Tools for Personalized Search

Devdatt Dubhashi
Department of Computer Science and Engineering
Chalmers University
Gothenburg, Sweden.

The Future of Web Search, Bertinoro, June 17–20, 2007

Outline

Outline

Outline

Outline

1. Introduction: Personalization Techniques
2. Ranking via Flows
3. co-Clustering

Review of Pagerank

- $\mathbf{C} := \alpha \mathbf{P} + (1 - \alpha)\frac{\mathbf{1}}{N}$.
- Uniform random surfer model.

Personalizing Pagerank: Teleport vector

- $\mathbf{C} := \alpha \mathbf{P} + (1-\alpha)\mathbf{v}$.
- Persomalization vector **v**.

Outline

Network flow based ranking

- Given a directed graph $G = (V, E)$, a flow

  $$f : E \to \mathbf{R},$$

  defines a ranking on the nodes.
- The rank of a node $v \in V$ with respect to the flow f is the inflow

  $$f(v) := \sum_{(u,v) \in E} f(u, v).$$

- Very general framework: the graph G can be the Web (with teleport node etc.) or a biological network (gene interactions) or a semantic entity-relationship graph, [1]

Example: Pagerank as Flow

- Given the transition matrix P of a reversible Markov chain, with the stationary distribution π, let

$$f(u,v) := \pi_u P(u,v).$$

- The ranking defined by this flow is exactly π:

$$\begin{aligned} f(v) &= \sum_u \pi_u P(u,v) \\ &= \sum_u \pi_v P(v,u) \\ &= \pi_v. \end{aligned}$$

Incorporating User Preferences

- Users are allowed to express preferences in the form:

$$u \prec v.$$

- Such preferences could be collected via clickthrough data (Joachims 05, 07).
- Given a set of such preferences $\mathcal{P}$, can they be incorporated into the ranking?
- Goal: Find a ranking which is as "close" to pagerank as possible, and also respects user preferences $\mathcal{P}$.

User Preferences: Convex Optimization

Chakrabarti *et al* [2] following others:

- Minimize the Kullback-Liebler Divergence

$$\min D(f||q) = \sum_{(u,v)} f(u,v) \log \frac{f(u,v)}{q(u,v)}$$

- with respect to the reference flow q say Pagerank,
- Subject to user preferences:

$$\sum_{(w,u)\in E} f(w,u) \leq \sum_{(w,v)\in E} f(w,v), \quad (u \prec v) \in \mathcal{P}.$$

Acceleration: Incremental Computation

- Rather than solving the program from scratch, update solution to reflect new preferences.
- Compute a sequence $f_0, f_1, \cdots$ where
- f_0 is the reference flow (Pagerank)
- f_{i+1} is computed by updating f_i to reflect a new preference $u \prec v$.
- Intuitively, f_{i+1} will be a “small” perturbation of f_i and hence can be computed fast.
- We use the fact that f_i is still dual feasible.

Acceleration: Local Computation

- Also f_{i+1} is likely to differ from f_i only locally in a neighbourhood of the newly expressed preference $u \prec v$.
- Hence we may hope to compute f_{i+1} from f_i by a local update on a small subgraph H around u and v.
- If H is much smaller than the whole graph, this results in significant savings.
- Corresponding to a new preference $u \prec v$, H will include the connected components of u and v in $\mathcal{P}$.
- Fix the flows between H and $G \setminus H$ (constraints) and solve optmization problem on H.

Outline

Co-clustering

- Co–clustering is simulataneous clustering of objects in different categories [3]
- $G := (V_1, V_2, \cdots, V_k, E)$ a multi–partite graph corresponding to k different categories. (The edge set E may also include intra-class edges.)
- Example: People versus Movies, with $k = 2$.
- A co–clustering is a simultaneous clustering withinh each of the classes $V_1, \cdots V_k$.
- Different from separate clusterings in every class.
- A good clustering in one class can help clustering in the other classes via the relations in E: serves as a kind of dimensionality reduction.

Application: Collaborative Filtering

- Users rate some movies.
- Based on these ratings, we can predict ratings for other movies.
- Windghager, Tansini *et al* apply it to data from the Hungarian web.

Application: Mining User Intent from Weblogs

- Users enter
- Queries which retrieve
- Documents
- Co-clustering can help detecting user intent.
- Tansini applies it to TodoCl data.

Agglomerative Iterative co-Clustering

- Label each object according to its link structure to classes in the other categories.
- Apply an agglomerative clustering technique based on thie representation.
- Iterate. In each iteration, the representation of an object is different corresponding to the current clustering in the other classes.

Information-theoretic co-clustering

- Dhillon *et al* [4] give an information–theoretic criterion for co–clustering.
- Maximize the mutual information between random variables correpsonding to the clusterings in different classes.
- Results in a convex optimization problem.
- Local gradient approaches.
- Randomized rounding approaches.

Semi-supervised Transductive Approaches

- Typically, labels on some objects may be known.
- Treat co–clustering as a semi–supervised or transductive learning problem.
- Conditional Random Fields.
- Label Propagayion via Graph Laplacians.
- Geometric regularization ...

For Further Reading I

- J.A. Tomlin
  " A New Paradigm for Ranking pages on the World Wide Web". WWW'03, 2003.
- A, Agrawal, S. Chakrabarti and S. Aggarwal
  " Learning to Rank Networked Entities", KDD'06, 2006
- A. Tanay, R. Sharan and R. Shamir
  " Biclustering Algorithms: A Survey" in *Handbook of Computational Molecular Biology*, S. Aluru, editor, pp. 26-1 - 26-17, Chapman and Hall / CRC Press 2006
- I. S. Dhillon, S. Mallela, and D. S. Modha
  " Information-Theoretic Co-clustering" KDD'03, 2003