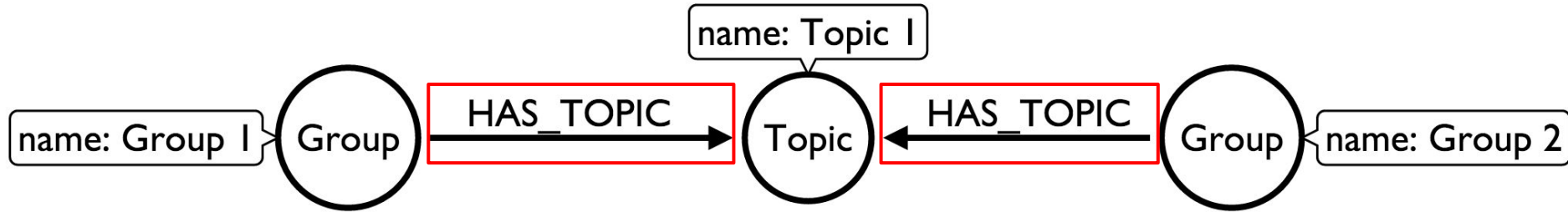
As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

Nodes



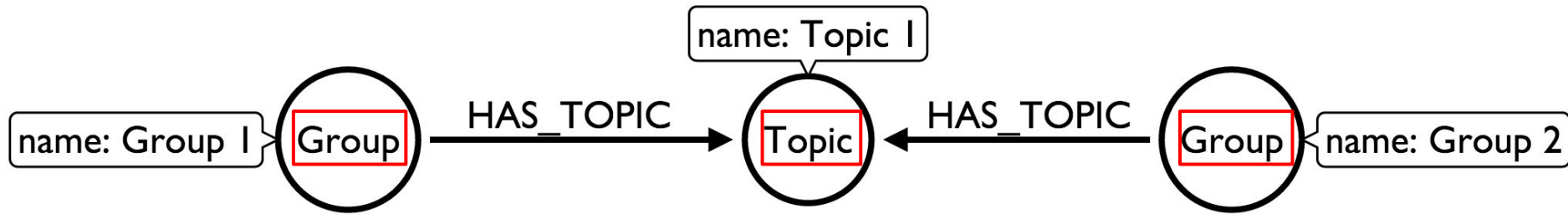
As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

Relationships



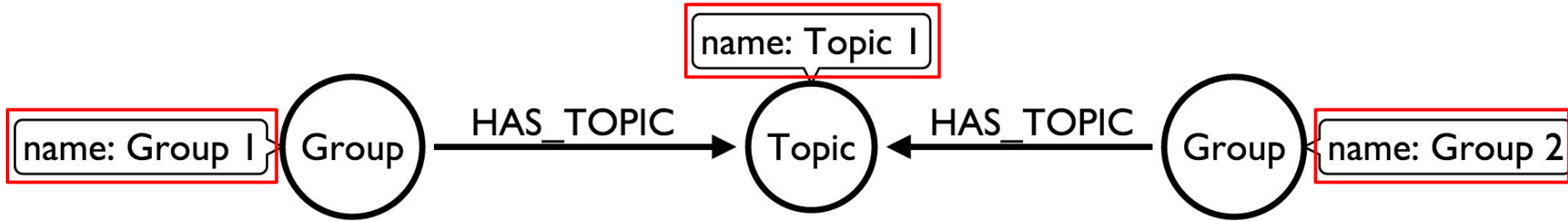
As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

Labels



As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

Properties

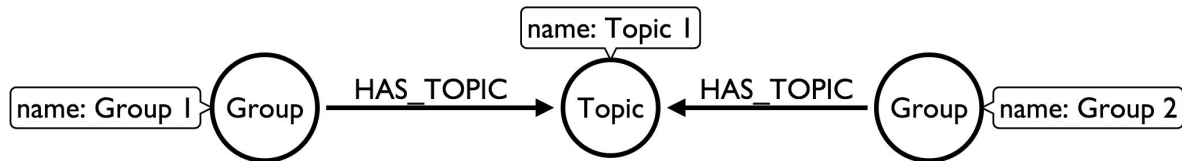


As a member of the Neo4j London group
I want to find other similar meetup groups
So that I can join those groups

