Real-Time Analytics through Convergence of User-Defined Functions

Vinay Deolalikar
HP-Autonomy Research
Sunnyvale, CA

June 27, 2013

Outline

Unstructured Information Management

Explosive growth in unstructured data

- Already comprises about 80% of enterprise data
- Growing faster than structured data

Enterprises recognize role of unstructured data in decision making

Tools from data mining increasingly being adopted

Near Real-Time Workflows

Users want to analyze their data interactively

- Each step must be done in near real-time

A little less accuracy is not a deal-breaker

- . . . but high latency within workflows is

However, several powerful analysis techniques take time to converge!

Document Clustering

Clustering finds *inherent groupings* in data

- As opposed to classification, which learns groupings that user provides

Document clustering a powerful and versatile technique in unstructured information management

Several analytics use clustering at their core

- Organize document collections for management, browsing
- Organize search results
- Label document collections automatically

Document Clustering Takes Time

Clustering takes time

- Usually linear in the non-zero size of term-document matrix and in number of clusters
- Depending on how optimized the implementations are, how many clusters are requested, typical unstructured corpora can take minutes to hours

Usually, clustering-based analytics part of a larger workflow

- User is frequently waiting for the clustering to provide results
- Holds up the entire workflow

Outline

k-means

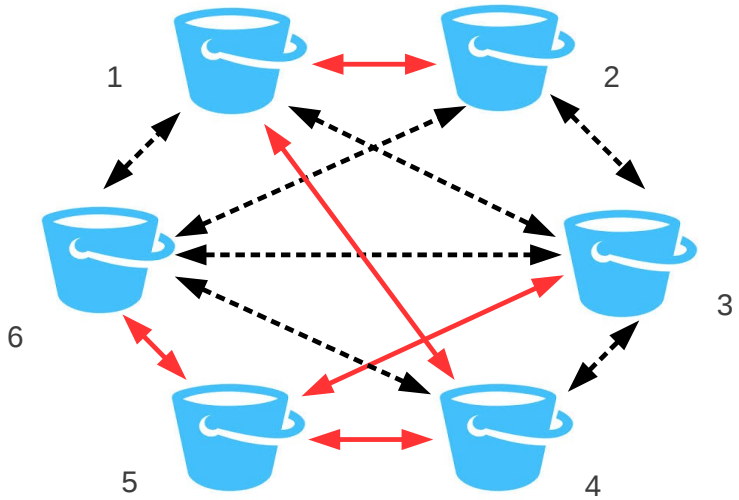
Arguably most important algorithm for clustering is *k*-means

We think of *k*-means as being in one of two states:

- Still running (aka, let's get coffee)
- Converged (aka, let's celebrate)

But is it really that simple?
(will return to this question later)

Let's First Visualize *k*-means



k-means as *flows*. Is this a two-state system?

Outline

What Does a User Care About?

Here is the objective function of k-means.

$$E(x, \mu) = \sum_{i=1}^{m} L(x_i, \mu_j), \tag{1}$$

(k-means minimizes this)

However, most users do not "think" in terms of such functions
(I hope)

Most enterprise applications of clustering do not care about this function

What Does a User Care About?

So what do users care about?

User cares about *meaning* of a cluster; contained in its set of *concepts*

Concept Flows: Idea

k-means is about documents moving around between clusters

How does meaning change when a document moves?

"Concept flow" when documents move between clusters C_i and C_j.

- We measure the presence of terms in the document that are concept labels for C_i and C_j
- Take the difference of the two-way deltas when a document moves: this is a measure of the "concept flow" associated to the movement of the document

