

Multi-Objective Ranking of Comments on Web

Onkar Dalal^{*}
Stanford University
California, USA
onkar@stanford.edu

Srinivasan H Sengamedu
Komli Labs
Bangalore, India
shs@komli.com

Subhajit Sanyal
Big Data Labs, American
Express
Gurgaon, India
Subhajit.Sanyal@aexp.com

ABSTRACT

With the explosion of information on any topic, the need for ranking is becoming very critical. Ranking typically depends on several aspects. Products, for example, have several aspects like price, recency, rating, etc. Product ranking has to bring the “best” product which is recent and highly rated. Hence ranking has to satisfy multiple objectives. In this paper, we explore multi-objective ranking of comments using Hodge decomposition. While Hodge decomposition produces a globally consistent ranking, a globally inconsistent component is also present. We propose an active learning strategy for the reduction of this component. Finally, we develop techniques for online Hodge decomposition. We experimentally validate the ideas presented in this paper.

Categories and Subject Descriptors

H.m [Information Systems]: Miscellaneous

General Terms

Algorithms

Keywords

Multi-Objective Ranking, Hodge Decomposition, Active Learning

1. INTRODUCTION

The amount of information available on the Internet on any topic is ever increasing. Hence the amount of information relevant to any topic is also on the increase. Ranking plays a crucial part in selecting relevant information. Ranking has traditionally been posed as a problem of matching queries to documents. Subsequent approaches to ranking emphasized document quality/authority in addition to matching. With the availability of complex data and information finding requirements, there is a realization that several aspects are important to ranking. In this paper, we address multi-objective ranking.

Multi-objective ranking is relevant to several applications – product ranking, video ranking, etc. We consider ranking

^{*}This work was conducted when the authors were with Yahoo! Labs, Bangalore

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

of comments on news articles. In a typical scenario, users comment on news articles. Users also rate the comments with “thumbs up” and “thumbs down”. The goal of ranking is to increase user engagement – roughly measured by the fraction of “thumbs up”. A simple strategy is to rank comments by the number of thumbs up they receive. This does not work well since new comments do not get sufficient ratings. Hence we resort to other measures like the quality of comments or reputation of commenters. So ranking of comments has to balance several objectives – actual rating, recency, commenter reputation, comment quality etc. – to arrive at a single ranking.

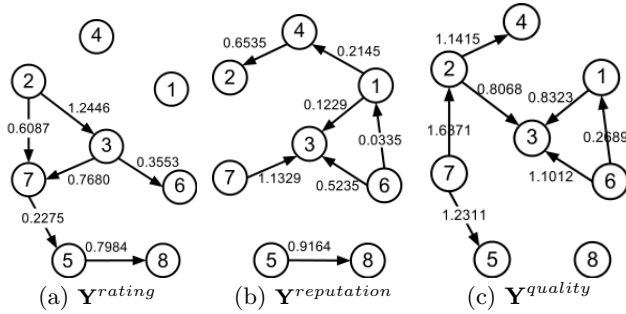
EXAMPLE 1. *We use Figure 1 as a running example in this paper. This illustrates the ranking of 8 comments to a news article in Yahoo News. We consider three ranking aspects: current rating, commenter’s reputation, and comment quality. For each of these aspects, each comment gets a score. These scores are incomplete because of the following: new comments have insufficient ratings, some commenters have no history of commenting, and we do not calculate quality scores for very short comments. We can construct a graph for each aspect. The nodes of the graph are comments and there is an edge between any pair of nodes if both the nodes have scores. See Figures 1(a), 1(b), and 1(c).* □

It is well-known such multi-aspect data can be have conflicts in the sense different aspects can have conflicting preferences or even be “cyclic”.

EXAMPLE 2. *Comments 2 has higher rating compared to comment 7 but the latter has a higher quality compared to the former. See Figures 1(a) and 1(c). In Figure 1(h) (which represents the harmonic flow as discussed in Section 2.2), comment 2, 3, 1, and 4 form a cycle: 4 is better than 2, 3 is better than 1, 1 is better than 4, and 4 is better than 1.* □

Such cyclic preferences are inherent in social choice data. In other scenarios, the cyclic preferences are artifacts of incomplete data and it is possible to make the data more complete at a cost. For example, in case of comments, it is possible to obtain more ratings for a comments (or pairs of comments) by ranking it (or them) high.¹ So if we identify critical dependencies and get more data (by displaying the comments, for example), we can resolve the critical conflicts and arrive at globally consistent ranking faster. This is related to active learning using explore-exploit or editorial strategies but in the Hodge framework.

¹Such exploration is possible with other data too. For example, In product ranking, popularity information for a product can be determined by showing the product more often.



EXAMPLE 3. In Figure 1(h), we identify that the information about relative preference between comments 1 and 2 is important. Getting this information reduces the cyclic preference. □

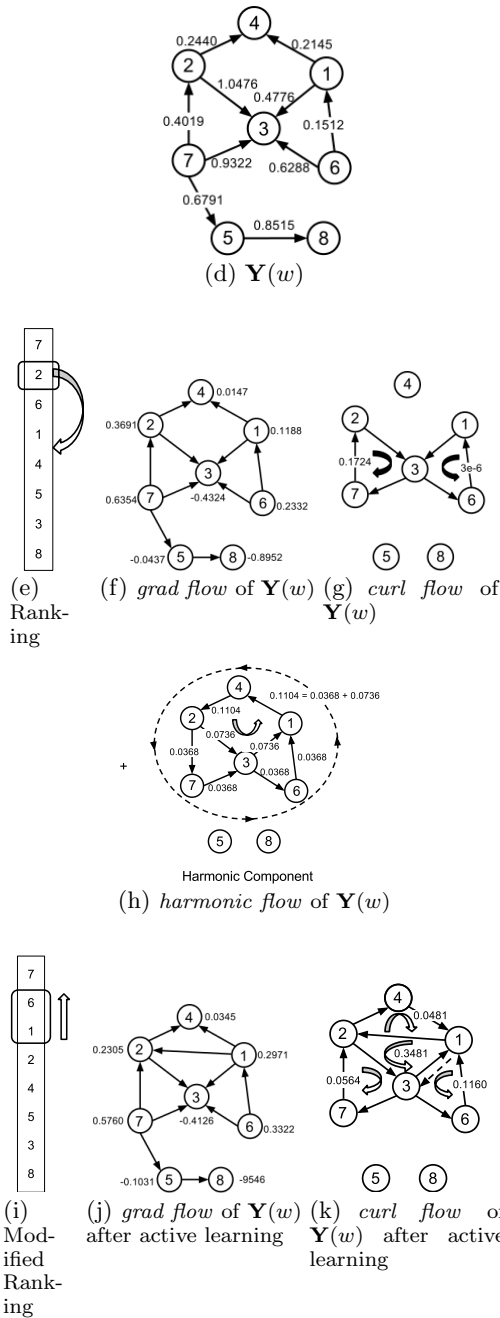
Contributions: The above examples highlight the following requirements in ranking applications. (1) Ability to fuse the preference data from multiple aspects. (2) Need to identify and resolve conflicts. Our contributions are,

- A technique for aggregating preference data from multiple aspects.
- An active exploration framework for identification of critical missing data for fast resolution of ranking conflicts.