Find similar groups to Neo4j



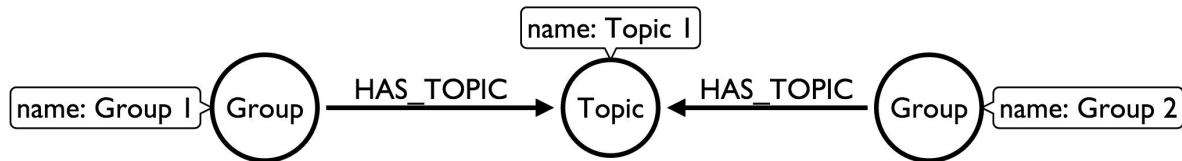
```
MATCH (group:Group)-[:HAS_TOPIC]->(t)<-[:HAS_TOPIC]-(otherGroup)
WHERE group.name = "Neo4j - London User Group"
RETURN otherGroup.name,
        COUNT(t) AS topicsInCommon,
        COLLECT(t.name) AS topics
ORDER BY topicsInCommon DESC, otherGroup.name
LIMIT 10
```





Match a pattern in the graph

```
MATCH (group:Group)-[:HAS_TOPIC]->(t)<-[:HAS_TOPIC]-(otherGroup)
WHERE group.name = "Neo4j - London User Group"
RETURN otherGroup.name,
       COUNT(t) AS topicsInCommon,
       COLLECT(t.name) AS topics
ORDER BY topicsInCommon DESC, otherGroup.name
LIMIT 10
```

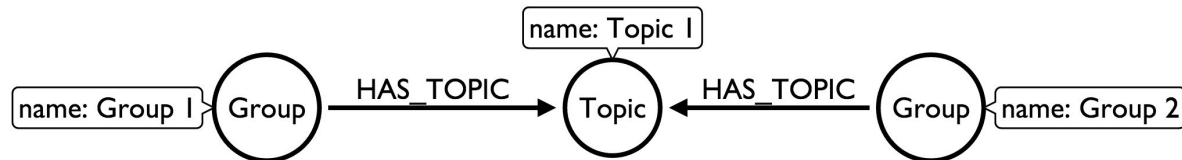




Anchor pattern in the graph

```
MATCH (group:Group)-[:HAS_TOPIC]->(t)<-[:HAS_TOPIC]-(otherGroup)
WHERE group.name = "Neo4j - London User Group"
RETURN otherGroup.name,
        COUNT(t) AS topicsInCommon,
        COLLECT(t.name) AS topics
ORDER BY topicsInCommon DESC, otherGroup.name
LIMIT 10
```

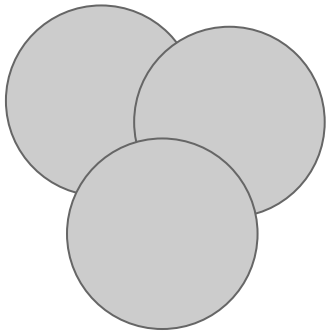
If an index for Group.name exists, Cypher will use it





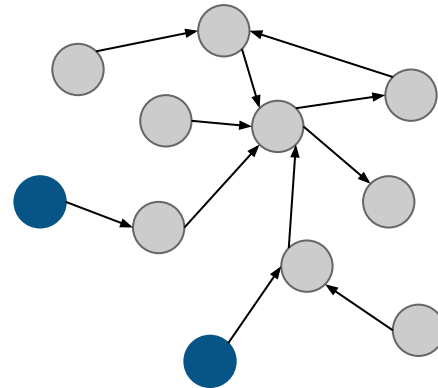
How does Neo4j use indexes?

Indexes are **only** used to find the starting points for queries.



Relational

Use index scans to look up rows in tables and join them with rows from other tables



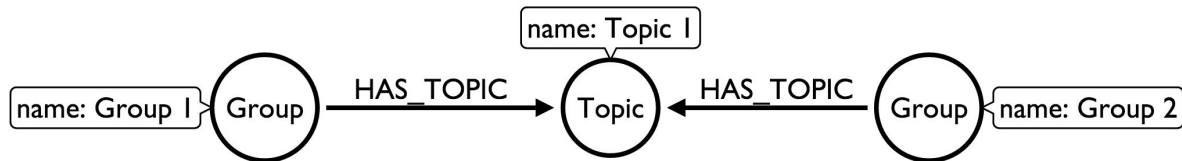
Graph

Use indexes to find the starting points for a query.

