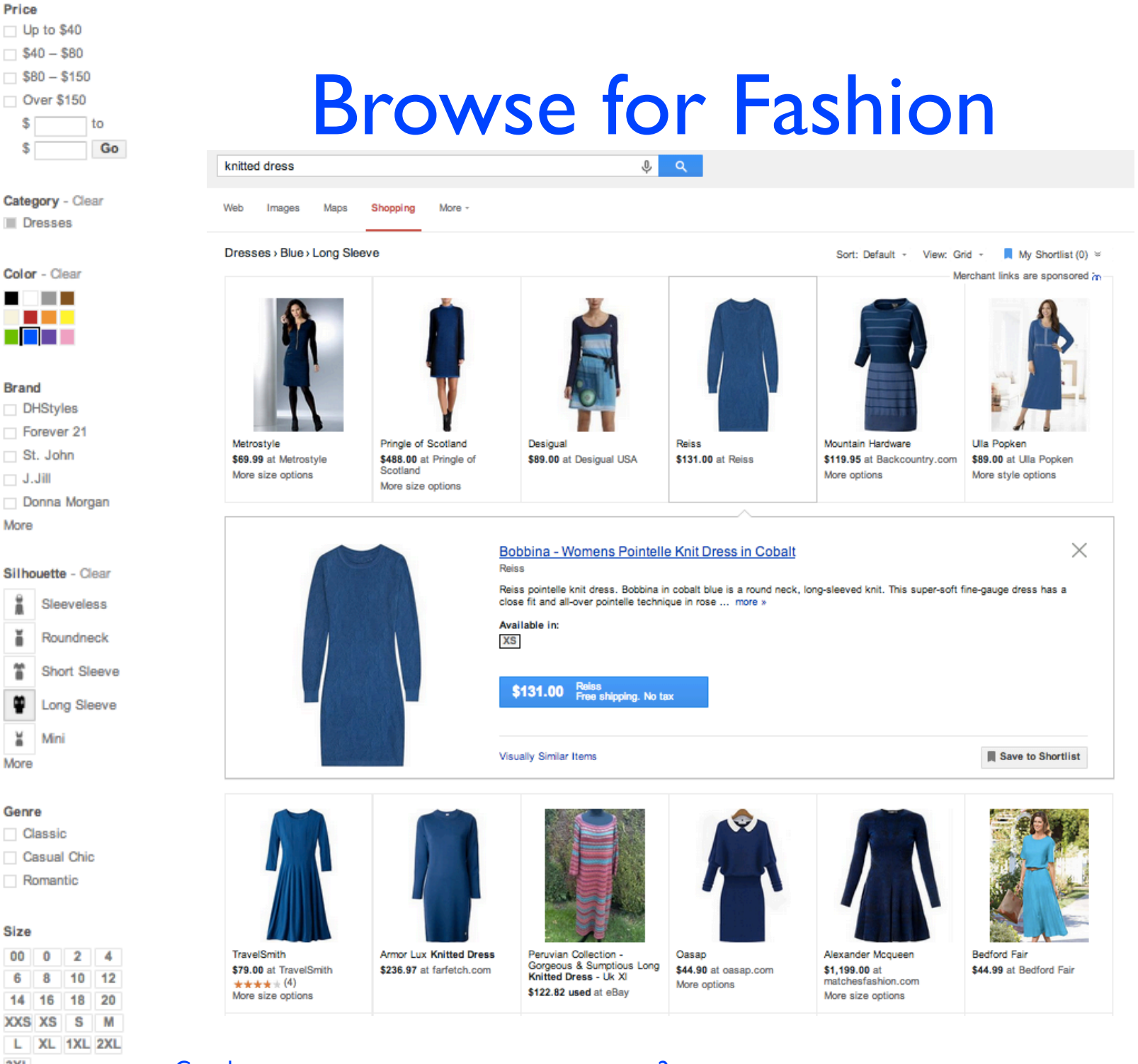
Searching for Similar Items in Diverse Universes

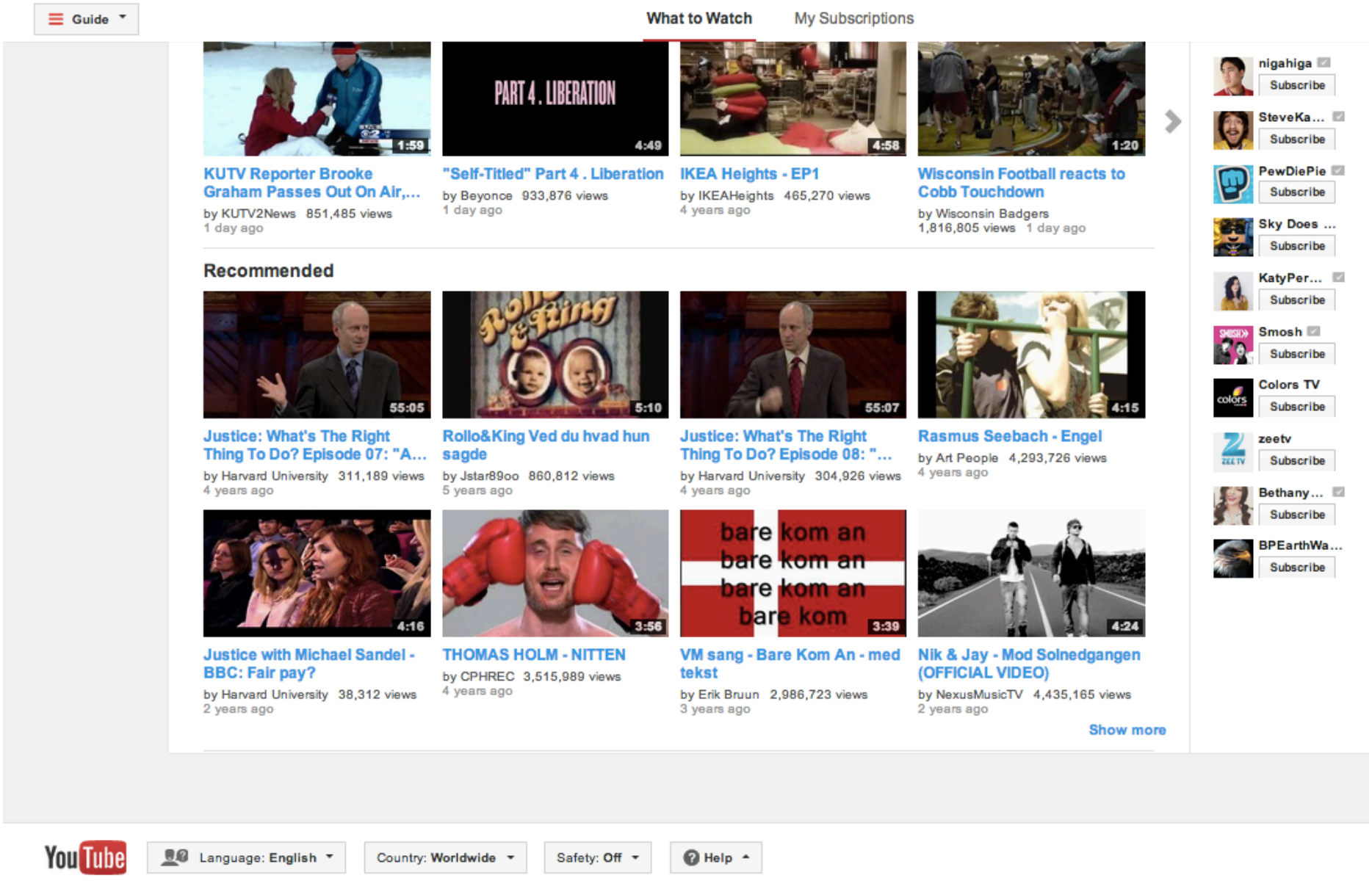
Large-scale machine learning at Google

Corinna Cortes
Google Research
corinna@google.com

Browse for Fashion



Browse for Videos



Outline

- Metric setting
  - Using similarities (efficiency)
  - Learning similarities (quality)
- Graph-based setting
  - Generating similarities

Image Browsing

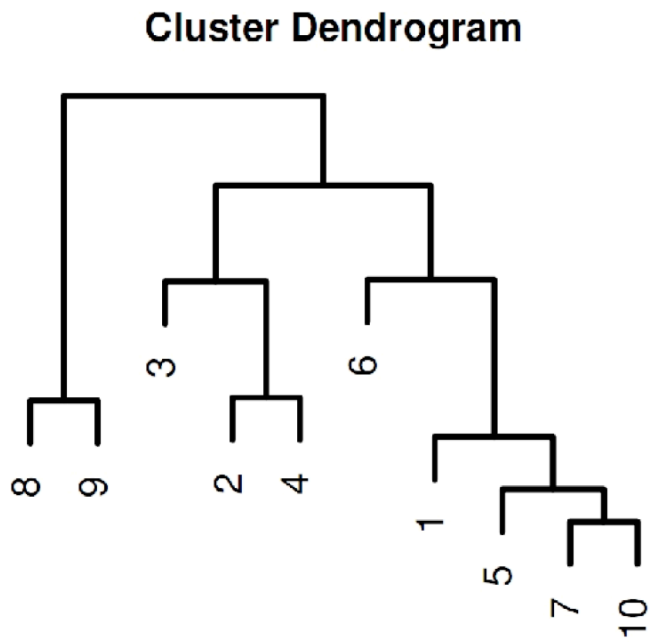
- Image browsing relies on some measure of similarity.

$$\begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1N} \end{pmatrix} \qquad \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2N} \end{pmatrix}$$

$$\mathrm{Sim}(\mathbf{x_1}, \mathbf{x_2}) = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1N} \end{pmatrix} \cdot \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2N} \end{pmatrix}$$