We exploit Hodge decomposition framework for ranking of preference data and identification of critical missing data. Hodge decomposition is a global framework which requires processing of the information about *all* comments of an article. Since new comments arrive steadily, we need a mechanism to incrementally update the ranking rather than running the ranking calculations from the scratch. In this case, online updates are more efficient. Hence we propose

- A technique for online update of ranking models.

Paper organization: This paper is organized as follows. Section 1.1 discusses the problem definition and multi-objective ranking. Section 2 reviews Hodge decomposition. Section 3 outlines the proposed technique for ranking data aggregation and Section 4 describes the use of Hodge decomposition on the aggregated matrix. While Hodge decomposition provides a global ranking, the quality of this ranking may not be high. Section 5 proposes an active learning framework for increasing the quality of global ranking. Online updates for Hodge ranking are discussed in Section 6. Section 7 provides experimental validation of the proposed techniques. Section 8 places this work in context with respect to related work. The paper closes with concluding remarks.



1.1 Problem Definition

The general problem tackled in this work can be defined as follows. We are given n objects to be ranked. (The notation in the following sections is summarized in Table 1.) In this paper, these are the comments to a single article. There are k criteria for comparisons (rating, author-reputation, readability, freshness, textual similarity with the article, collaborative factors, etc.). For each comparison criterion r , we have a preference score matrix \mathbf{Y}^r with \mathbf{Y}_{ij}^r denoting the rank-score difference between comment i and comment j according to the criteria r . Note that the matrix \mathbf{Y}^r is skew-symmetric ($\mathbf{Y}_{ij}^r = -\mathbf{Y}_{ji}^r$).

As mentioned in Example 1, the entries observed or observable are sparse. The sparsity is measured by a parameter p which is the fraction of the total entries observed. And the support set for criteria r is denoted by \mathbf{S}_r which is a symmetric binary matrix. In this paper, we address the following.

Rank Aggregation: We find a ranking preference matrix $\mathbf{Y}(w)$ which respects each of the criteria \mathbf{Y}^r (in a pre-decided weighted manner). See Section 3.

Figure 1: Example: Rank aggregation for ranking a set of eight comments.

Table 1: Table of Notations

e	Vector of 1's
e_i	Unit vector with dimension i set to 1
k	Number of comparison criteria
n	Number of comments to rank
p	Sparsity ratio in preference data
\mathbf{R}	Residue matrix
s	Individual ranking-score vector
\mathbf{S}_r	Binary support matrix of \mathbf{S}_r
\mathbf{X}_G	Gradient component
\mathbf{X}_C	Curl Flow component
\mathbf{X}_H	Harmonic Flow component
\mathbf{Y} (\mathbf{Y}^r)	Preference scores (for criterion r)
$\mathbf{Y}(w)$	Aggregated \mathbf{Y}^r s using weight vector w
α_r	Preference weight for criterion r
$\ \mathbf{Y}\ _{F,S}$	Frobenius norm of \mathbf{Y} on support set \mathbf{S}

Global ranking: Once we find the “consensus” preference matrix, we find the “optimal” global ranking consistent with it (Section 4).

Active Learning: In active learning we identify the pair of comments, for which if more data is acquired, the global inconsistency maximally reduces (Section 5).

Online Update: Here we compute the ranking incrementally as in online learning framework (Section 6).

2. HODGE DECOMPOSITION

In this section, we review the combinatorial Hodge decomposition as described in [16] for arriving at globally consistent rankings from pairwise preference information \mathbf{Y} . The decomposition also provides a quality score (“residue”) for the global ranking as well as a resolution of the residue to insignificant (*curl*) and significant (*harmonic*) components.

2.1 Globally Consistent Ranking

A globally consistent ranking is characterized a single score function on comments. Suppose $s \in \mathbf{R}^n$ is the vector of ranking scores for the comments such that $s_i > s_j$ implies that comment i is better than comment j . Then preference matrix induced by s is given by

$$\mathbf{X}_{ij} = s_j - s_i \quad (1)$$

This can also be written in matrix form as

$$\mathbf{X} = es^T - se^T \quad (2)$$

The RHS can be viewed as a gradient operator on vectors s , $\text{grad} : \mathbf{R}^n \rightarrow \mathbf{R}^{n \times n}$ and the \mathbf{X} is a rank-2, skew-symmetric matrix can be viewed as a *gradient flow*.

While the matrix \mathbf{X} is cycle free, the data matrix \mathbf{Y} can have conflicts. We define the optimal global ranking \mathbf{X}_G of \mathbf{Y} as

$$\begin{aligned} & \underset{\mathbf{X}_G}{\text{minimize}} && \|\mathbf{Y} - \mathbf{X}_G\|_{F,S}^2 \\ & \text{subject to} && \mathbf{X}_G = es^T - se^T, \\ & && e^T s = 0. \end{aligned} \quad (3)$$

The objective function is the *Frobenius*-norm of the residue after subtracting a gradient flow from the data matrix \mathbf{Y} ,

weighted with \mathbf{S} . The norm is defined as

$$\|\mathbf{Y} - \mathbf{X}\|_{F,S}^2 = \sum_{\forall i,j} \mathbf{S}_{ij} (\mathbf{Y}_{ij} - \mathbf{X}_{ij})^2 \quad (4)$$

This constrained-linear least squares regression problem finds the globally consistent ranking closest to \mathbf{Y} in L_2 sense. Since “ $s + \text{constant}$ ” is also a valid solution, the second constraint is added to get a unique solution without changing the ranking order.

We can solve similar problem with minimizing L_1 -norm in the objective function or a slightly modified *nuclear-norm* minimizing, matrix completion problem as given in [10]. However, the rankings obtained by either approaches are very similar to each other. Hence, we continue with L_2 -norm for the experiments. The inconsistency in data is characterized by the residue obtained after extracting the optimal global component. The residue $\mathbf{R}^* = \mathbf{Y} - \mathbf{X}_G^*$ is a divergence-free cyclic flow. The consistency quality of observed data is characterized using the *cyclicity ratio* given by

$$C_p = \frac{\|\mathbf{R}^*\|_{2,S}^2}{\|\mathbf{Y}\|_{2,S}^2} \quad (5)$$

EXAMPLE 4. Figure 1(f) shows the gradient flow of the graph in Figure 1(d). The ranking derived from this flow is shown in Figure 1(e). \square