Inline the anchoring



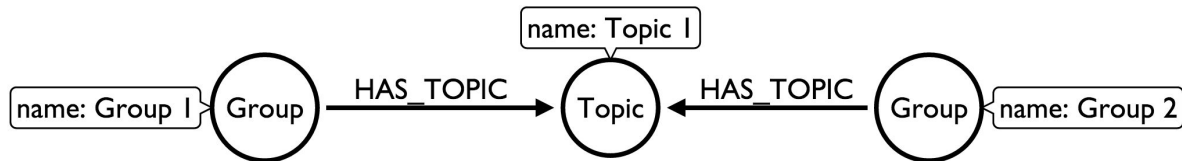
```
MATCH (group:Group {name: "Neo4j - London User Group"})
      -[:HAS_TOPIC]->(t)<-[:HAS_TOPIC]-(otherGroup)
RETURN otherGroup.name,
       COUNT(t) AS topicsInCommon,
       COLLECT(t.name) AS topics
ORDER BY topicsInCommon DESC, otherGroup.name
LIMIT 10
```



Create projection of results



```
MATCH (group:Group {name: "Neo4j - London User Group"})
      -[:HAS_TOPIC]->(t)<-[:HAS_TOPIC]-(otherGroup)
RETURN otherGroup.name,
       COUNT(t) AS topicsInCommon,
       COLLECT(t.name) AS topics
ORDER BY topicsInCommon DESC, otherGroup.name
LIMIT 10
```



Find similar groups to Neo4j



```
$ MATCH (group:Group {name: "Neo4j - London User Group"}) -[:HAS_TOPIC]->(topic)-[:HAS_TOPIC]-(otherGroup) RETURN otherGroup.name, COUNT(topic) AS to...
```



	otherGroup.name	topicsInCommon	topics
Rows	Python for Quant Finance	8	[New Technology, Cloud Computing, Big Data, Data Analytics, Data Visualization, Computer programming, Open Source, Data Mining]
Code	Closed scaling group	7	[Cloud Computing, Web Development, Big Data, Computer programming, Java, Open Source, NoSQL]
	Couchbase London	7	[Cloud Computing, Big Data, Database Development, Data Analytics, Open Source, NoSQL, Data Mining]
	London PostgreSQL Meetup Group	7	[New Technology, Cloud Computing, Web Development, Big Data, Database Development, Open Source, NoSQL]
	Business, Analytics and Data Science	7	[New Technology, Big Data, Data Analytics, Data Visualization, Computer programming, Open Source, Data Mining]
	Analytics.Club London	7	[Cloud Computing, Big Data, Data Analytics, Data Visualization, Open Source, NoSQL, Data Mining]
	Hadoop Users Group UK	7	[Cloud Computing, Big Data, Data Analytics, Java, Open Source, NoSQL, Data Mining]
	London NoSQL	6	[New Technology, Cloud Computing, Web Development, Big Data, Open Source, NoSQL]
	Data Science & Business Analytics London Meetup	6	[Big Data, Data Analytics, Data Visualization, Open Source, NoSQL, Data Mining]
	Big Data Week London Meetup	6	[Big Data, Data Analytics, Data Visualization, Open Source, NoSQL, Data Mining]
	Returned 10 rows in 22 ms.		

Neo4j <3 Clojure



Neocons

An idiomatic, feature rich Clojure client for Neo4J REST API

[Get Started! »](#)

[Read doc guides »](#)

[Join the mailing list »](#)

[Contribute »](#)

[Current Version - 2.0.1](#)

```
[clojurewerkz/neocons "3.1.0"]
```

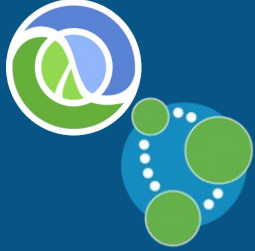
Connect to Neo4j



```
(:require
  [clojurewerkz.neocons.rest :as nr])

(def conn
  (nr/connect
    "http://neo4j:password@localhost:7474/db/data/"))
```

Find meetups by topic



```
(:require [clojurewerkz.neocons.rest.cypher :as cy]
          [clojure.walk :as walk]))

(def query
  "MATCH (group:Group)-[:HAS_TOPIC]->(:Topic {name: {topic}})
   RETURN group
   LIMIT 5")

(->> (cy/tquery conn query {:topic "<Insert topic here>"})
      walk/keywordize-keys
      (map #(-> % :group :data)))
```


Find Clojure meetups



```
(->> (cy/tquery conn query {:topic "Clojure"})  
      walk/keywordize-keys  
      (map #(-> % :group :data)))
```

```
{:created 1384181724000, :rating 4, :name "London Functional  
Programmers", :id "11057352", :urlname "London-Functionals"}
```

```
{:created 1231235336000, :rating 4, :name "OpenSource & Agile  
Community Events", :id "1350857", :urlname "skillsmatter"}
```