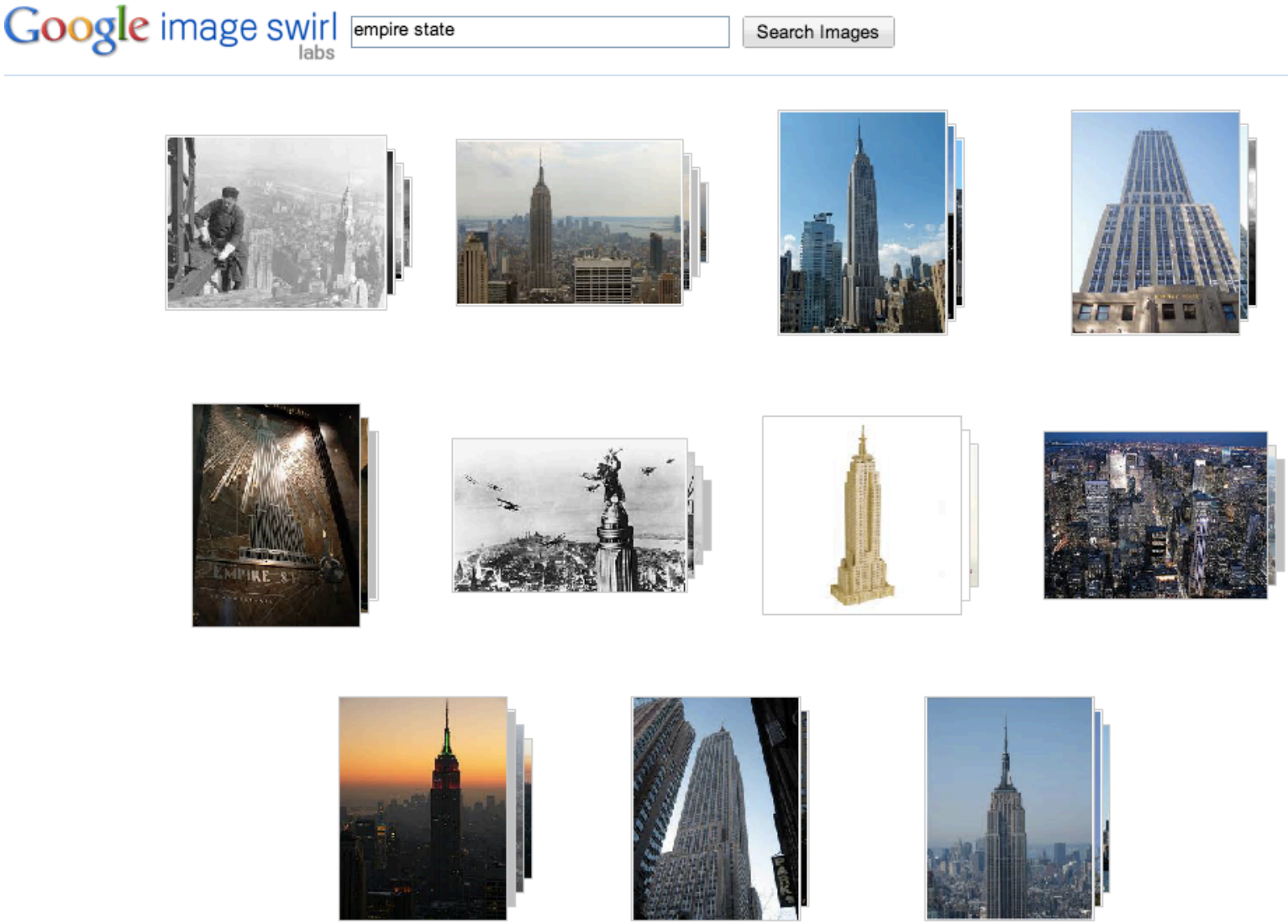
Clustering of Images

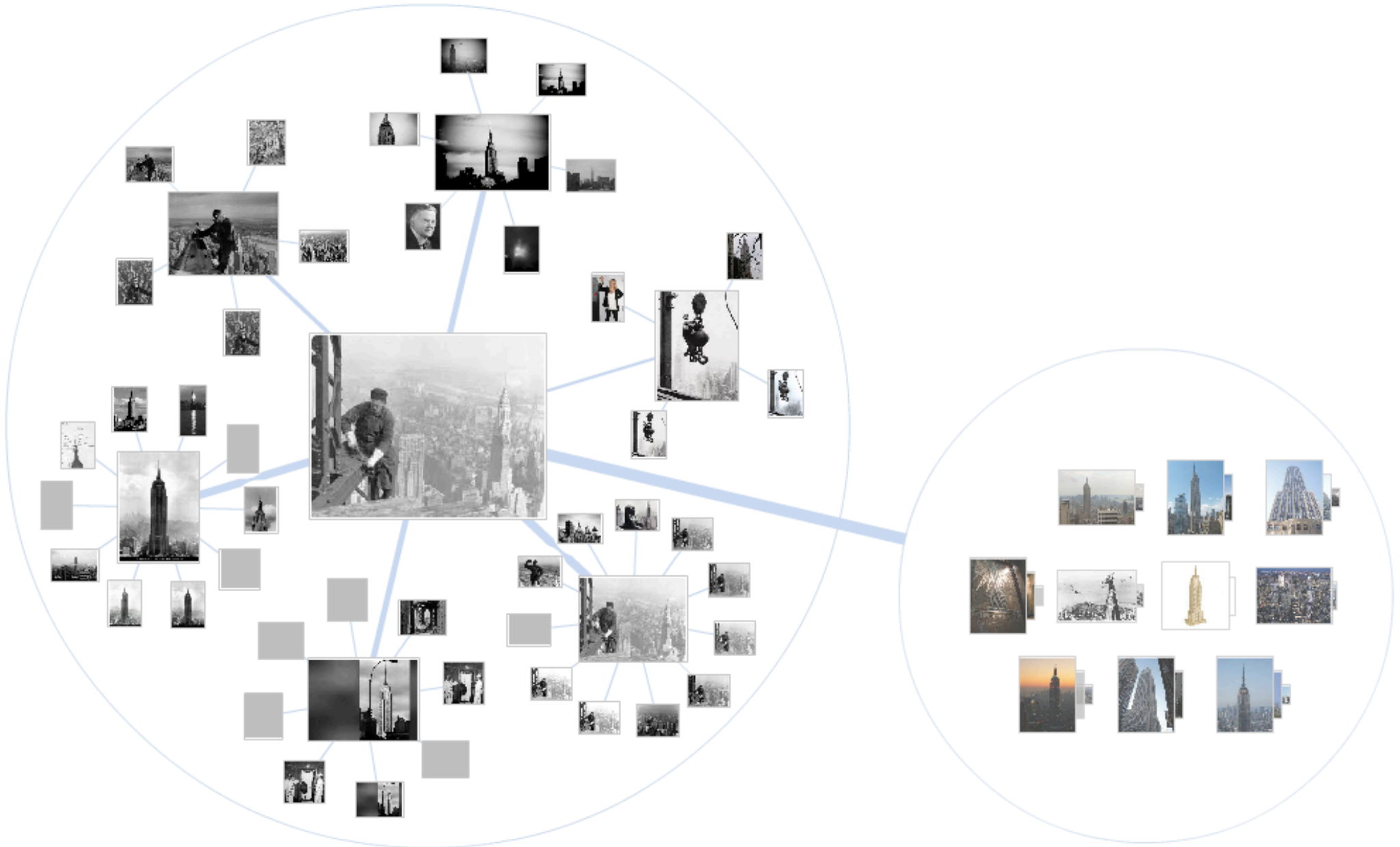
- Compute all the pair-wise distances between related images and form clusters:



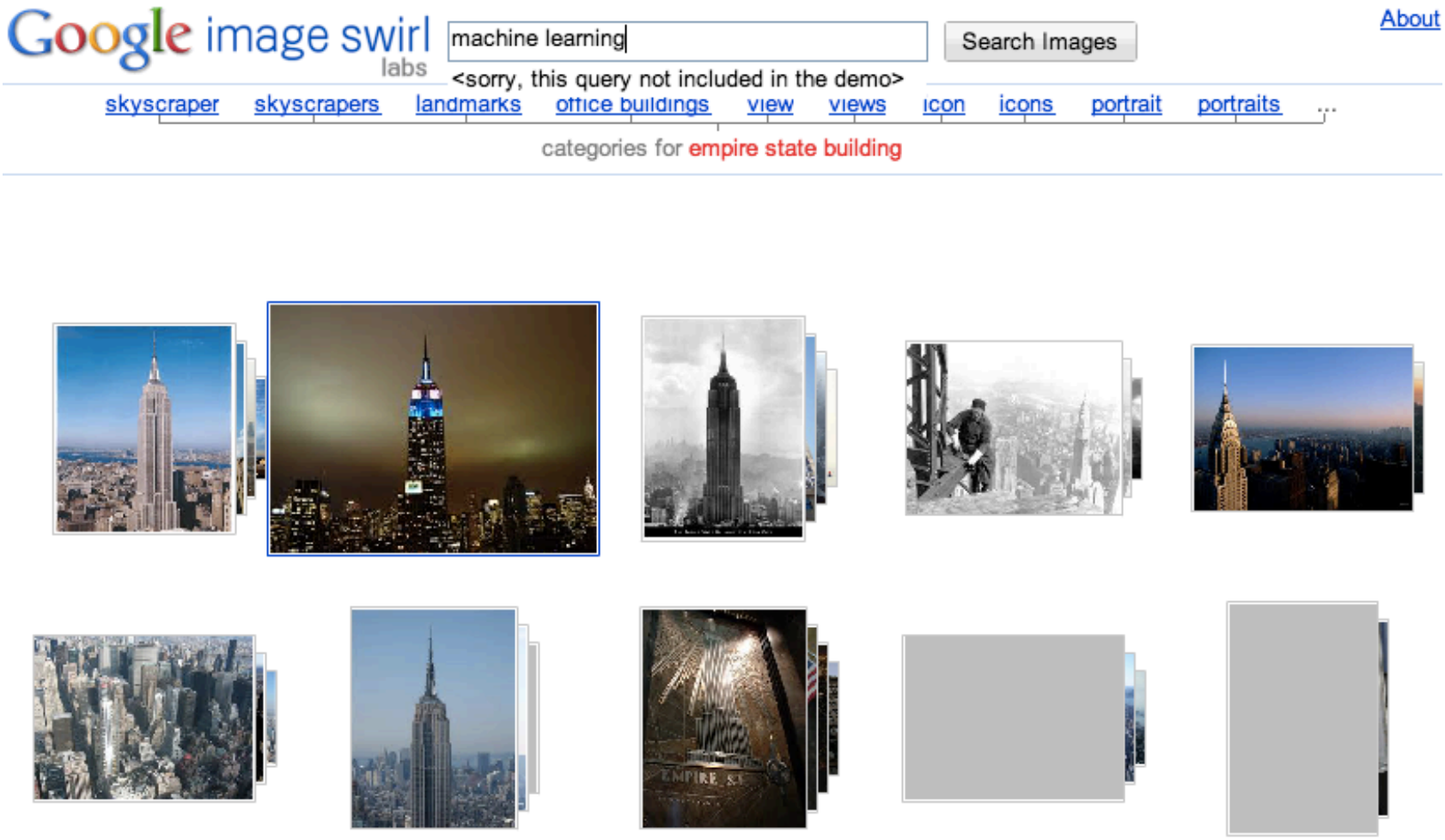
Similar Images - version 0.0

Demo




Google image swirl
labs
empire state
Search Images

“Machine Learning” comes up empty



The Web is Huge

- Image Swirl was precomputed and restricted to top 20K queries.
- People search on billions of different queries.
- Cannot precompute billions of distances

  → we need to do something smarter.