2.2 Inconsistencies: Local + Global

In this section, we look deeper into the inconsistencies and split them into local and global inconsistencies of the data. The local inconsistency is a *curl flow* which is combination of triangular cyclic flows. And the global inconsistency is an orthogonal (in L_2 sense) flow which is locally acyclic but globally cyclic. This flow is called the *harmonic flow*. The local and global components are separated by solving another optimization problem,

$$\begin{aligned} & \underset{\mathbf{X}_C}{\text{minimize}} && \|\mathbf{R}^* - \mathbf{X}_C\|_{F,S}^2 \\ & \text{subject to} && \mathbf{X}_C = \text{curl}^* \Phi, \end{aligned} \quad (6)$$

where the 3-dimensional tensor Φ is of size $n \times n \times n$ and satisfies

$$\begin{aligned} \Phi(i, j, k) &= \Phi(j, k, i) &= \Phi(k, i, j) \\ &= -\Phi(i, k, j) &= -\Phi(j, i, k) \\ &= -\Phi(k, j, i). \end{aligned} \quad (7)$$

The operator curl applied to a matrix Y gives a $n \times n \times n$ tensor Φ such that

$$\Phi(i, j, k) \doteq \text{curl}(\mathbf{Y})_{ijk} \doteq \mathbf{Y}_{ij} + \mathbf{Y}_{jk} + \mathbf{Y}_{ki} \quad (8)$$

and its dual operator curl^* applied to a 3-dimensional tensor Φ gives a matrix X_C such that

$$X_C(i, j) = \sum_k \Phi(i, j, k)$$

Here, the entry corresponding to an edge (i, j) is the sum of all the triangular flows which share that particular edge.

If \mathbf{Y}_{ij} is cost of traversing the edge (i, j) , then for \mathbf{X}_G any two paths between same source and destination nodes would cost exactly the same. In other words, \mathbf{X}_G satisfies the *zero-curl* condition,

The optimization problem (6) is a linear least squares regression for the matrix \mathbf{X}_C which is constrained to the space of curl^* of three-dimensional $n \times n \times n$ tensors. Where, curl^* is dual of the curl operator defined in Expression (8). The solution \mathbf{X}_C denotes the *curl flow* which comprises of the local-inconsistencies and can be written as sum of triangular inconsistencies. The residue flow $\mathbf{R}^* - \mathbf{X}_C$ denoted by \mathbf{X}_H is the divergence-free, curl-free flow called the *harmonic flow*. The matrix \mathbf{X}_H is a locally consistent but globally inconsistent flow which comprises of cycles of size ≥ 4 . The global inconsistency usually occurs due to some missing \mathbf{Y} values. In case of complete \mathbf{S} , the global inconsistency is 0 as all the large cycles are decomposed into gradient flow and smaller triangular flows.

EXAMPLE 5. Figures 1(g) and 1(h) show the curl and harmonic flows of the graph in Figure 1(d). The cycles in the harmonic flow (in Figure 1(h)) is of length 4. Recall that cycles of length 4 or more are called globally inconsistent. The curl component (Figure 1(g)) has cycles of length 3 and these are called locally inconsistent. Note that our proposed active learning technique (in Section 5) tries to eliminate global inconsistencies. \square

Similar to “cyclicity ratio” (see Expression 5), we define norm-ratios for curl and harmonic components as

$$\frac{\|\mathbf{X}_C\|_{2,\mathbf{S}}^2}{\|\mathbf{Y}\|_{2,\mathbf{S}}^2} \text{ and } \frac{\|\mathbf{X}_H\|_{2,\mathbf{S}}^2}{\|\mathbf{Y}\|_{2,\mathbf{S}}^2} \quad (9)$$

to quantify local and global inconsistencies.

Therefore, we have decomposed the aggregated preference data \mathbf{Y} into

$$\mathbf{Y} = \mathbf{X}_G + \mathbf{X}_C + \mathbf{X}_H \quad (10)$$

which is known as the **Hodge decomposition** of a matrix into (a) curl-free *gradient flow* (b) divergence-free *curl flow* and (c) curl & divergence-free *harmonic flow*.

3. RANKING DATA AGGREGATION

Hodge decomposition provides a way of decomposing a single preference matrix. As described earlier in Section 1.1, we have several preference matrices \mathbf{Y}^r for $r = 1, 2, \dots, k$ – each one being partial and possibly noisy. These are aggregated to get an incomplete matrix $\mathbf{Y}(w)$. (We extract the most appropriate individual ranking score s from $\mathbf{Y}(w)$.)

In this section, we describe the initial preprocessing step of aggregating the ranking data from various criteria. Given n objects (comments) to be ranked, there are k criteria for comparisons (rating, recency, author-reputation, readability, etc). For each comparison criteria r , preference score for every pair (i, j) may not be available or meaningful. We denote the sparse support for each criteria with a symmetric binary matrix \mathbf{S}^r with $\mathbf{S}_{ij}^r = 1$ for every available preference score (i, j) . The sparsity is measured by a parameter p denoting the fraction of total entries observed. The actual value of the difference in ranking score between a pair of comments is given by a skew-symmetric matrix \mathbf{Y}^r . In the most consistent case \mathbf{Y}^r is a rank 2 skew-symmetric matrix with e (vector of ones) as one of its generators as explained in the previous section.

We now describe our procedure to aggregate the incomplete preference data from different criteria to get a balanced

aggregate preference measure. Firstly, we observe that an edge (i, j) with $\mathbf{Y}_{ij}^r = 0$ in a preference matrix, can either have the corresponding $\mathbf{S}_{ij}^r = 0$ or $\mathbf{S}_{ij}^r = 1$. The distinction between the two being that a 0 of first kind denotes lack of information for that edge, while the second kind indicates that the two comments i and j are equal in preference scores according to the criteria r . (The latter is called a “structural zero”.) Hence, we say that an edge (i, j) belongs to the support set \mathbf{S}^r if \mathbf{Y}_{ij}^r value is known with high confidence and it denotes the preference measure between i and j .

Drawing parallel from the rank-aggregation of ranking lists [5], we combine the individual preference matrix \mathbf{Y}^r into a combination preference matrix given by

$$\mathbf{Y}(w) = \sum_{r=1}^{r=k} w_r \mathbf{Y}^r. \quad (11)$$

The weights w are learned by solving the following optimization problem:

$$\begin{aligned} & \underset{w}{\text{minimize}} && \sum_{r=1}^{r=k} \alpha_r \|\mathbf{Y}(w) - \mathbf{Y}^r\|_{F, \mathbf{S}^r}^2 \\ & \text{subject to} && \mathbf{Y}(w) = \sum_{r=1}^{r=k} w_r \mathbf{Y}^r, \\ & && e^T w = 1, \quad w \geq 0. \end{aligned} \quad (12)$$

The objective function is the weighted sum of the *Frobenius* distances of the aggregated matrix from the individual components. See Equation (4). And the parameters α_r are chosen from prior confidence or importance assigned to respective criteria. The α_r can also be tailored to specific user preferences using declared interests or learned from user behavior history.

Note that each edge (i, j) may belong to multiple or one or none of the comparison criteria support sets \mathbf{S}^r . Let $\hat{\mathbf{S}}$ denote the union of support sets \mathbf{S}^r . (Note that $\hat{\mathbf{S}}$ does not depend on w .) Thus, $\hat{\mathbf{S}}_{ij} = 1$ implies that for at least one criteria r , the preference information for edge (i, j) is available. Due to the non-uniformity of the sparsity patterns of different criteria, the weights have to be carefully normalized over the available information for each edge belonging to $\hat{\mathbf{S}}$. The solution of this constrained-convex optimization problem gives us a balanced aggregate preference measure $\mathbf{Y}(w)$.

