Clustering Applications at Yahoo!

Deepayan Chakrabarti
(deepay@yahoo-inc.com)
Outline

- Clustering, in itself, is often not the primary problem
  - Exploratory analysis is rarely needed
- Methods are often tied to the “final” goal
Outline

- Advertiser-keyword graph (+ social networks)
  - Graph Partitioning
- CTR predictions for ads on webpages
  - Co-clustering
- Query refinement and suggestions
  - Local search methods
- Conclusions
Outline

Advertiser-keyword graph (+ social networks)
- Graph Partitioning
- CTR predictions for ads on webpages
  - Co-clustering
- Query refinement and suggestions
  - Local search methods
- Conclusions
Graph Partitioning (Applications)

- Find clusters of advertisers and keywords
  - Keyword suggestions
  - Running experiments on some “natural” clusters

- Similar:
  - Y! Answers
  - Flickr
Graph Partitioning (Applications)

- Find clusters of IM users
  - Targeted advertising
  - Exploratory analysis

- Clusters of the Web Graph
  - Distributed pagerank computation

~100M nodes
Graph Partitioning (Methods)

- **Basic “Global” spectral partitioning [Ng+/01]**
  - Find 2\textsuperscript{nd} eigenvector of the graph Laplacian
  - This embeds all nodes on the real line
  - Split the line in two, to get two clusters
    - Can approximate the optimal conductance cut
  - For more clusters:
    - Use k eigenvectors (for known k), or
    - Split in two, and recurse on each cluster
Graph Partitioning (Methods)

- However, this has problems [Lang/05, Leskovec+/08]:
  - Min. conductance or quotient cuts lead to “small chunks”

Better balance ➔ worse cuts

Large-sized low-quotient cuts are actually just unions of whiskers

“Whiskers”
Graph Partitioning (Methods)

- However, this has problems:
  - Min. conductance or quotient cuts lead to “small chunks”
    - Balance is not very strongly encouraged
  - Recursive partitioning takes too long
    - Each eigenvector computation yields only a small cluster being broken off

- Two alternatives:
  - More balanced cuts \( \Rightarrow \) recursion is faster
  - Unbalanced cuts, but much faster computation
Graph Partitioning (Methods)

- Achieving better balance
  - Combine algorithms (e.g., [Anderson+/08])
    - METIS (more balanced cuts), followed by
    - Flow-based improvement (conductance)
  - Stronger balance constraints
    - Perfect balance on real line: NP-Hard
    - Spectral embedding
    - Perfect balance on hypersphere: SDP formulation [Lang/05]
Graph Partitioning (Methods)

- Faster Computation via “local” graph partitioning [Spielman+/04, Anderson+/06]
  - Pick seeds randomly
  - Build local clusters around seed
  - Bite off cluster, and repeat
  - Time for each iteration is proportional to the size of the local cluster (scalability)
  - Better for large graphs, or when not all clusters are needed
Graph Partitioning (Issues)

- Many complex networks are very different from planar/mesh-like networks
  - Good small cuts
  - Good large cuts are hard to find, and may not even exist
    - Hard to have a good hierarchical partitioning

- Is conductance really the best objective?
  - METIS+flow finds lower conductance cuts, but
  - Local spectral methods finds more “tightly-knit” cuts [Leskovec+/08]
  - Is there a good compromise?
Graph Partitioning (Issues)

- Speed and Scalability
  - Most implementations have trouble with large graphs, or graphs with some extremely high-degree nodes
  - Map-reduce style partitioning algorithms?
Outline

- Advertiser-keyword graph (+ social networks)
  - Graph Partitioning

- CTR predictions for ads on webpages
  - Co-clustering

- Query refinement and suggestions
  - Local search methods

- Conclusions
Co-clustering (Applications)

- The Content Match problem
  - Predict click-thru rate (CTR)
  - Extreme sparsity
    - Few views
    - Even fewer clicks

\[
\text{Webpages} \quad \text{Ads} \quad \text{clicks} \quad \text{views} = \text{CTR}
\]
Co-clustering (Methods)

- Approximate cell CTR using block CTR [Dhillon+/03]
- Minimize divergence between the original matrix and its reconstruction
- Combats sparsity
Co-clustering (Issues)

- Picking the right number of clusters
  - Use MDL [Chakrabarti+/2004]