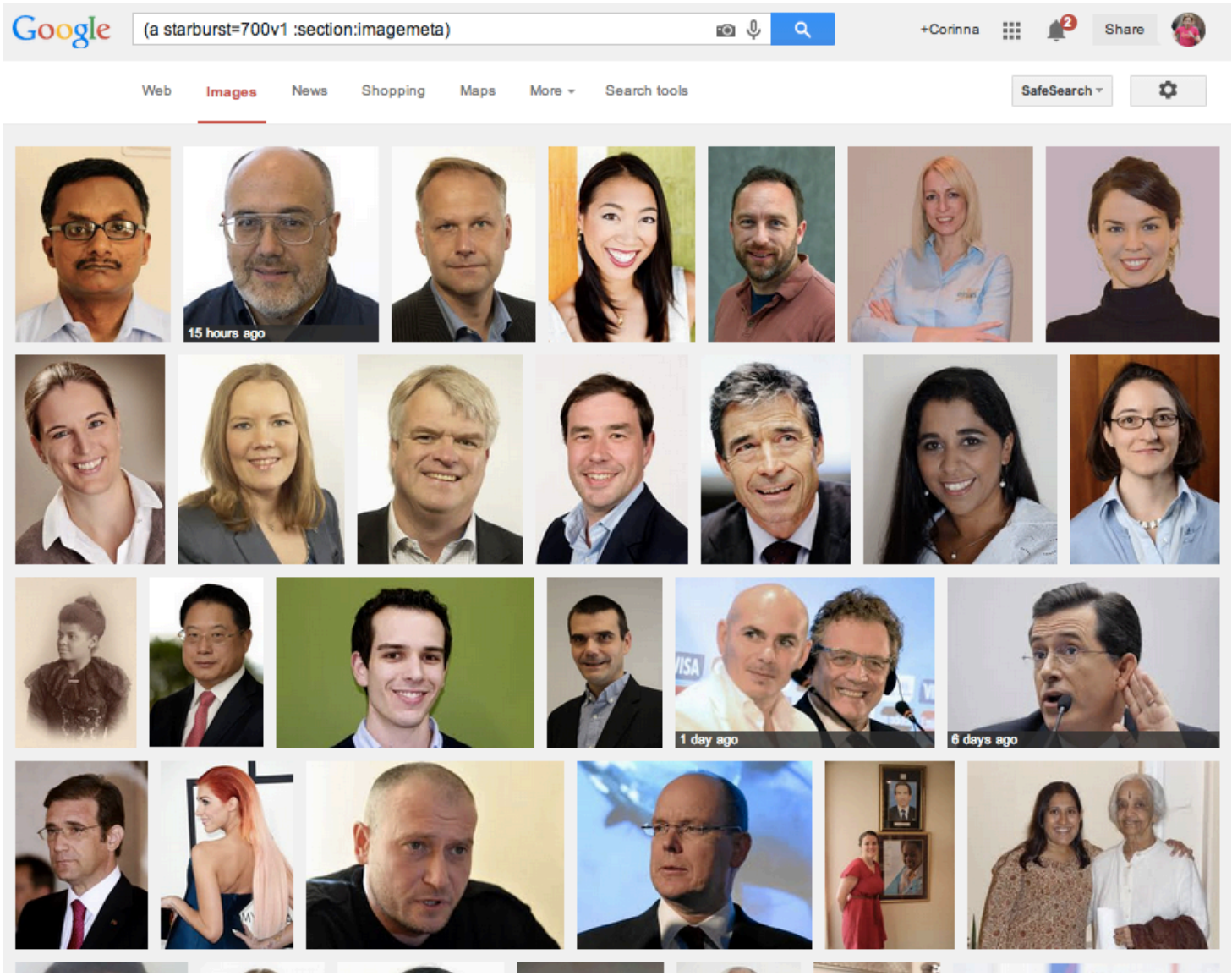
Approximate Nearest Neighbors

- Preprocessing:
  - Represent images by short vectors, kernel-PCA;
  - Grow tree top-down based on 'random' projections. Spill a bit for robustness;
  - Save the node ID(s) with each image.
- At query time:
  - propagate down tree to find node ID;
  - retrieve other image with same ID;
  - rank according to similarity with the short vector and other meta data.

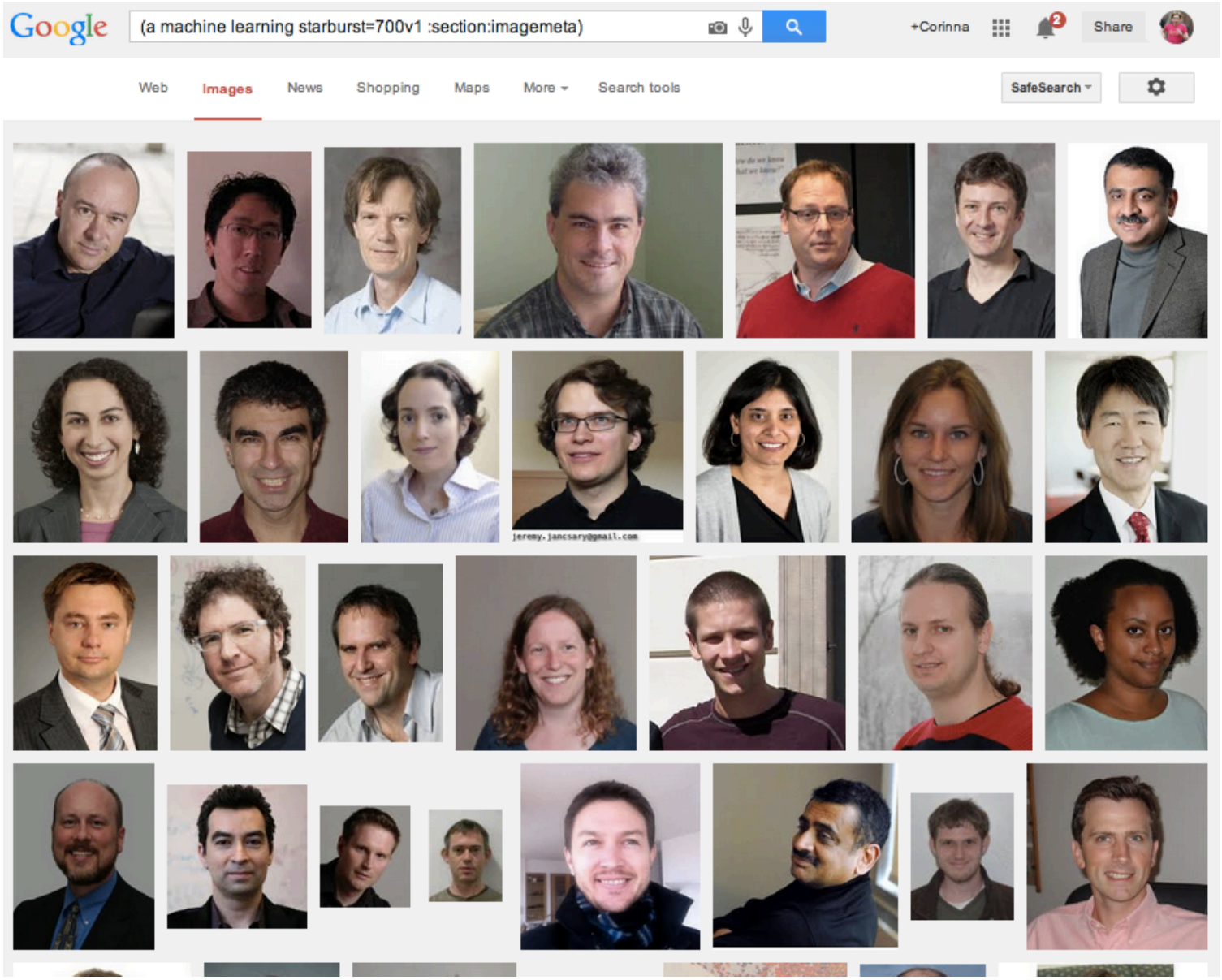
Similar Images, Version 1.0

- Live Search, Similar Images
- Inspect the single clusters
  - cluster 700
    - cluster 700 + machine learning
  - cluster 1000
  - cluster 400