EXAMPLE 6. Figure 1(d) shows the preference graph corresponding to $\mathbf{Y}(w)$. \square

4. HODGE RANK

In this section, we show how the ideas discussed in the previous sections fit together.

1. We start with k preference matrices: $\mathbf{Y}^1, \mathbf{Y}^2, \dots, \mathbf{Y}^k$.
2. We use rank aggregation (optimization problem (12)) to arrive at the aggregated preference matrix $\mathbf{Y}(w)$.
3. We first separate out the *gradient flow* through the optimization problem (3).
4. We extract the *curl flow* through the optimization problem (6).

The quality of decomposition is estimated by the norm of residues (see Expressions (5) and (9)) while the quality of the ranking is measured by comparing the sum of the Kendall- τ distances from individual rankings. See Section 7 for details.

EXAMPLE 7. Figure 1(f) shows the preference graph corresponding to this approximating preference matrix. We refer to this approximating preference relationship as the gradient flow of the aggregate preference matrix. The ranking obtained as a result is shown in Figure 1(e).

We now move to the analysis of the residue. One component captures local inconsistencies in the aggregate preference relationship while the other component captures global inconsistencies. Figure 1(g) shows the preference graph corresponding to the local inconsistencies, which is referred to as the curl flow. These are in the form of short cyclic preferential relationships amongst the comments indicated as triangular cyclic components in the figure. The preference graph pertaining to the global component (referred to as the harmonic flow) is shown in Figure 1(h) and exhibit preferential chains of longer cycles. In our example, the harmonic component is $\{1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 1\}$ and $\{6 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 7 \rightarrow 3 \rightarrow 6\}$. \square

5. ACTIVE LEARNING

Having obtained the Hodge decomposition into three components, we will now look at each of it closely. Ideally, with complete data and no noise, we would expect $\mathbf{Y}(w)$ to be of the form of *gradient flow* and extracting a global ranking-score s from it would be straight forward. But as most settings with real online data, the noise in behavior of large user base is inevitable. And with complete noisy data, even though the global inconsistencies are zero, the local inconsistencies are unavoidable. In this work, we do not focus on methods for reducing the local inconsistencies. In our setting, the sparsity adds another twist leading to a non-zero harmonic flow and thereby a globally inconsistent component.

Our next step is to eliminate the global inconsistencies in the harmonic component which are caused by the sparsity of data. For this, we will select edges for active-learning and use editorial or explore-exploit methods to obtain the preference data for these comment-pairs. Evaluating all the absent edges would solve our problem. However, the evaluation process is either very expensive (for editorial) or takes a long time (to increase ratings using explore-exploit methods). Hence, it is vital to choose the edges intelligently to eliminate the harmonic-component as cheaply and quickly as possible. And the experimental evidence suggests that the global inconsistencies can be reduced substantially with relatively very few edge evaluations. Among the various different methods for choosing edges for evaluation, we compare the following three methods:

1. Baseline method of adding edges randomly.
2. Edge which maximizes the number of triangles completed by its addition,
3. Edge which maximizes the weighted sum of triangles completed by its addition,

Section 7 shows the experimental results of these three strategies for reducing the harmonic component.

EXAMPLE 8. As we saw earlier in this example that there is an uncertainty in the preference relationships between comments 1 and 2. Having a direct preferential relationship between comments 1 and 2 is more reliable than to infer it through second order relationships. To resolve this, we evaluate this edge directly and introduce the correct preference explicitly which is $1 \rightarrow 2$. Figure 1(i) shows the updated ranking order after the introduction of this edge. Figure 1(j) shows the gradient flow component after the addition of the new edge and Figure 1(k) shows the updated curl flow of the updated aggregate preference matrix. \square

6. ONLINE RANK CALCULATIONS

In this section, we describe how to update the rank when new information/data arrives/is obtained in the form of a new comment or an update from active learning iteration. One would predict the existing order to not change substantially by addition of a new comment. Hence, the new comment can be expected to be inserted in the existing order with few changes in the order of “nearby” comments. In order to define the notion of “nearby” comments, we will first look at an alternate formulation for the optimization problem (3). Given the $n \times n$ preference data matrix \mathbf{Y} , we consider a directed graph on n nodes with weighted edges from node j to node i if $\mathbf{Y}_{ij} > 0$. Let \mathbf{A} denote the adjacency matrix of this graph, \mathbf{A} corresponds to the positive part of the sparse support \mathbf{S} . Let $\mathbf{D} = \mathbf{A}e$, where e , as before, is the vector of 1s. Then the Laplacian matrix of the graph is given by $\mathbf{G} = \mathbf{D} - \mathbf{A}$. Substituting the *gradient flow* constraint in the objective function of the optimization problem (3) and differentiating with respect to s , we get the following equivalence:

$$\arg \min_s \|\mathbf{Y} - \text{grad}(s)\|_{F,S} \equiv \{s : (\mathbf{D} - \mathbf{A})s = \mathbf{G}s = \beta(\mathbf{Y})\} \quad (13)$$

where β is a constant vector which linearly depends on \mathbf{Y} , in fact, β_i is the total sum of the weights of edges leaving and entering the node i with appropriate signs. Therefore,

$$\beta(\mathbf{Y}) = \mathbf{Y}e \quad (14)$$

It is known that for a connected component with n_c nodes, the corresponding block Laplacian has rank $n_c - 1$. Hence, for a graph with c connected components, the Laplacian \mathbf{G} is rank deficient with rank $n - c$. For simplicity, we assume we have a single component, and the procedure described can be extended to multiple components by repeating the same for each block. As in the previous formulation, the solution to the linear system is not unique under translation. Thus we replace the first row in \mathbf{G} with a row of all 1s to get $\bar{\mathbf{G}}$ and the corresponding entry in β with 0 to get $\bar{\beta}$. The solution to this system is unique, and for multiple components we can do the same for each by replacing first rows of each block with a row of appropriate 1s and 0s.

The key insight of our online update formulation is that the optimal ranking score s^* from a data matrix \mathbf{Y} can be computed by splitting the matrix as $\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2$, computing the individual ranking scores s_1^* and s_2^* to get the effective score $s^* = s_1^* + s_2^*$. However, in order to split the L.H.S. of the linear system, it is important to note that both the problems have to be solved with same support set \mathbf{S} even if certain entries of individual \mathbf{Y}_1 or \mathbf{Y}_2 are 0 (more specifically, structural zeros).