Find Neo4j meetups



```
(->> (cy/tquery conn query {:topic "Neo4j"})  
      walk/keywordize-keys  
      (map #(-> % :group :data)))
```

```
{:created 1306977843000, :rating 4, :name "Neo4j - London User  
Group", :id "1954021", :urlname "graphdb-london"}
```

```
{:created 1440455142000, :rating 0, :name "London Data  
Visualization", :id "18864999", :urlname "London-Data-  
Visualization"}
```

Building a Clojure web app



Luminus is a Clojure micro-framework based on a set of lightweight libraries. It aims to provide a robust, scalable, and easy to use platform. With Luminus you can focus on developing your app the way you want without any distractions.

Clojure Web Development Made Simple

Quick Start Guide

```
$ lein new luminus my-app
$ cd my-app
$ lein run
Started server on port 3000
```

Get started in seconds using
Leiningen.

Wiring everything up



```
(defroutes home-routes  
  (GET "/" [] (home-page)))
```

Define route

Wiring everything up



```
(defroutes home-routes
  (GET "/" [] (home-page)))

(defn home-page []
  (layout/render
    "home.html" { :suggested-groups (suggested-groups logged-in-user)
                  :suggested-events (suggested-events queries/suggested-events)}))
```

Create HTML page and outline
which data it's going to have

Wiring everything up



```
(defroutes home-routes
  (GET "/" [] (home-page)))

(defn home-page []
  (layout/render
    "home.html" { :suggested-groups (suggested-groups logged-in-user)
                  :suggested-events (suggested-events logged-in-user /suggested-events)}))

(defn suggested-groups [name]
  (let [result (cypher/execute queries/suggested-groups {:name name})]
    (->> result
      (map #(assoc % :score (scoring/score-row %)))
      (sort-by :score >)
      (take 12))))
```

Execute Cypher query against Neo4j
and do post processing of the result

Wiring everything up



```
(defroutes home-routes
  (GET "/" [] (home-page)))

(defn home-page []
  (layout/render
    "home.html" { :suggested-groups (suggested-groups logged-in-user)
                  :suggested-events (suggested-events queries/suggested-events)}))

(defn suggested-groups [name]
  (let [result (cypher/execute queries/suggested-groups {:name name})]
    (->> result
      (map #(assoc % :score (scoring/score-row %)))
      (sort-by :score >)
      (take 12))))
```

Suggested groups



Suggested Groups

Analytics Network

8 topics / 994 members / 6 recent events

286.65657821991084

PyData London Meetup

7 topics / 2651 members / 3 recent events

286.1046685975722

Women Who Code London

4 topics / 2287 members / 11 recent events

283.88029437568923

Python for Quant Finance

10 topics / 1166 members / 3 recent events

282.558303354221

PHP London

5 topics / 4415 members / 3 recent events

280.3796114161744

Learn Software Development

5 topics / 977 members / 8 recent events

279.6404781692217

DataKind UK

10 topics / 1746 members / 2 recent events

276.03450587463783

FREE Marketing, Analytics & Digital Skills in London

3 topics / 5491 members / 25 recent events

275.17068435438404

West London Hack Night

5 topics / 943 members / 4 recent events

274.4717650232895

London Ajax User Group

4 topics / 1433 members / 3 recent events

270.9970989549757

London Internet Startups

3 topics / 1359 members / 17 recent events

270.90907447488087

Meteor London

5 topics / 1673 members / 2 recent events

266.8676279564873

Anatomy of a recommendation



```
MATCH (member:Member {name: {name}})-[:INTERESTED_IN]->()-[:HAS_TOPIC]-(otherGroup)
WHERE NOT (member)-[:MEMBER_OF]->(otherGroup)

WITH otherGroup,
     COUNT(*) AS topics,
     SIZE((otherGroup)-[:MEMBER_OF]-()) AS numberOfMembers

OPTIONAL MATCH (otherGroup)-[:HOSTED_EVENT]->(event)
WHERE (timestamp() - 90*24*60*60*1000 ) < event.time < timestamp()

RETURN otherGroup, topics , numberOfMembers, COUNT(event) AS recentEvents
```

Anatomy of a recommendation



```
MATCH (member:Member {name: {name}})-[:INTERESTED_IN]->()-[:HAS_TOPIC]-(otherGroup)
WHERE NOT (member)-[:MEMBER_OF]->(otherGroup)
```