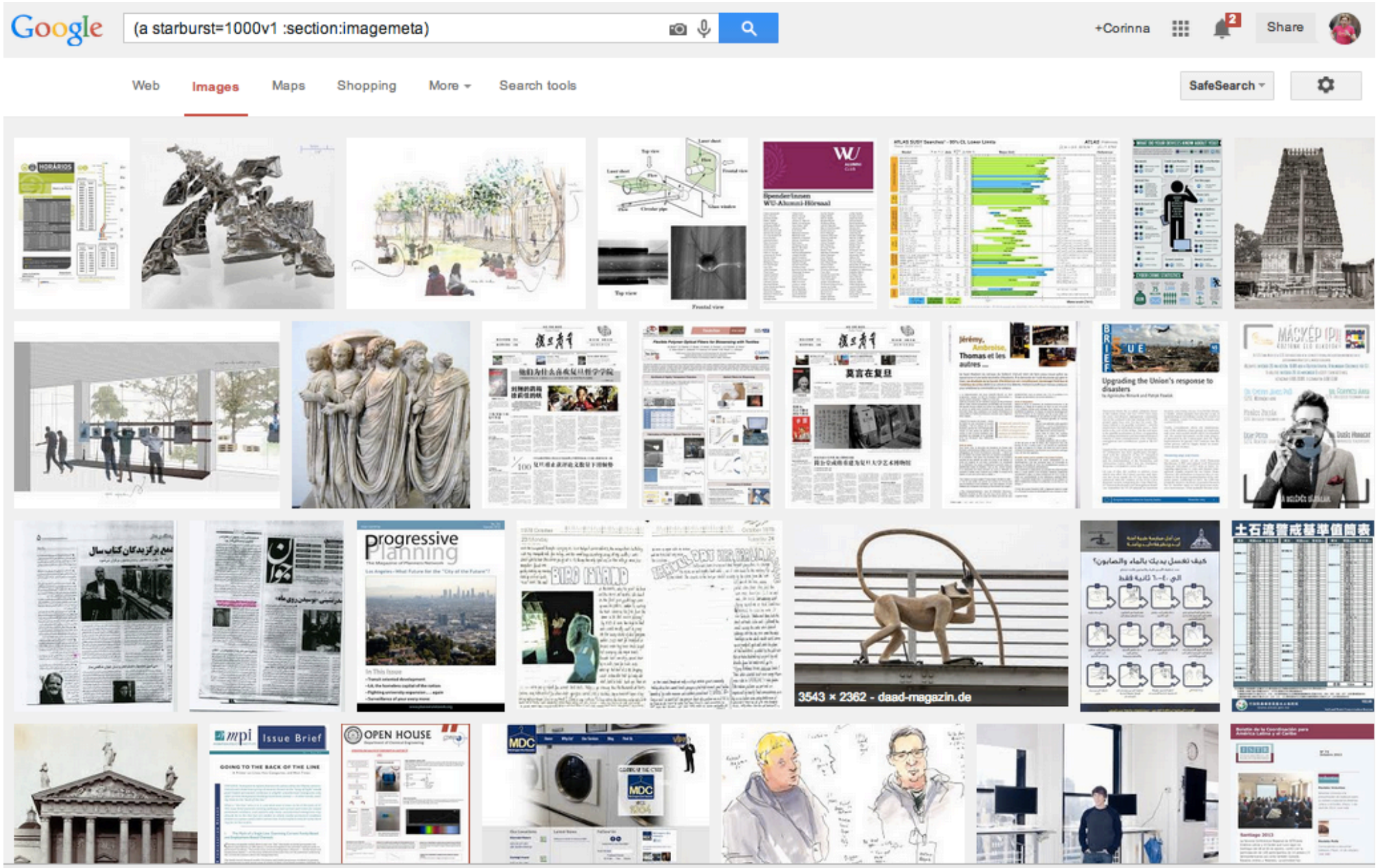
Cluster 700


Google
(a starburst=700v1 :section:imagemeta)
+Corinna
Share
Web
Images
News
Shopping
Maps
More
Search tools
SafeSearch
15 hours ago
1 day ago
6 days ago