We will use this result to compute new ranking when a new comment is added. This adds a new node to the graph. The edges added will be based on the scores like author-reputation, readability, etc. which can be calculated instantly for the new comment. Let the new node be $n + 1$ and suppose it is connected to d of the old nodes denoted by the neighborhood set \mathcal{N}_d . The binary vector indicating this connectivity is denoted by $g \in \mathbf{R}^n$ and corresponding weights y_{new} . Then, the new system of equations is given by:

$$\begin{bmatrix} \bar{\mathbf{G}} + \mathbf{D}_g & -\bar{g} \\ -g^T & d \end{bmatrix} s_{\text{new}} = \begin{bmatrix} \bar{\beta}_{\text{old}} + \beta_{\text{inc}} \\ \beta_{n+1} \end{bmatrix} \quad (15)$$

Here, \bar{g} is g with first entry set to 1, \mathbf{D}_g is diagonal matrix of g and β_{inc} is the increment to β after appending of y_{new} to the data \mathbf{Y} .

Let $s_g = s \circ g$ denote the entry-wise (Hadamard/Schur) product of s and g , and $\bar{s}_g = \frac{g^T s}{d}$ be the mean of these d non-zero values to give $\bar{y} = s_g - \bar{s}_g g$. Now we split the new problem into two problems, both with the new support, first with the \mathbf{Y} entries for all the old edges and $s_i - \bar{s}_g$ for the new edges between $i \in \mathcal{N}_d$ and the new node. The second part with 0 for all the old edges and the remaining difference of new preference values for the new edges. Thus we write the new preference matrix as:

$$\begin{bmatrix} \mathbf{Y} & y_{\text{new}} \\ -y_{\text{new}} & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{Y} & \bar{y} \\ -\bar{y}^T & 0 \end{bmatrix}}_{\mathbf{Y}_{\text{aug}}} + \underbrace{\begin{bmatrix} \mathbf{0} & \Delta y \\ -\Delta y^T & 0 \end{bmatrix}}_{\mathbf{Y}_{\text{inc}}} \quad (16)$$

where, $\Delta y = y_{\text{new}} - \bar{y}$. The first system is very similar to the older problem with only few new edges, and the second system is sparse with a large number of 0s and a very few non-zero values.

The new system of equations for the first problem is given by:

$$\begin{bmatrix} \bar{\mathbf{G}} + \mathbf{D}_g & -\bar{g} \\ -g^T & d \end{bmatrix} s_{\text{aug}} = \begin{bmatrix} \bar{\beta}_{\text{old}} + s_g - \bar{s}_g g \\ 0 \end{bmatrix} \quad (17)$$

Here, \bar{g} is g with the first entry set to 1 to maintain the uniqueness condition of first row. The solution for this problem has a closed form solution which can be written as:

$$s_{\text{aug}}^* = \begin{bmatrix} s - \delta e \\ n\delta \end{bmatrix} \quad (18)$$

where, the scalar $\delta = \frac{\bar{s}_g}{n+1}$. Note that the form of s_{aug}^* guarantees the first condition of zero-sum to hold.

The second problem is more involved and the ease of solving it depends on the structure of the existing graph and the new edges added. The new system is given by:

$$\begin{bmatrix} \bar{\mathbf{G}} + \mathbf{D}_g & -\bar{g} \\ -g^T & d \end{bmatrix} s_{\text{inc}} = \begin{bmatrix} \beta_{\text{inc}} - s_g + \bar{s}_g g \\ \beta_{n+1} \end{bmatrix} \quad (19)$$

The L.H.S. is same as first problem and the R.H.S. is sparse with all 0s except the neighbors of the new node and the last entry. As expected, the influence of the few non-zero edges will diminish as we go further away from the new node. The final ranking score is given by sum of the two solutions

$$s_{\text{new}}^* = s_{\text{aug}}^* + s_{\text{inc}}^* \quad (20)$$

To solve the second problem in online fashion, we will maintain and update the inverse $\bar{\mathbf{G}}^{-1}$. For this, it is vital to note

that the new matrix can be written as a rank $d + 1$ update of an invertible matrix.

$$\begin{aligned} \bar{\mathbf{G}}_{\text{new}} &= \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} + \sum_{i \in \mathcal{N}_d} (e_i - e_{n+1})(e_i - e_{n+1})^T \\ &\quad + \begin{bmatrix} \mathbf{0}^T & \mathbf{1}_{1 \notin \mathcal{N}_d} \\ \mathbf{0}^T & -1 \end{bmatrix} \\ &= \bar{\mathbf{G}}_+ + \mathbf{U}\mathbf{V}^T. \end{aligned} \quad (21)$$

where, \mathbf{U} and \mathbf{V} are both $n \times (d+1)$ matrices. The first term $\bar{\mathbf{G}}_+$, which is $\bar{\mathbf{G}}$ appended with a column and row of 0s and 1 on diagonal, is easily invertible given the inverse matrix $\bar{\mathbf{G}}^{-1}$ from previous iteration. The second term is the sum of d rank-1 updates for each edge added to the graph and the third term denotes the adjustment required to $\bar{\mathbf{G}}_{\text{new}}(1, n)$ to have first row of 1s for the uniqueness condition and to $\bar{\mathbf{G}}_{\text{new}}(n, n)$ to get d . And the new matrix inverse $\bar{\mathbf{G}}_{\text{new}}^{-1}$ is given by the *Sherman-Morrison-Woodbury* identity [12] for inverse of low-rank updates as:

$$\bar{\mathbf{G}}_{\text{new}}^{-1} = \bar{\mathbf{G}}_+^{-1} - \bar{\mathbf{G}}_+^{-1} \mathbf{U} \left(\mathbf{I}_d + \mathbf{V}^T \bar{\mathbf{G}}_+^{-1} \mathbf{U} \right)^{-1} \mathbf{V}^T \bar{\mathbf{G}}_+^{-1}. \quad (22)$$

Thus to solve the second problem, we compute the updated inverse which can be used for further calculations. This computation involves inverting a small $(d+1) \times (d+1)$ matrix and involves $O(n^2 d)$ operations compared to $O(n^3)$ in any other direct method using *LU* or *QR* decompositions. However, a similar incremental update of *LU* factors can be used, but *LU* is highly dependent on ordering of nodes, in the worst case of the new comment being connected to node 1, the whole factor will have to be updated. One would argue that this updated inverse could be used directly to solve the original new system. However, the benefit of splitting the problem lies in the first split problem requiring $O(n)$ time for computing the closed form solution s_{aug}^* , and the second split problem exploiting the sparseness in R.H.S. to compute s_{inc}^* with a factor of $O(n/d)$ faster. We can further speed up the calculations at the expense of accuracy of the ranking scores ignoring small entries while solving the second problem. Since the ranking order is robust to small changes in the ranking score s_{new}^* , we would expect small error in s_{new}^* to not disturb the final ordering. And the complete problem can be resolved at regular intervals to avoid error buildup.

In the case of an edge update from active-learning described in previous section, an edge (i, j) gets added to the graph. The Laplacian matrix gets updated by a rank-1 matrix as

$$\bar{\mathbf{G}}_{\text{new}} = \bar{\mathbf{G}} + (e_i - e_j)(e_i - e_j)^T. \quad (23)$$

And the *Sherman-Morrison-Woodbury* identity reduces the inverse update to

$$\bar{\mathbf{G}}_{\text{new}}^{-1} = \bar{\mathbf{G}}^{-1} - \frac{\bar{\mathbf{G}}^{-1}(e_i - e_j)(e_i - e_j)^T \bar{\mathbf{G}}^{-1}}{(1 + (e_i - e_j)^T \bar{\mathbf{G}}^{-1}(e_i - e_j))}. \quad (24)$$

This update requires no matrix-inversion and the 3 matrix-vector multiplication can be done carefully using sparsity of $(e_i - e_j)$, reducing the complete updating procedure to $O(n^2)$ flops.