Concept Flows: Properties

Concept flows are between pairs of clusters

Concept flows capture what user cares about

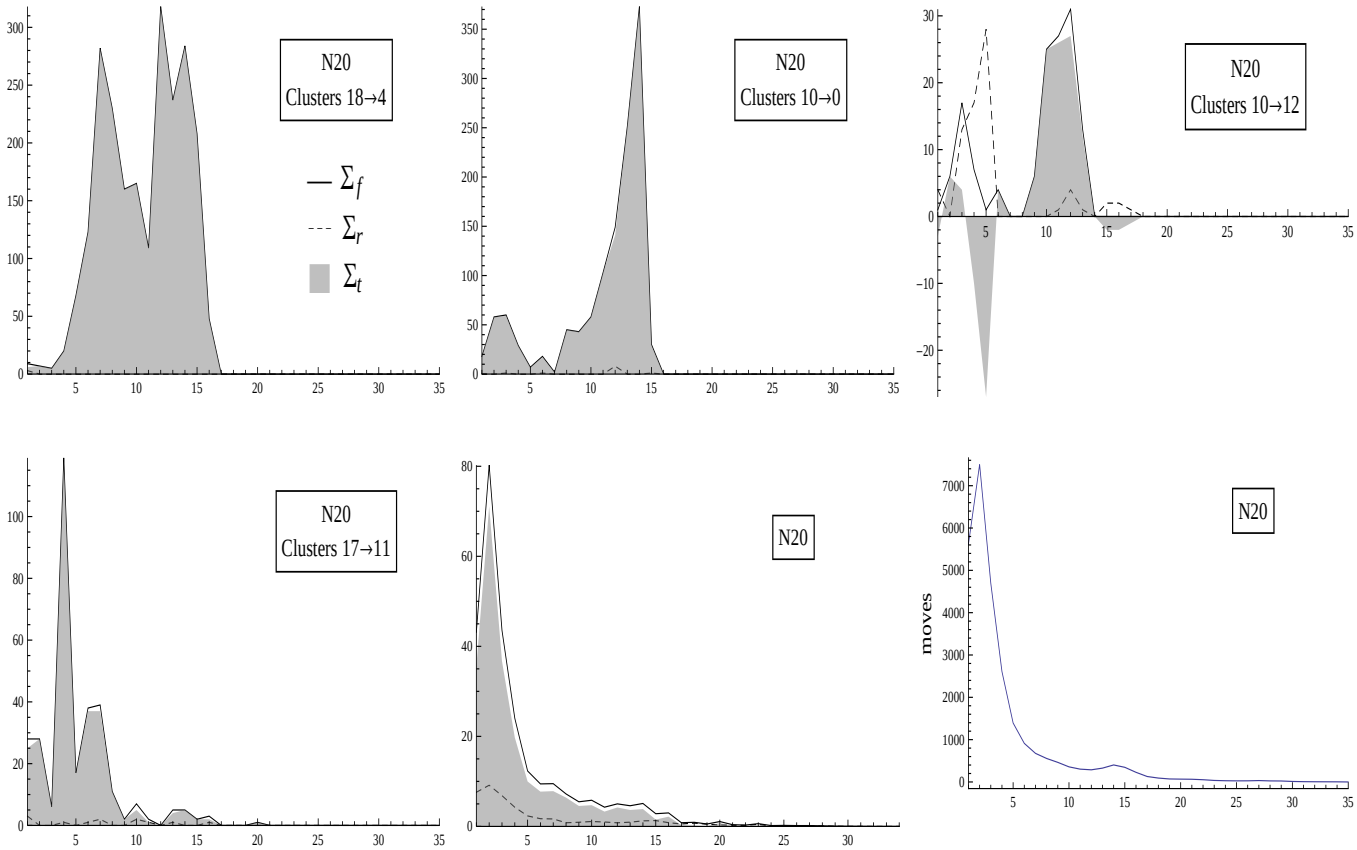
Concept flows can be measured locally

Concept flows can be measured cheaply

- Text is very high dimensional (1000's of dimensions)
- But concepts per cluster are few (~ 10)
- Each cluster can be calibrated to measure these

Outline

Concept Flows on 20 Newsgroups ($k = 20$)



Top four concept flows between pairs of clusters for a run of k-means for `N20`; average concept flows; document movements.

Outline

What can we Provide to User in Real-Time?

Most labels have already stablized by iteration 5!

Several workflows only need labels for first stage

- Scatter-Gather browsing

They can be provided this information in $1/10^{\text{th}}$ the time!

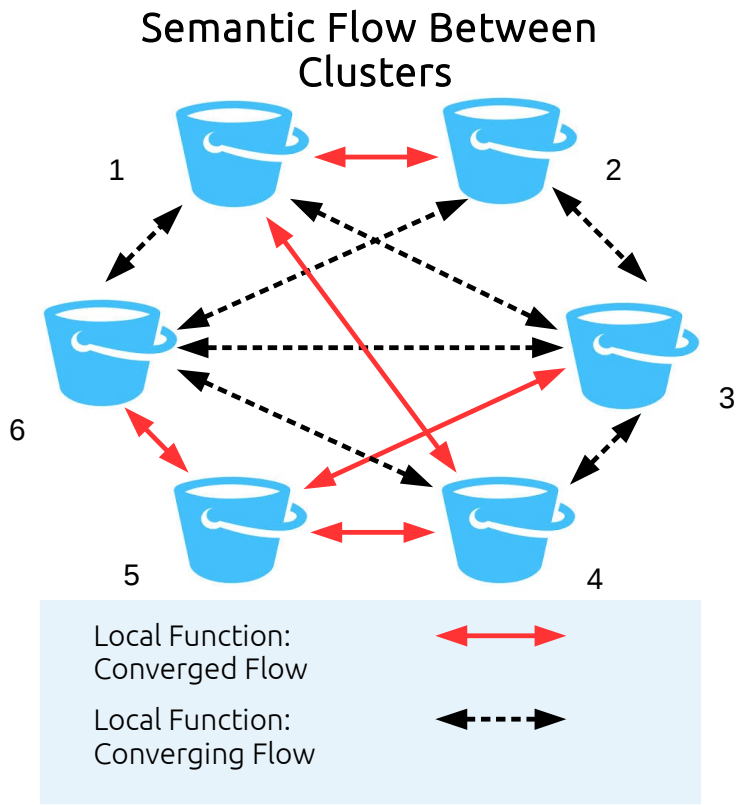
When Can We Provide Cluster?

Once concept flows have abated at a cluster, provide cluster

For majority of clusters, this happens by iteration 15

User can begin their next step well before final convergence

Is *k*-means a Two-State System?



k-means is a *set* of pairwise flows, most of which abate early. Labels can be computed by iteration 5. Once flows to and from a cluster have abated—as has happened to Cluster 5—, we may extract semantic meaning from it. This can happen very early during run-time, long before final convergence of *k*-means.