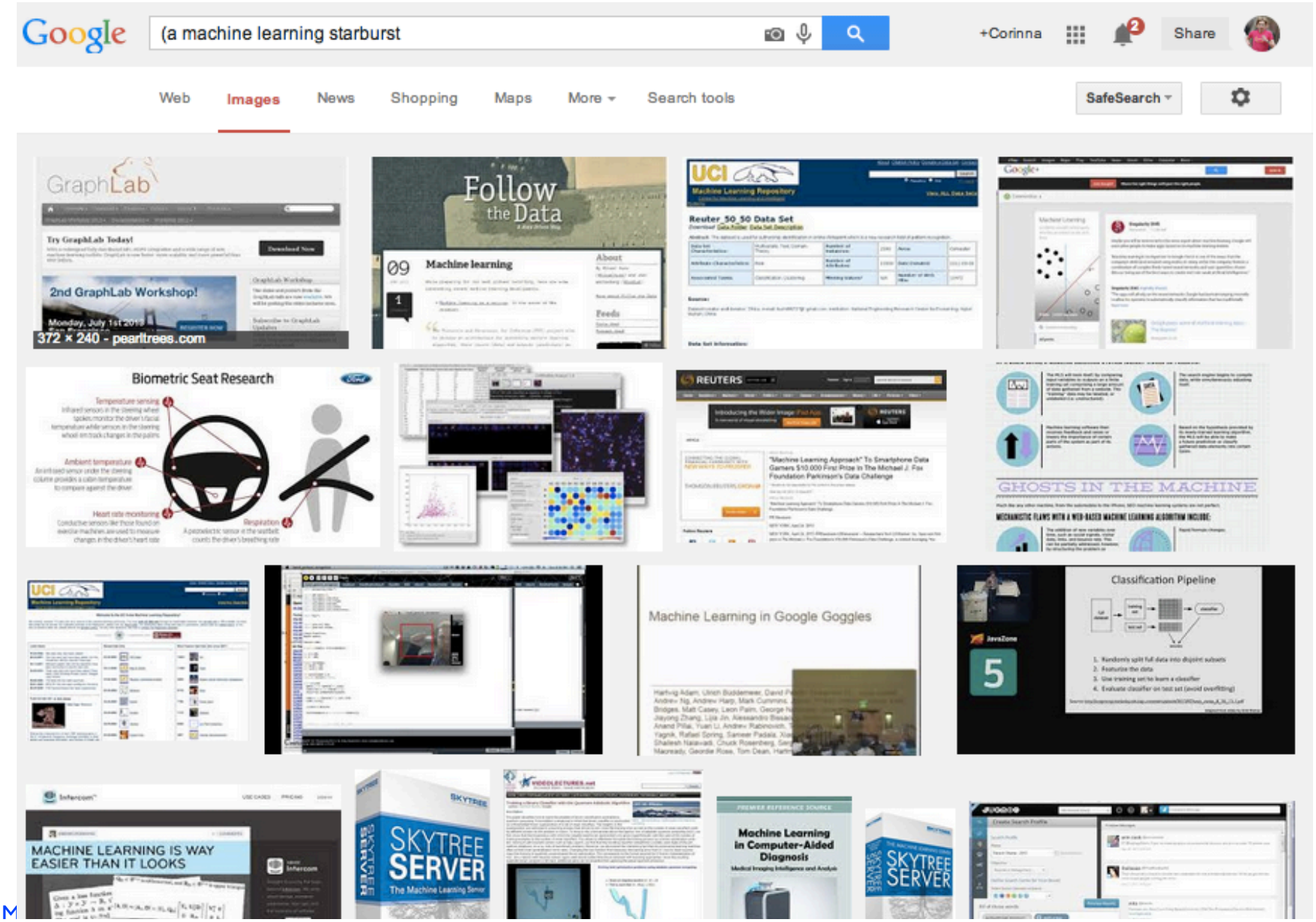
Cluster 700 + Machine Learning



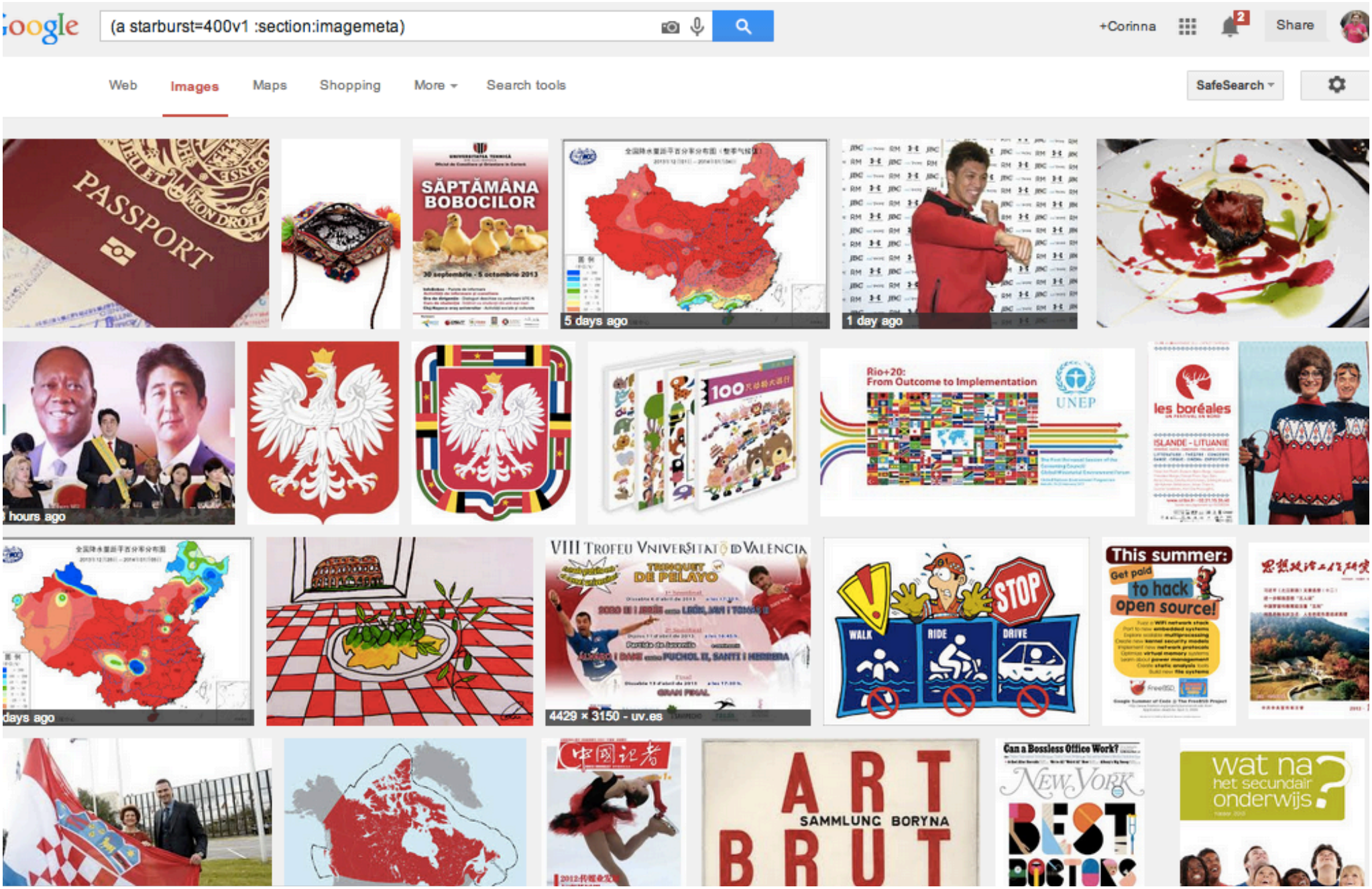
Cluster 1000



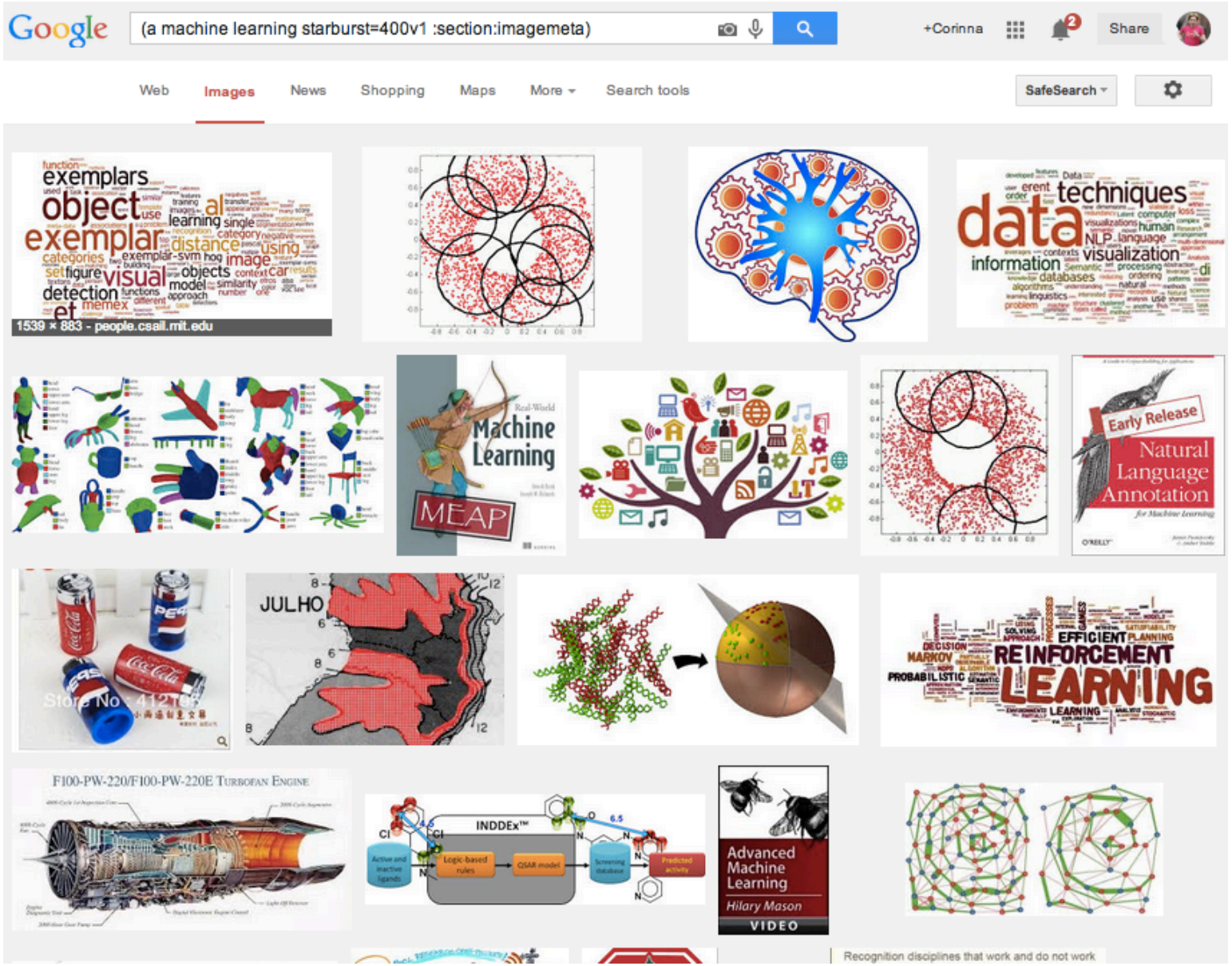
Cluster 1000 + Machine Learning



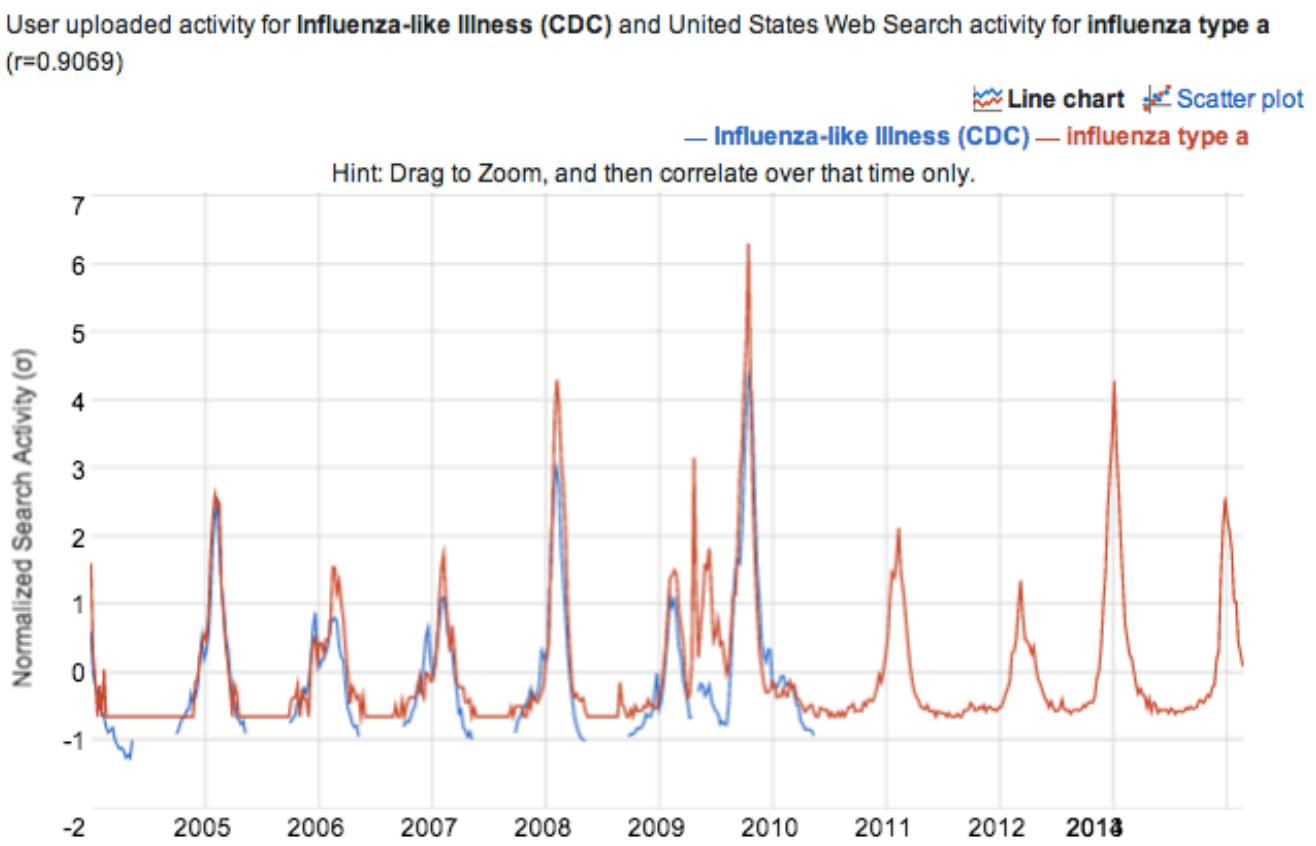
Cluster 400


Google
(a starburst=400v1 :section:imagemeta)
+Corinna
Share
Web
Images
Maps
Shopping
More
Search tools
SafeSearch
PASSPORT
SĂPTĂMÂNA BOBOCILOR
5 days ago
1 day ago
hours ago
Rio+20:
From Outcome to Implementation
UNEP
les boréales
ISLANDE - LITUANIE
days ago
VIII TROFEU VNIVERSITAT DE VALENCIA
TRIQUET DE PELAYO
4429 × 3150 - uv.es
STOP
WALK
RIDE
DRIVE
This summer:
Get paid
to hack
open source!
ART
SAMMLUNG BORYNA
BRUT
Can a Bossless Office Work?
NEW YORK
BEST
wat na
het secundair
onderwijs?