Thus we have described a method to update the hodge-rank for incremental addition of nodes or edges to the data.

7. EXPERIMENTAL RESULTS

In this section, we provide experimental validation of the proposed methods.

Dataset: We use 200 articles along with 50 comments for each article from Yahoo! News. The comments have the following associated information: time stamp, commenter, and rating (number of thumbs up and thumbs down) along with comment text. The articles appeared during Sep 2011.

Ranking aspects: We use three ranking criteria namely, rating, reputation of author, and quality of the comment. The reputation information for the commenters is the average rating obtained during the three month period June 2011 to Aug 2011. This information is not available for all commenters in the dataset. For comment quality, we use the framework of [4]. The rating information is given by the fraction of thumbs up to the total number of rating for a comment.

The above information is available in the form of absolute scores. We calculate the preference information as $(score_j - score_i)$. Note that the ratings of a old comment and new comment are not commensurate since the old comment could have accumulated a large number of ratings and the new comment may not have had that chance. So we calculate the preference information only between “commensurate” comments.

Metrics: We use Kendall τ rank correlation coefficient to characterize the distance between two rankings. Kendall τ is defined as

$$Q = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{n(n-1)}{2}}$$

τ values are in the range $[-1, 1]$ with -1 indicating reverse order and +1 indicating same order. Hence higher values indicate better performance. If R_1, R_2, \dots, R_k are the individual rankings and R_A is the aggregated rank, then we use the Q -metric defined below.

$$Q(R_A) = \frac{1}{k} \sum_{i=1}^k \tau(R_A, R_i)$$

to characterize the quality of R_A . A higher values of $Q(R_A)$ indicate better aggregation.

To show the sizes of various components (like residue, harmonic and curl), we use the three norm-ratio measures defined in Expressions (5) and (9).

Note the lack of readily available golden data in this task. It is close to impossible to editorially rank a set of comments. The best approach is to do live tests. In this paper, we use the above measures as proxies for live tests.

Baselines: For rank aggregation, we use *simple mean* and *weighted mean*. In simple mean, the scores are averaged first and the ranking derived from the scores is used. In weighted mean, the weights obtained from the optimization problem (12) are used for averaging and the ranking induced by the weighted scores is used.

7.1 Rank aggregation

We first use optimization problem (12) described in Section 3 to calculate the aggregated preference matrix and the globally consistent Hodge rank is computed using the optimization problem (3). (We set all $\alpha_r = 1$ in equation (12)). Further, two baseline rankings are computed by simple mean and weighted mean of the individual partial ranking lists of

Table 2: Q -metric from individual rankings

	Sparsity p	Hodge Rank	Simple Mean (% Gain)	Weighted Mean (% Gain)
A.m	0.3	0.4185	0.3422(-22.31)	0.3445(-21.48)
	0.4	0.4110	0.3492(-17.53)	0.3497(-17.53)
	0.5	0.4047	0.3578(-13.09)	0.3597(-12.51)
A.1	0.3	0.3810	0.3208(-19.48)	0.3124(-22.98)
	0.4	0.3907	0.3397(-15.55)	0.3318(-18.50)
	0.5	0.3911	0.3539(-10.82)	0.3489(-12.57)
A.2	0.3	0.3941	0.3501(-13.51)	0.3372(-18.34)
	0.4	0.3854	0.3512(-10.50)	0.3452(-12.39)
	0.5	0.3812	0.3554(- 7.89)	0.3544(- 8.09)
A.3	0.3	0.4180	0.3814(-10.15)	0.3846(- 9.34)
	0.4	0.4105	0.3824(- 7.63)	0.3859(- 6.49)
	0.5	0.4041	0.3814(- 6.17)	0.3867(- 4.59)
A.4	0.3	0.3816	0.3313(-16.55)	0.3213(-20.34)
	0.4	0.3858	0.3395(-14.27)	0.3368(-15.44)
	0.5	0.3838	0.3443(-12.01)	0.3418(-12.83)

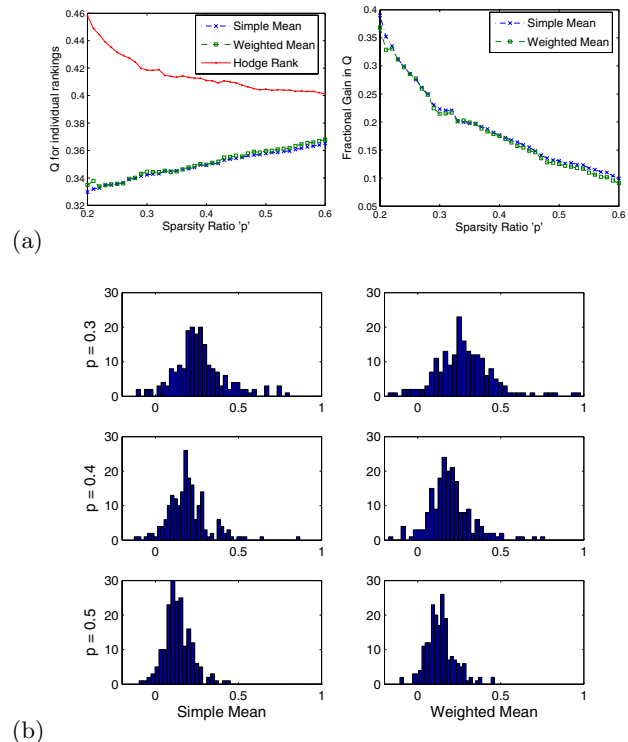


Figure 2: (a) $Kendall$ - τ from individual rankings (left) and Fractional Rise in Total $Kendall$ - τ (right). (b) Histogram for Fractional Rise for 200 articles.

each criteria. We then compare the quality of Hodge rank to the baseline ranks using the total $Kendall$ - τ of these ranks from the individual partial rankings lists.