Find groups that
share our interests

```
WITH otherGroup,
     COUNT(*) AS topics,
     SIZE((otherGroup)-[:MEMBER_OF]-()) AS numberOfMembers
```

```
OPTIONAL MATCH (otherGroup)-[:HOSTED_EVENT]->(event)
WHERE (timestamp() - 90*24*60*60*1000 ) < event.time < timestamp()
```

```
RETURN otherGroup, topics , numberOfMembers, COUNT(event) AS recentEvents
```

Anatomy of a recommendation



```
MATCH (member:Member {name: {name}})-[:INTERESTED_IN]->()-[:HAS_TOPIC]-(otherGroup)
WHERE NOT (member)-[:MEMBER_OF]->(otherGroup)
```

Filter out the ones
we're already in

```
WITH otherGroup,
     COUNT(*) AS topics,
     SIZE((otherGroup)-[:MEMBER_OF]-()) AS numberOfMembers
```

```
OPTIONAL MATCH (otherGroup)-[:HOSTED_EVENT]->(event)
WHERE (timestamp() - 90*24*60*60*1000 ) < event.time < timestamp()
```

```
RETURN otherGroup, topics , numberOfMembers, COUNT(event) AS recentEvents
```

Anatomy of a recommendation



```
MATCH (member:Member {name: {name}})-[:INTERESTED_IN]->()-[:HAS_TOPIC]-(otherGroup)
WHERE NOT (member)-[:MEMBER_OF]->(otherGroup)
```

```
WITH otherGroup,
     COUNT(*) AS topics,
     SIZE((otherGroup)-[:MEMBER_OF]-()) AS numberOfMembers
```

Collect some metrics to
evaluate group relevance
and popularity

```
OPTIONAL MATCH (otherGroup)-[:HOSTED_EVENT]->(event)
WHERE (timestamp() - 90*24*60*60*1000 ) < event.time < timestamp()
```

```
RETURN otherGroup, topics , numberOfMembers, COUNT(event) AS recentEvents
```

Anatomy of a recommendation



```
MATCH (member:Member {name: {name}})-[:INTERESTED_IN]->()-[:HAS_TOPIC]-(otherGroup)
WHERE NOT (member)-[:MEMBER_OF]->(otherGroup)
```

```
WITH otherGroup,
     COUNT(*) AS topics,
     SIZE((otherGroup)-[:MEMBER_OF]-()) AS numberOfMembers
```

Check if the
group is active

```
OPTIONAL MATCH (otherGroup)-[:HOSTED_EVENT]->(event)
WHERE (timestamp() - 90*24*60*60*1000 ) < event.time < timestamp()
```

```
RETURN otherGroup, topics , numberOfMembers, COUNT(event) AS recentEvents
```

Scoring the recommendation



```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
      0
      (let [alpha (/ (log2 5) eighty)
            exp (Math/exp (* (- alpha) row))]
          (* maximum (- 1 exp)))))
```

```
(defn score-item [{:minimum :minimum maximum :maximum eighty :eighty n :n}]
  (score minimum maximum eighty n))
```

```
(defn score-row [row]
  (let [topics  {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events  {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```

Scoring the recommendation



```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
      0
      (let [alpha (/ (log2 5) eighty)
            exp (Math/exp (* (- alpha) row))]
          (* maximum (- 1 exp)))))
```

```
(defn score-item [{:minimum :minimum :maximum :maximum :eighty :eighty :n :n}]
  (score minimum maximum
```

Take in raw scores for
number of topics, members
and recent events

```
(defn score-row [row]
  (let [topics  {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events  {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```

Scoring the recommendation



```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
    0
    (let [alpha (/ (log2 5) eighty)
          exp (Math/exp (* (- alpha) row))]
      (* maximum (- 1 exp)))))
```

```
(defn score-item [{:minimum
                   :score
                   :minimum
                   :maximum
```

Take in raw scores for
number of topics, members
and recent events

```
{ :topics 3
  :numberOfMembers 932
  :recentEvents 3 }
```

```
(defn score-row [row]
  (let [topics {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```


Scoring the recommendation



```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
      0
      (let [alpha (/ (log2 5) eighty)
            exp (Math/exp (* (- alpha) row))]
          (* maximum (- 1 exp)))))
```

```
(defn score-item [{:minimum :minimum :maximum :maximum :eighty :eighty :n :n}]
  (score minimum maximum eighty n))
```

Apply a weighting
to each of those
properties