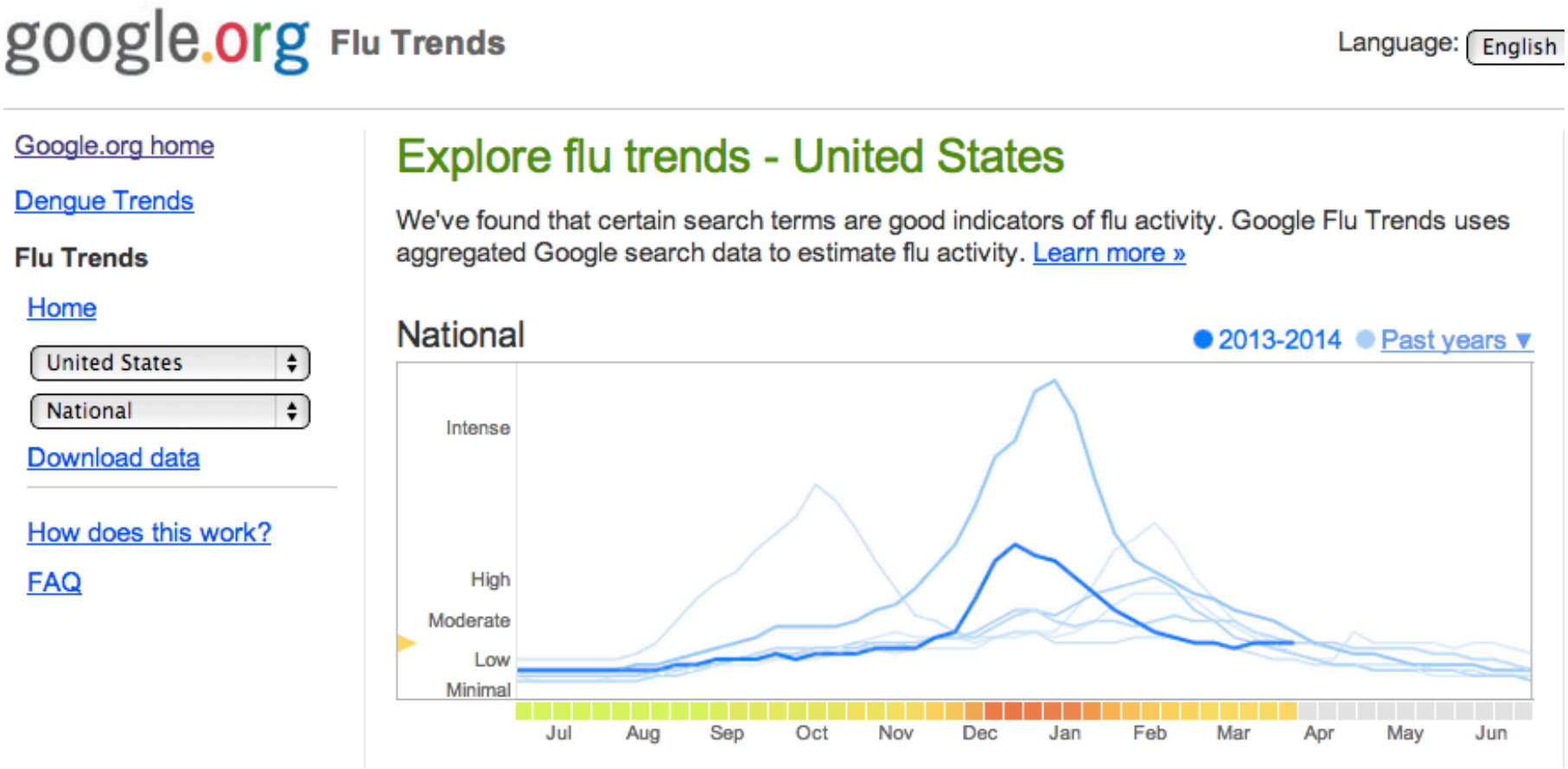
Cluster 400 + Machine Learning



Finding Similar Time Series



Flu Trends



http://www.google.org/flutrends/us/#US

Flu Trends

google.org Flu Trends

Language: English

Google.org home

Dengue Trends

Flu Trends

Home

United States

Cities (Experimental)

New York, NY

Download data

How does this work?

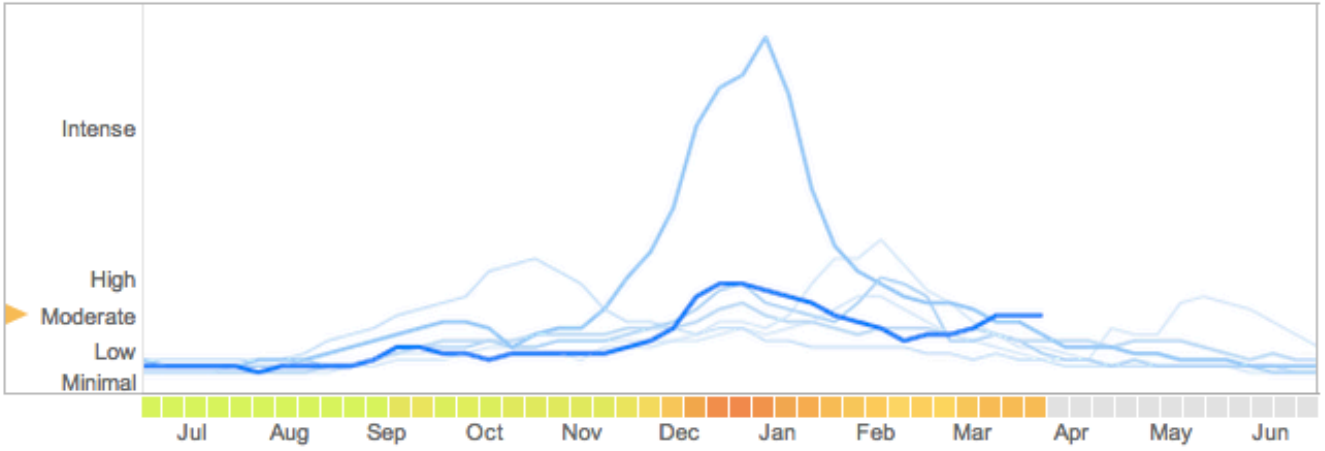
FAQ

Explore flu trends - United States

We've found that certain search terms are good indicators of flu activity. Google Flu Trends uses aggregated Google search data to estimate flu activity. Learn more »

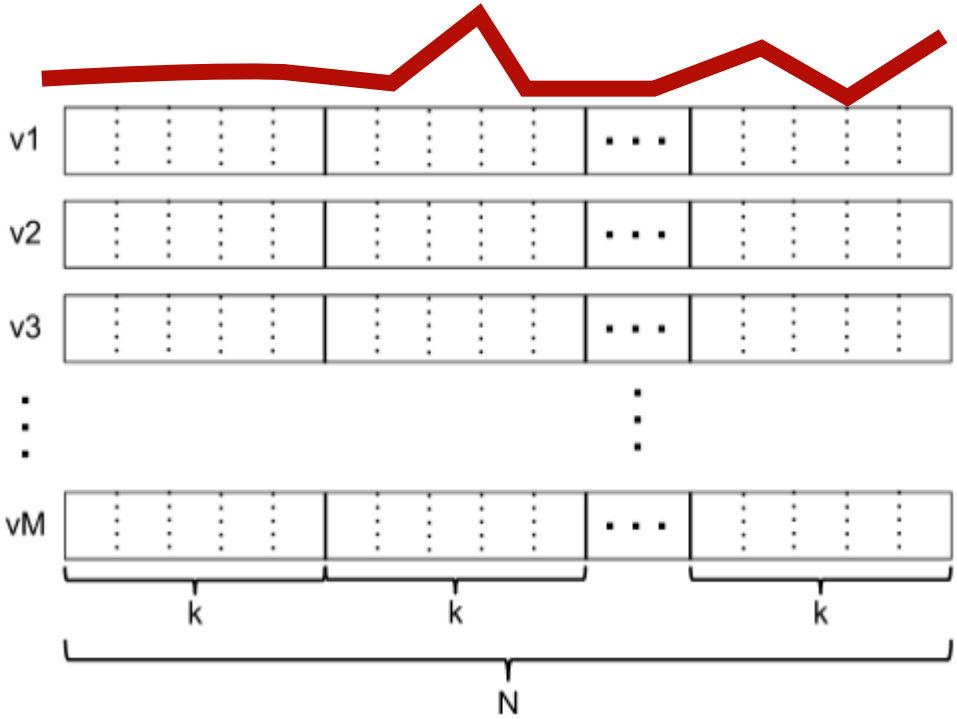
United States > New York, NY

2013-2014 Past years



Asymmetric Hashing, Indexing

- Split time series into N/k chunks:



- Represent each chunk with a set of cluster centers (256) using k-means. Save the coordinates of the centers, (ID, coordinates).
- Save each series as a set of closest IDs, hashcode.

Asymmetric Hashing, Searching

- For given input u, divide it into its N/k chunks, u_j:
  - Compute the N/k * 256 distances to all centers.
  - Compute the distances to all hash codes:

$$d^2(u, v^i) = \sum_{j=1}^{N/k} d^2(u_j, c(v_j^i))$$

- MN/k additions needed.
- The "Asymmetric" in "Asymmetric Hashing" refers to the fact that we hash the database vectors but not the search vector.

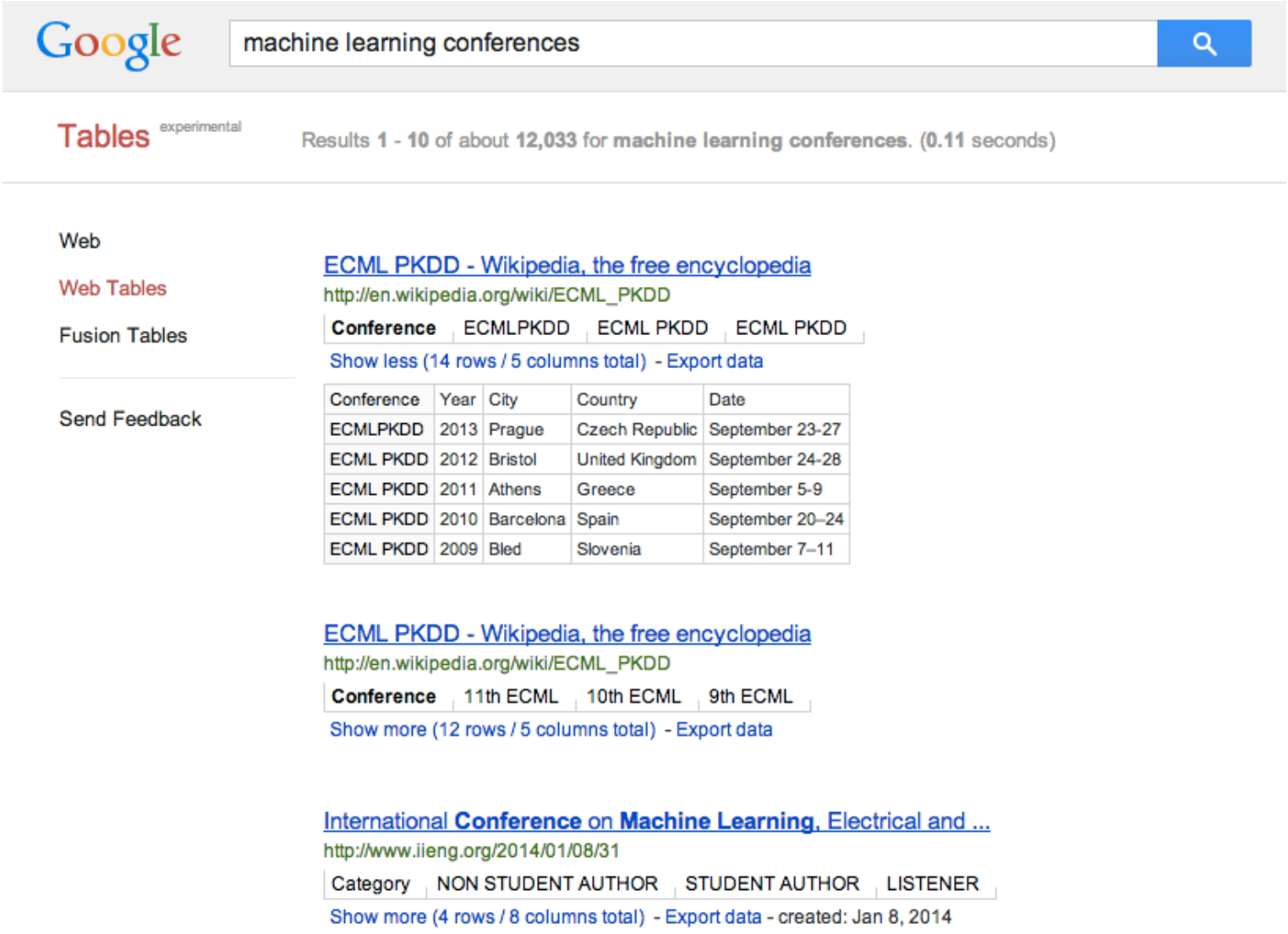
Google Correlate

- http://www.google.com/trends/correlate/

Outline

- Metric setting
  - Using similarities (efficiency)
  - Learning similarities (quality)
- Graph-based setting
  - Generating similarities