Table 2 illustrates the typical quality, $Q(\cdot)$, values for Hodge rank, simple mean rank and weighted mean rank. This is shown for the 200 article dataset (“A.avg”) as well as for four arbitrarily chosen articles. In the latter, we average over 25 instances of randomly chosen 50 comments for each article. The table also shows the gain of the baselines compared to Hodge rank. It can be seen that Hodge rank

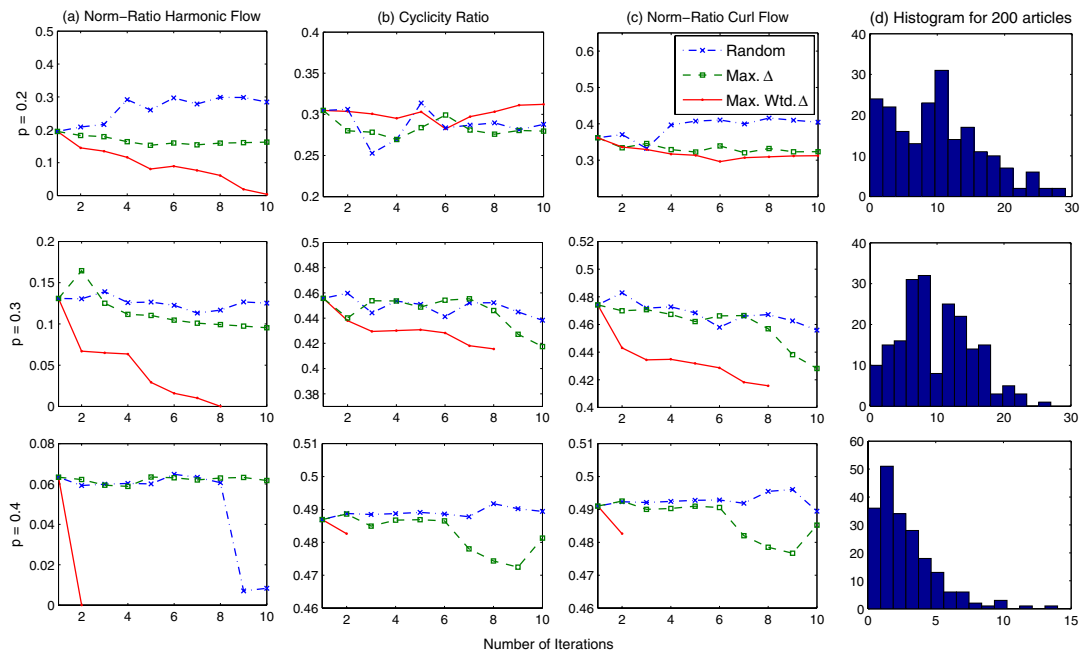


Figure 3: Active Learning. (a) Norm-Ratio Harmonic Component. (b) Cyclicity Ratio. (c) Norm-Ratio Curl Component. (d) Histogram of Number of Iterations.

is up to 30% better than the baselines. Also, the weighted average on par or slightly better than simple average but not uniformly so – as can be seen from Table 2 and Figure 2.

Figure 2 shows the Quality scores averaged over 200 articles in different formats. In Figure 2(a), we show the typical dependence of the *Kendall- τ* values of the three rankings for different sparsity p in data, ranging from $p = 0.2$ to $p = 0.6$. Figure 2(b) shows the typical fractional gain in these values for Hodge rank. As expected, as the sparsity in data reduces (p increases) the performance of simple and weighted mean ranks catch up with the Hodge Rank as seen in Figure 2(a). The histogram for the fraction gain in *Kendall- τ* from individual rankings for Hodge rank is given in Figure 2(b) for three different sparsity values. Note that for majority of the articles, Hodge rank shows increasing improvement over the mean ranks with sparsity (the histogram shifts towards 0 as p increases). However, for about 3 – 5% of the articles, Hodge Rank performs worse than the mean rankings.

7.2 Active Learning

In second part of the experiments, we solve the optimization problem (6) to separate the global and local inconsistencies. The harmonic component is then used to identify the edges for active learning to reduce the global inconsistency. Three strategies as described in Section 5 are used in parallel and at each iteration the unknown data value for the respective edge is added to the individual \mathbf{Y}^r matrix.

Figure 3(a)-(c) show the norm-ratio of the harmonic component, the cyclicity ratio and the norm-ratio of curl component² as a function of number of active learning iterations for three different values of p . (For $p = 4$, the norm ratio of the harmonic component becomes 0 in two iterations.

²See Expressions (5) and (9) for definitions.

Hence we do not show the Cyclicity ratio and the norm ratio of curl flow beyond two iterations.) As can be seen, the weighted maximum heuristic performs the best in reducing the harmonic component. At the same time, the cyclicity ratio reduces confirming that additional data improves the rating. Further decrease in norm-ratio of curl component indicates that additional data does not just transfer inconsistency from global component to local component. Figure 3(d) shows the histograms of number of active learning iterations required to eliminate the harmonic component for 200 articles. As expected, the average number of edges required reduces with sparsity (increase in p).

7.3 Online Updates

Finally, we investigate the computation times for online updates for addition of edges from active-learning iterations or addition of nodes as a result of receiving new comments. Computation times for computing the Hodge rank for 10 active-learning iterations is shown in left plot of Figure 4. The right plot of Figure 4 illustrates the computation time required for incrementally updating the Hodge rank as number of comments increase from 50 to 60, one at a time.

8. RELATED WORK

Comments and ratings form a key component of the Social web and are one of the primary contributors to its success. There is a significant volume of research effort in the recent years focused on comments and ratings. User studies in [18, 11] illustrate the importance of comments in blogging communities. In [24], Witschge explores public online debates in the form of comments on strong socio-political issues and in [21] Schuth et al. study the discussion structure in comments associated with online news articles. In [20],

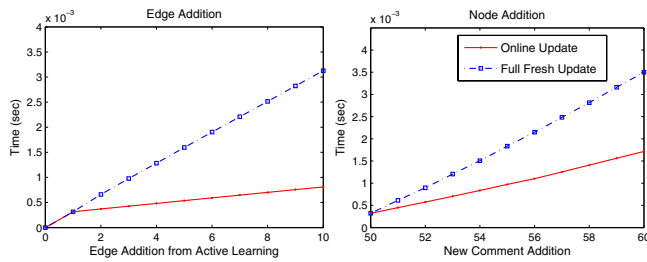


Figure 4: Computation Times for Online Updates

the authors study comments associated with blog posts and [14] utilizes the comments associated with blog posts for summarization of the blog article.

Ratings and Quality: A number of recent efforts are in the direction of understanding the quality of user-contributed comments. In [19], Mishne et. al. utilize a language modeling approach to detect *link spam* in comments made on blogs. The quality measure for comments used in our experiments is based on the work of Chen et al. [4]. In [4], Chen et al. presents a supervised approach to determine reputation of users in the context of comments and ratings. To this end they editorially judge the quality of comments according to a pre-defined set of guidelines. Utilizing this as a training set, they learn regression models to characterize the quality of a comment. The predicted reputation of a user as an author of comments is obtained as mean quality score of the comments posted by them. The authors also introduce a support-based reputation computation. This is defined as the average rating a user receives from all the users for comments made by him in a certain category. However, this is difficult to compute empirically as in practice a user receives ratings for his comments only from a small subset of users. To mitigate this issue, the authors propose a latent factor model to predict the ratings of a comment. This model is personalized in the sense that it predicts the rating that a particular user will provide to a comment made by another user in a certain context. They show that this tensor model can be marginalized easily to obtain the support based reputation score for the users in a certain context.