```
(defn score-row [row]
  (let [topics  {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events  {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```

Scoring the recommendation



```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
      0
      (let [alpha (/ (log2 5) eighty)
            exp (Math/exp (* (- alpha) row))]
          (* maximum (- 1 exp)))))
```

Apply an exponential (Pareto) function to the curve so that high scores don't dominate too much.

```
(defn score-item [{:minimum :minimum maximum :maximum eighty :eighty n :n}]
  (score minimum maximum eighty n))
```

```
(defn score-row [row]
  (let [topics  {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events  {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```

Scoring the recommendation

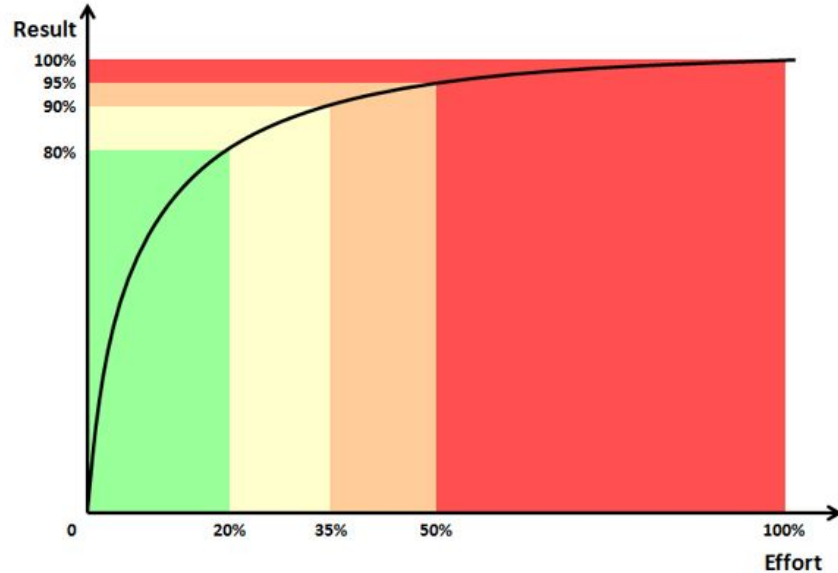


```
(defn log2 [n] (/ (Math/log n) (Math/log 2)))
```

```
(defn score [minimum maximum eighty row]
  (if (< row minimum)
    0
    (let [alpha (/ (log2 5) eighty)
          exp (Math/exp (* (- alpha) row))]
      (* maximum (- 1 exp)))))
```

```
(defn score-item [{:minimum :minimum :maximum :
                  :score :score :minimum :maximum :eighty :eighty :n}]
  (score :minimum :maximum :eighty :n))
```