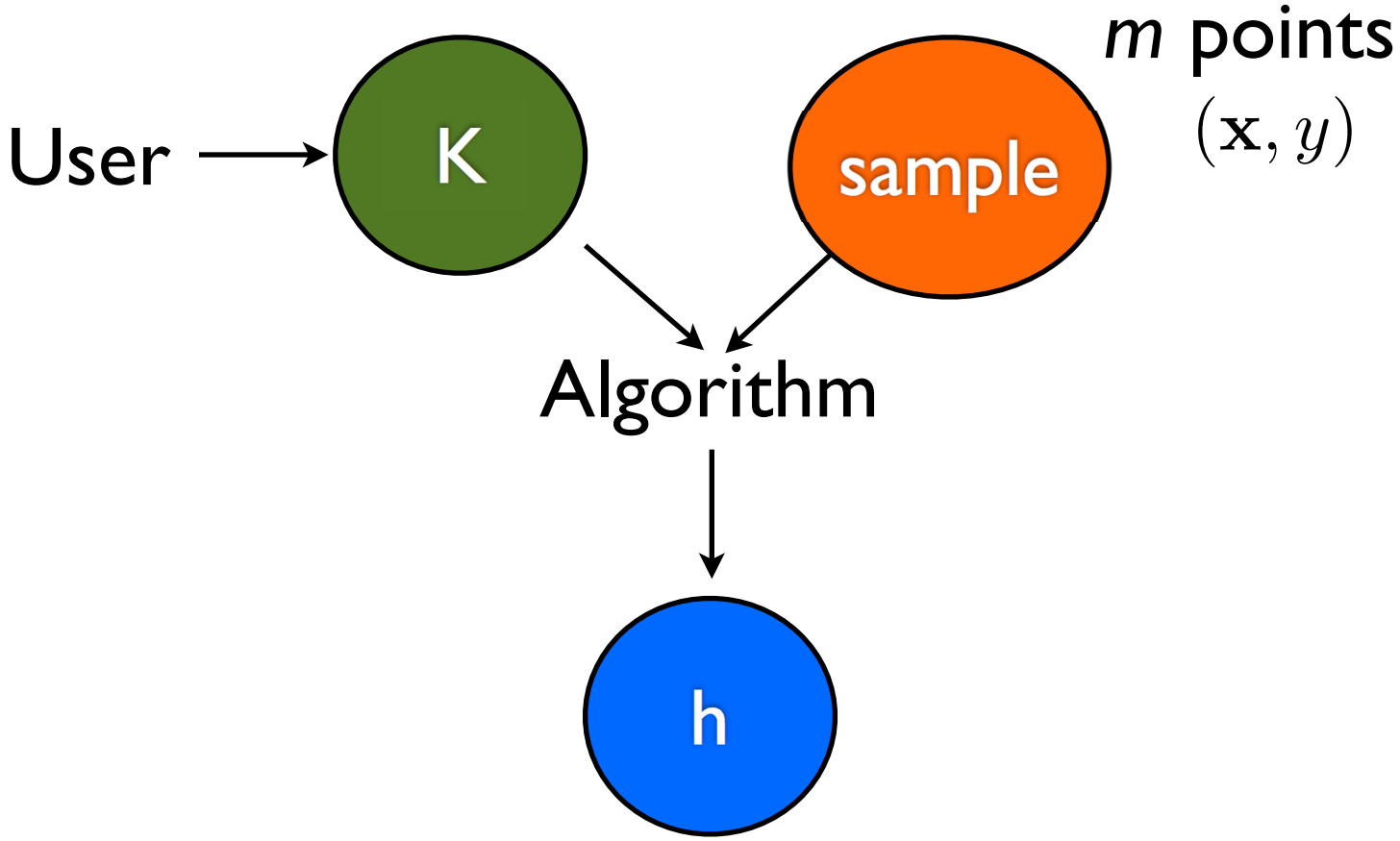
Table Search



http://webdatacommons.org/webtables/

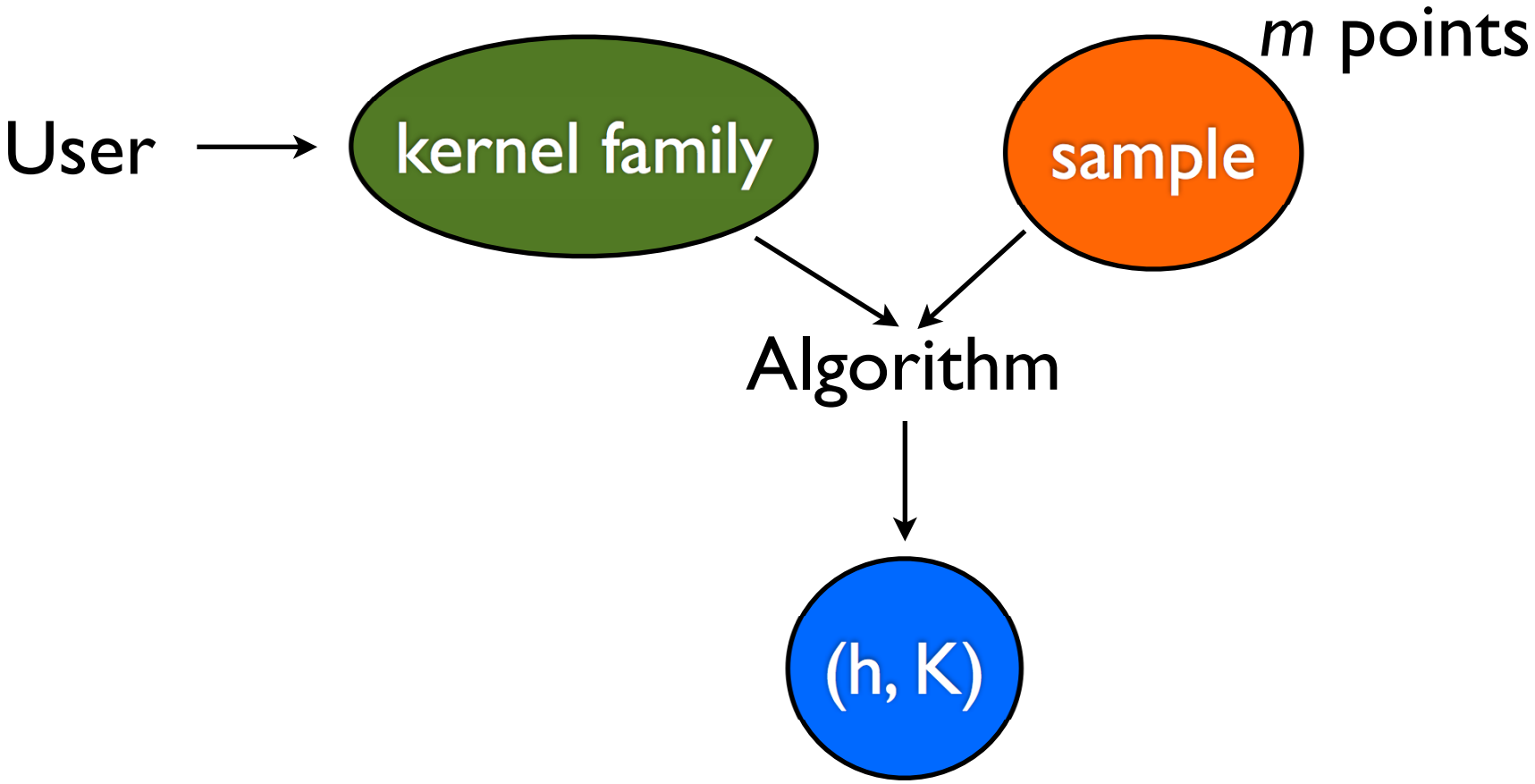
March 5, 2014: 11 billion HTML tables reduced to 147 million quasi-relational Web tables.

Standard Learning with Kernels



→ The user is burdened with choosing an appropriate kernel.