- Handling new ads/pages
  - Combine co-clustering with feature-based prediction models [Agarwal+/2007]

- Handling extra information
  - E.g., if each page and ad can be categorized into a taxonomy [Chakrabarti+/2007]
  - Can the taxonomy be automatically modified?
Co-clustering (Issues)

- Iterative process (hard for map-reduce)
  - Factor-3 approximation by just clustering webpages and ads separately [Dasgupta+/2008]
Outline

- Advertiser-keyword graph (+ social networks)
  - Graph Partitioning
- CTR predictions for ads on webpages
  - Co-clustering
- Query refinement and suggestions
  - Local search methods
- Conclusions
Query Refinement

- User inputs ambiguous query ("madonna")
- Search engine asks: "Did you mean: songs, videos, pictures?"

Refinement = cluster of terms

lyrics "song title" album ...
Query Refinement

Suppose we could relate queries with keywords

- Madonna
- Beatles
- Honda
- Ford

Related Keywords

- songs
- lyrics
- albums
- pictures
- photos
Query Refinement (Problem)

- Different from plain bipartite graph partitioning
  - Don’t confuse users!
    - only 3 or fewer clusters for each query
    - only a few easily-distinguishable clusters overall

- Clustering quality now also depends on
  - the query workload
  - the algo that picks the “top-3” clusters for any query
Query Refinement (Method)

- Can optimally pick top-k clusters for any query [Wang+/2009]
  - for a wide range of matching functions
  - only if clusters are disjoint
- Iteratively improve clustering via local search
  - Move a keyword to a new cluster
  - Update top-k clusters for all queries in workload
  - Repeat
Query Refinement (Issues)

- Iterative ➔ slow
  - Each iteration has to go over the entire historical query logs
  - Optimality guarantees?
- Modeling issues:
  - How do we present a cluster to the user?
  - Cluster naming?
  - How does a user interact with a cluster?
Outline

- Advertiser-keyword graph (+ social networks)
  - Graph Partitioning
- CTR predictions for ads on webpages
  - Co-clustering
- Query refinement and suggestions
  - Local search methods

Conclusions
Conclusions

- Standalone clustering applications are rare!
  - Constraints on clustering
    - What use will it serve (as in query refinement)?
    - What is the desired “natural” cluster size?
  - Clustering for predictions (e.g., CTR)
    - Missing values
    - Extreme sparsity
  - Combining clustering with explore-exploit
Conclusions

- Even for standalone clustering:
  - Scalability concerns
    - 10s to 100s of millions of nodes
    - Skewed degree distributions
    - Algorithms amenable to map-reduce
  - The right “balance” relaxation
    - Tradeoffs between cluster “compactness”, conductance, and partition sizes
    - Is it even reasonable to require balance?
References

- On Spectral Clustering: Analysis and an Algorithm, by Ng, Jordan, & Weiss, in NIPS 2002
- Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems, by Spielman & Teng, STOC 2004
- Fixing two weaknesses of the Spectral Method, by Lang, NIPS 2005
- Local Graph Partitioning using PageRank Vectors, by Anderson, Chung, & Lang, SODA 2008
- An algorithm for improving graph partitions, by Anderson & Lang, SODA 2008
- Finding dense and isolated submarkets in a sponsored search spending graph, by Lang & Anderson, CIKM 2007
- Clustering of bipartite advertiser-keyword graph, by Carrasco, Fain, Lang, & Zhukov, IEEE Computer Society, 2003
- Statistical Properties of Community Structure in Large Social and Information Networks, by Leskovec, Lang, Dasgupta, & Mahoney, WWW 2008
- Approximate Algorithms for Co-Clustering, by Anagnostopoulos, Dasgupta, & Kumar, PODS 2008
- Information-theoretic co-clustering, by Dhillon, Mallela, & Modha, in KDD 2003
- Fully Automatic Cross-Associations, by Chakrabarti, Papadimitriou, & Faloutsos, in KDD 2004
- Predictive discrete latent factor models for large scale dyadic data, by Merugu, & Agarwal, in KDD 2007
- Estimating Rates of Rare Events at Multiple Resolutions, by Agarwal, Broder, Chakrabarti, Diklic, Josifovski, & Sayyadian, in KDD 2007
- Mining Broad Latent Query Aspects from Search Sessions, by Wang, Chakrabarti, & Punera, in KDD 2009