```
(defn score-row [row]
  (let [topics  {:n (-> row :topics) :minimum 1 :maximum 100 :eighty 5}
        members {:n (-> row :numberOfMembers) :minimum 50 :maximum 100 :eighty 1000}
        events  {:n (-> row :recentEvents) :minimum 1 :maximum 100 :eighty 3}]
    (reduce #(+ %1 (score-item %2)) 0 [topics members events])))
```



What about events?



MONDAY, NOVEMBER 30

2:30 PM

LondonR

✓ Yes

LondonR Meeting (and Workshop)

209 R Users going

9:00 AM

OpenSource & Agile Community Events

CSM Course: Advanced Certified ScrumMaster Course by Martine Devos

5 OpenSource and Agile developers going

10:00 AM

OpenSource & Agile Community Events

iOSCon 2015 - The Conference for iOS and Swift Developers

9 OpenSource and Agile developers going

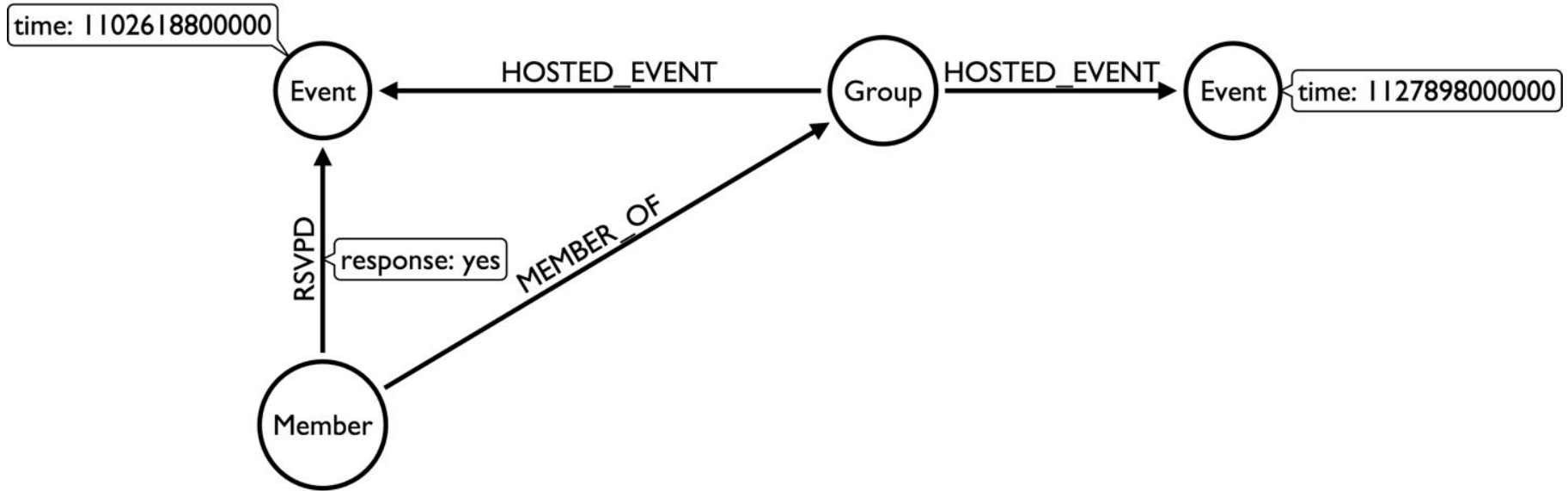
7:00 PM

Cleanweb London

COP21 Cleanweb and the Climate

62 Members going

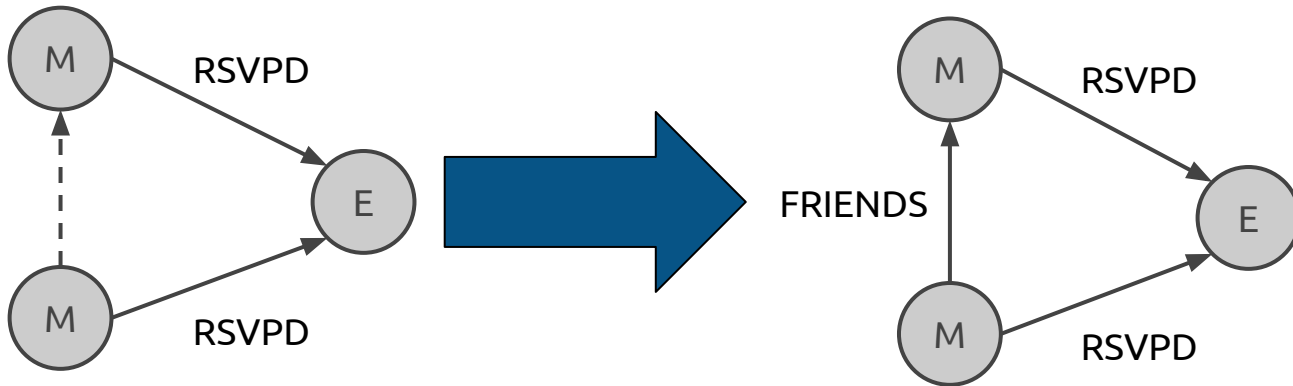
Modeling events



Who are my meetup friends?



There's an **implicit** FRIENDS relationship between people who attended the same events. Let's make it **explicit**.



Who are my meetup friends?



```
MATCH (m:Member)
```

```
WITH m1 LIMIT {limit}
```

```
MATCH (m)-[:RSVPD {response: 'yes'}]->(e:Event)<-[:RSVPD {response: 'yes'}]-(m2:Member)
```

```
WITH m, m2, COLLECT(e) AS events, COUNT(*) AS times
```

```
WHERE times >= 5
```

```
WITH m, m2, times,
```

```
    [event IN events | SIZE((event)<-[:RSVPD {response: 'yes'}]-( ))] AS attendances
```

```
WITH m, m2, REDUCE(score = 0.0, a IN attendances | score + (1.0 / a)) AS score
```

```
MERGE (m)-[friendsRel:FRIENDS]-(m2)
```

```
SET friendsRel.score = row.score
```

Who are my meetup friends?