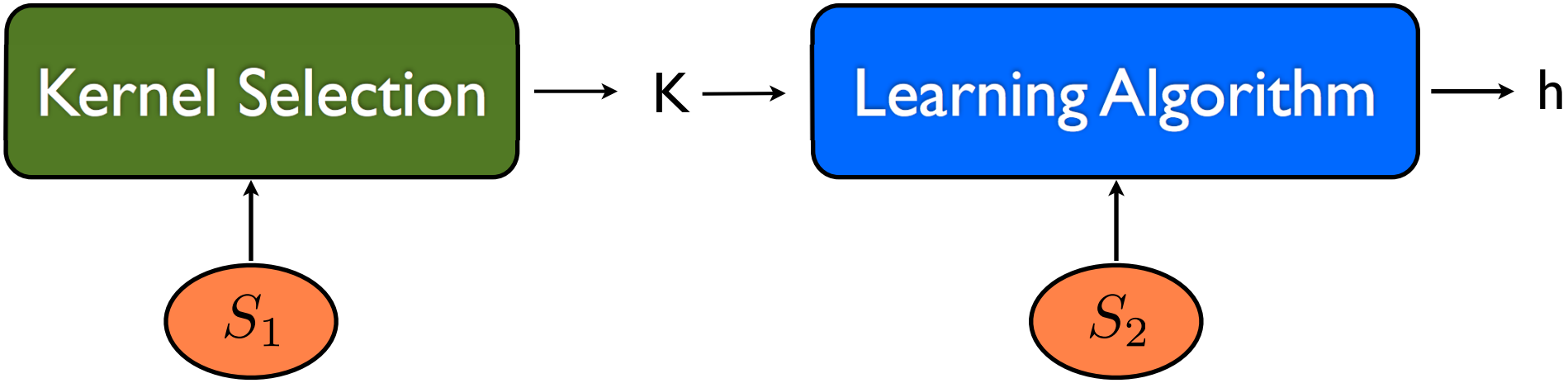
Learning Kernels



Demands less commitment from the user: instead of a specific kernel, only requires the definition of a family of kernels.

Centered Alignment-Based LK

- Two stages:



- Outperforms uniform baseline and previous algorithms.
- Centered alignment is key: different from notion used by (Cristiannini et al., 2001).

[C. Cortes, M. Mohri, and A. Rostamizadeh: Two-stage learning kernel methods, ICML 2010]

Centered Alignment

- Centered kernels:

$$K_c(x, x') = (\Phi(x) - \mathbf{E}_x[\Phi])^\top (\Phi(x') - \mathbf{E}_{x'}[\Phi])$$

- Centered alignment:

$$\rho(K, K') = \frac{\mathbf{E}[K_c K'_c]}{\sqrt{\mathbf{E}[K_c^2]\mathbf{E}[K_c'^2]}}$$

where expectation is over pairs of points.

- Choose kernel to maximize alignment with the target kernel:

$$K_Y(x_i, x_j) = y_i y_j.$$

Alignment-Based Kernel Learning

- Theoretical Results:
  - Concentration bound.
  - Existence of good predictors.
- Alignment algorithms:
  - Simple, highly scalable algorithm
  - Quadratic Program based algorithm.

Table Search

- http://research.google.com/tables
  - Machine Learning Conferences
  - Marathons USA
- Research Tool in Docs

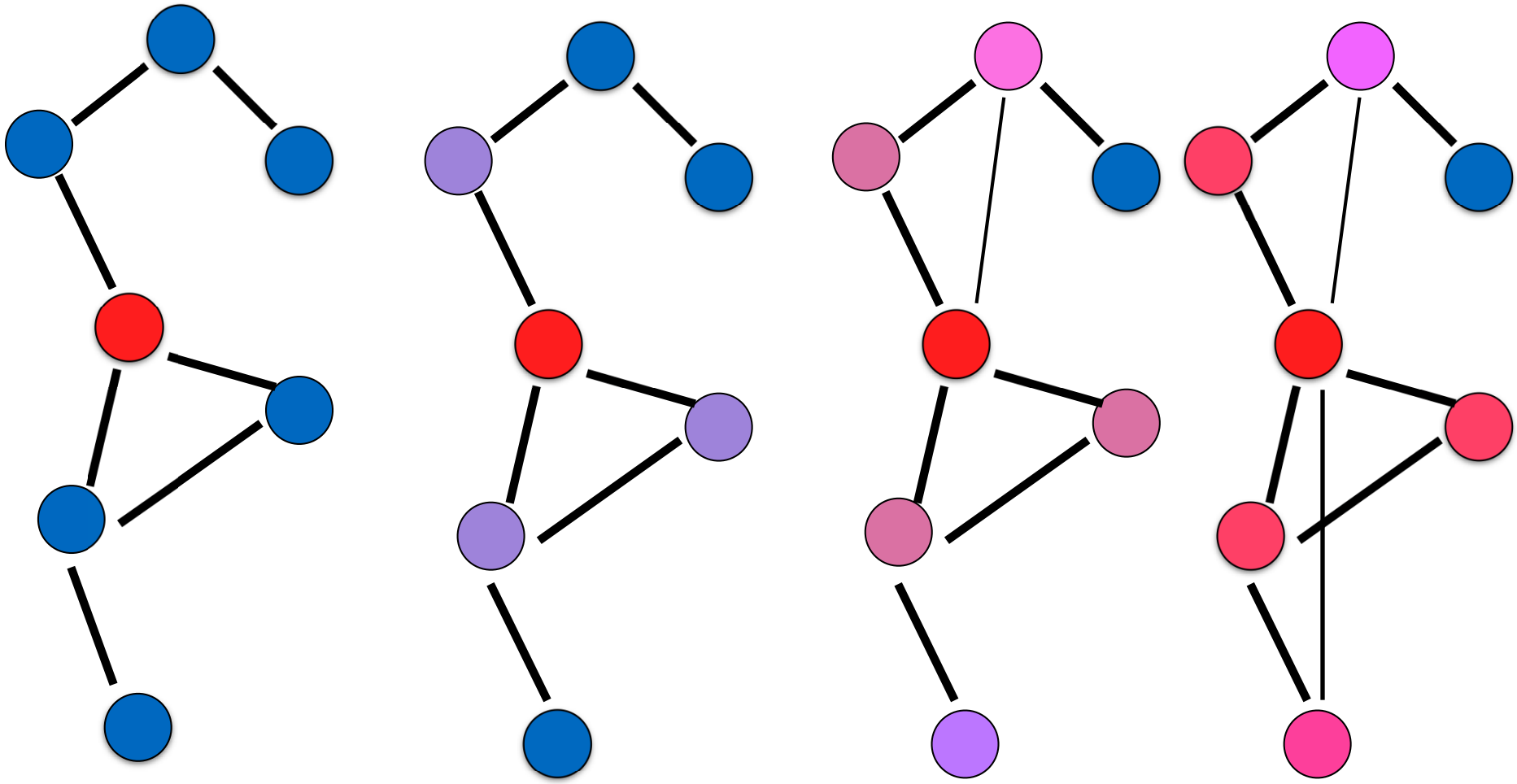
Outline

- Metric setting
  - Using similarities (efficiency)
  - Learning similarities (quality)
- Graph-based setting
  - Generating similarities

Graph-based setting

- Personalized Page Rank
  - PPR used to densify the graph
  - Single-Linkage Clustering used to find clusters in the graph

Example



PPR Clustering

- Personalized PageRank (PPR) of $u \longrightarrow v$:
  - Probability of visiting v in a random walk starting at u:
    - With probability $1 - \alpha$, go to a neighbor uniformly at random.
    - With probability α, go back to u;
- Resulting graph: single hops enriched with multiple hops, resulting in a denser graph. Threshold graph at appropriate level.

Algorithm

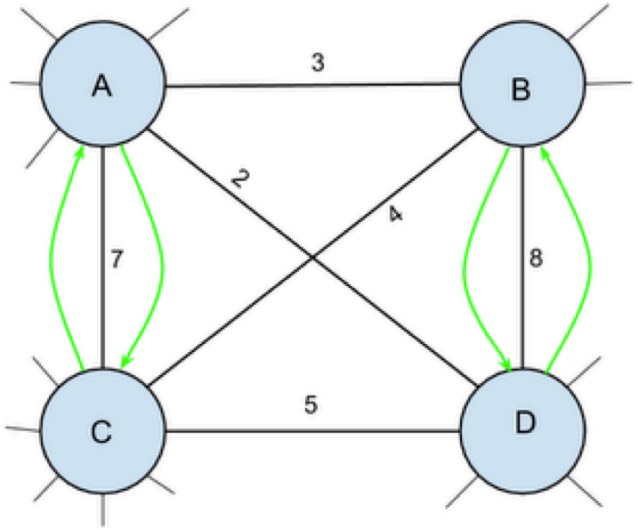
- 2 Pass Map-Reduce
  - Each push operation takes a single vertex u, moves an α fraction of the probability from r(u) onto p(u), and then spreads the remaining $(1 - \alpha)$ fraction within r, as if a single step of the lazy random walk were applied only to the vertex u.
  - Initialization: p = ~0 and r = indicator function at u

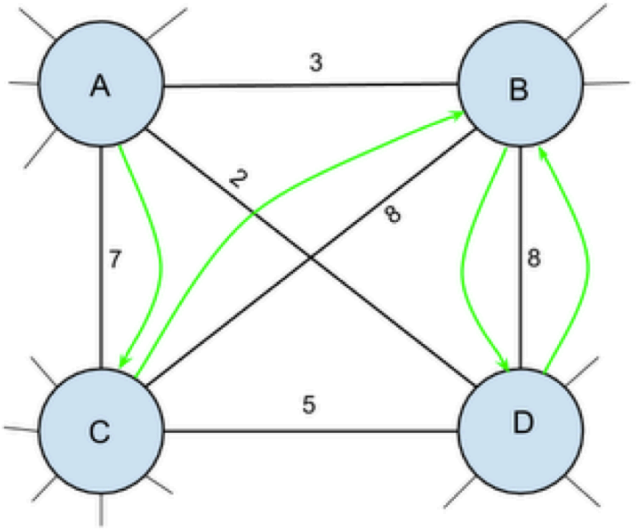
$\texttt{push}_u(p, r)$:

1. Let $p' = p$ and $r' = r$, except for the following changes:
   (a) $p'(u) = p(u) + \alpha r(u)$.
   (b) $r'(u) = (1 - \alpha)r(u)/2$.
   (c) For each v such that $(u, v) \in E$: $r'(v) = r(v) + (1 - \alpha)r(u)/(2d(u))$.
2. Return (p', r').

“Single-Linkage” Clustering

- Examples





- Can be parallelized;
- Can be repeated hierarchically.

Demo, Phileas, Landmark

- The goal of this project is to develop a system to automatically recognize touristic landmarks and popular location in images and videos. Applications include automatic geo-tagging and annotation of photos or videos. Current clients of the service include Personal Photos Search, and Google Goggles,
- Map Explorer

Summary

- Challenging very large-scale problems:
  - efficient and effective similarity measures algorithms.
- Still more to do:
  - learning similarities for graph based algorithms;
  - similarity measures vs discrepancy measures:
    - match time series on spikes
    - match shoes with shoes, highlighting differences.