```
MATCH (m:Member)
```

```
WITH m1 LIMIT {limit}
```

```
MATCH (m)-[:RSVPD {response: 'yes'}]->(e:Event)<-[:RSVPD {response: 'yes'}]-(m2:Member)
```

```
WITH m, m2, COLLECT(e) AS events, COUNT(*) AS times
```

```
WHERE times >= 5
```

```
WITH m, m2, times, [event IN events | (m)-[:RSVPD {response: 'yes'}]-(event)] AS attendances
```

I can only be friends with someone
if I've attended 5 or more of the
same meetups as them

```
WITH m, m2, REDUCE(score = 0.0, a IN attendances | score + (1.0 / a)) AS score
```

```
MERGE (m)-[friendsRel:FRIENDS]-(m2)
```

```
SET friendsRel.score = row.score
```


Who are my meetup friends?



```
MATCH (m:Member)
```

```
WITH m1 LIMIT {limit}
```

```
MATCH (m)-[:RSVPD {response: 'yes'}]->(e:Event)<-[:RSVPD {response: 'yes'}]-(m2:Member)
```

```
WITH m, m2, COLLECT(e) AS events, COUNT(*) AS times
```

```
WHERE times >= 5
```

```
WITH m, m2, times,
```

```
    [event IN events | SIZE((event)<-[:RSVPD {response: 'yes'}]-( ))] AS attendances
```

```
WITH m, m2, REDUCE(score = 0.0, a IN attendances | score + (1.0 / a)) AS score
```

```
MERGE (m)-[friendsRel:FRIENDS]-(m2)
```

```
SET friendsRel.score = row.score
```

We create a score for the friendship based on the likelihood that we met them at an event - the more people that attended the less likely we met

Who are my meetup friends?



PHP London

(on [meetup.com](#))

People 4415

Similar groups

London Tech Meetup
Meteor London

BS Congress
Polish Developers in London

Drupal Dinners London
Geek Girls Carrots UK

Friends

31
M.G.Srikanthan
Hammad Soomro
Peter Morgan
Jean-Francois
Marcin Tolysz
Ravi
Reshmee
Martin Belan
JUNIOR SAYLES
Andrei Ruse
Nitin Patel

Peter Arato
Raul Kumar
Cezar Grigore
Jelena Kosakovskaja
Eftar Miah
James Rowlands
Chris Jeffery
Eddie H
dharshi thiru
Erika Pellegrino
gary aghedo

Ajit
Fox Reymann
Vladyslav Piskunov
Ben Auffarth
James Churchman
Marcos Santos
Mircea Danila Dumitrescu
Amit
Enzo Martoglio

When Clojure, When Cypher?



- ▶ The combination work very well for **data oriented programming**



When Clojure, When Cypher?

- ▶ The combination work very well for **data oriented programming**
- ▶ Cypher works best for...
 - expressing graph patterns
 - exploring relationships in data

When Clojure, When Cypher?



- ▶ The combination work very well for **data oriented programming**
- ▶ Cypher works best for...
 - expressing graph patterns
 - exploring relationships in data
- ▶ Clojure is more suitable for...
 - chaining functions to be applied to data
 - Manipulating and massaging data

A different way of recommending



- ▶ Popular approaches use global number crunching e.g. item based similarity

A different way of recommending



- ▶ Popular approaches use global number crunching e.g. item based similarity
- ▶ Our approach is **more personalised and makes use of local searches** around the user's neighbourhood

What could we do next?

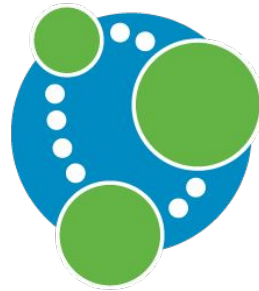


- ▶ Comments sentiment analysis
 - do people actually like the events they go to?
- ▶ Topic ontology
 - how are topics related? e.g. Neo4j, Cassandra, MongoDB are part of NoSQL
- ▶ Event similarity based on descriptions
 - use Latent Dirichlet Allocation to derive categories

What could we do next?



- ▶ Social network
 - what events do our twitter/Facebook friends attend?
- ▶ Location
 - do we favour events in a certain part of town?
- ▶ Day of the week
 - do we only go to events on certain days of the week?
 - do we go to different events on weekdays vs weekend?



That's all for today!
Questions? :-)

Mark Needham @markhneedham

<https://github.com/neo4j-meetups/modeling-worked-example>

<https://github.com/mneedham/clojure-recommendations>