With the increase in user participation on commenting, ranking/recommendation of comments becomes an important problem to facilitate efficient consumption of existing comments on a web site. [22] presents a detailed study of user comments in YouTube videos. The authors examine the comments and their ratings in YouTube videos to investigate the relationships between the language and sentiment expressed and the ratings of the comments. Treating highly vs lowly rated comments as two classes, they also learn a discriminative model to predict whether a given comment is likely to be highly or lowly rated. [13] proposes a rating prediction approach to user comments on the social news aggregator site, Digg. In Digg, users can rate (provide thumbs up and thumbs down votes) to each comment (associated with a submitted story). The authors consider the difference of the total thumbs up and total thumbs down as the aggregate rating of a comment. They adopt the learning to rank approach to predict the ratings of comments. To this end they compute various features pertaining to each comment, based on the commenting activity of the commenter

(user reputation) and content of the comment. The various user and content based features adopted in this work is similar to what we use in our work. They utilize historical data of comment ratings to train support vector regression models based on these features. As the rating received by a comment is biased by the posting time of the comment (older comments get more visibility and hence more ratings), the paper also presents experimental results where they explicitly account for this bias while training the ranking model. In [1] Agarwal et al. presents a framework for recommendation of user comments so as to surface comments (in an online news service) to users (in a personalized fashion) which they are likely to rate positively. They adopt a generalized linear model framework where given a user and a comment, the mean rating for it consists of several factors like rater bias, comment popularity, author reputation. Additionally they also include factors representing author-rater affinity and rater-comment affinity. They estimate the latent factors using MLE in a Bayesian setting where they learn appropriate priors for the factors by pooling data across similar users and comments.

Multi-objective Ranking: In the recent years, the challenge of optimizing for multiple criterion in learning to rank paradigm has attracted a lot of attention in the research community [23, 6]. Dai et al. in [6] adapts the conventional learning to rank paradigm to simultaneously optimize for freshness and relevance in web search. To this end they extend the Divide and Conquer ranking framework [3]. The training phase involves clustering queries into several clusters based on the retrieval features computed from the top ranked results (according to a reference ranking model) for these queries. The clustering employed is a soft-clustering where each query is associated with all the clusters with different association weights. They learn multiple ranking models for each of these clusters where they incorporate the notion of freshness into the traditional letor approach by generating hybrid labels based on relevance and freshness judgments (similar to Dong et al. [8]). While serving results for a query, they determine the affinity (association weights) of a query to each of the clusters. Then the results are scored according to each of the pre-trained models and the scores are combined using these association weights to obtain the final ranking of the results.

Rank aggregation: Rank aggregation is a widely studied problem in the domain of social choice and voting theory [2]. In the context of the web, one of the early works on rank aggregation is seen in [9]. In [9] they illustrate the use of rank aggregation for aggregating results from multiple search engines. They rely on the Condorcet criterion for mitigating problems of spam, where their suggested approach satisfies the extended Condorcet criterion. While, Kemeny optimal aggregation satisfies this criterion, finding a kemeny optimal is known to be NP hard. The authors introduce a concept of *locally Kemeny optimal* which states that an aggregated permutation is locally Kemeny optimal if one can not decrease the Kendall τ measure (of the aggregated permutation with the component rank orders) by swapping adjacent elements in the aggregated order. In essence, their approach is to start with any rank aggregation technique and do a *local Kemenization*. The locally Kemeny optimal aggregate can be obtained by computing the Hamiltonian path in the majority graph.

In the context of rank aggregation, there have been some

efforts in the direction of semi-supervised [5] and supervised [17] rank aggregation. In [17], Liu et al. propose a framework which utilizes the ordinal information and a labeled training data to learn a rank aggregation model in a supervised setting. To this end they adapt the Markov Chain based rank aggregation approach (which is non-convex) into that of semi-definite programming. They show encouraging results on the OSHUMED dataset and also in a rank aggregation task of web search results from six different commercial search engines.

In [10], Gleich et al. adopt a matrix completion approach towards rank aggregation. They utilize the scores from the component rankers to generate pairwise preference matrices of the items to be ranked. They suggest various aggregation schemes (based on arithmetic mean, geometric mean, etc.) to generate a composite pairwise matrix from the component rankings. The final ranking is obtained as a rank-2 skew symmetric completion and approximation of the composite matrix. To this end they utilize a singular value projection based algorithm. Results on a synthetic and the Netflix dataset illustrate the efficacy of their method.

We have adopted Combinatorial Hodge Theory based approach towards rank aggregation. The application of Hodge Theory for ranking was first proposed in [15, 16]. In our rank aggregation framework, the comments are associated with *cardinal* scores based on quality, reputation and ratings. Unlike traditional rank aggregation methods which consider *ordinal* orderings based on these scores, the Hodge decomposition approach allows us to leverage these *cardinal* scores in a more direct fashion to establish the pairwise preference graph (see Section 3). Further, as discussed in Section 3 the pairwise preference graphs obtained using these individual aspects (rating, quality, and author reputation) are incomplete. The Hodge decomposition framework in essence generalizes the well-known Borda count [7] scheme to scenarios where the ranking data is incomplete. The framework, besides providing an elegant approach for rank aggregation, also provides some notion of a confidence measure on the quality of the global ranking that is obtained. This is expressed in terms of the *curl flow* and the *harmonic flow* components of the decomposition (see Section 2.2). In this work, we adopt an active learning based approach to *minimize* the *harmonic flow* component (global inconsistencies) and obtain a more globally consistent ranking.

9. CONCLUSIONS

Multi-objective rank aggregation is becoming essential in several applications. In this paper, we use the framework of Hodge decomposition for rank aggregation problems. We propose a novel technique for aggregating individual preference functions and experimentally show that the proposed approach is superior to commonly used baselines. We then formulate the problem of reducing global inconsistencies and propose techniques for identifying local observations which can maximally reduce global inconsistencies. We finally show how we can perform the decompositions in an online fashion. One of our future directions is to formulate the theory for choosing the edges optimally.

10. REFERENCES

- [1] D. Agarwal, B.-C. Chen, and B. Pang. Personalized recommendation of user comments via factor models. In *EMNLP*, July 2011.
- [2] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, 2nd edition, Sept. 1970.
- [3] J. Bian, X. Li, F. Li, Z. Zheng, and H. Zha. Ranking specialization for web search: a divide-and-conquer approach by using topical ranksvm. WWW, 2010.
- [4] B.-C. Chen, J. Guo, B. Tseng, and J. Yang. User reputation in a comment rating environment. In *SIGKDD*, 2011.
- [5] S. Chen, F. Wang, Y. Song, and C. Zhang. Semi-supervised ranking aggregation. CIKM, 2008.
- [6] N. Dai, M. Shokouhi, and B. D. Davison. Learning to rank for freshness and relevance. SIGIR, 2011.
- [7] J. C. de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1784.
- [8] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz. Towards recency ranking in web search. In *WSDM*, 2010.
- [9] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. WWW, 2001.
- [10] D. F. Gleich and L.-H. Lim. Rank aggregation via nuclear norm minimization. In *KDD*, 2011.
- [11] M. Gumbrecht. Blogs as ‘Protected Space’. In *WWW*, 2004.
- [12] W. W. Hager. Updating the inverse of a matrix. 1989.
- [13] C.-F. Hsu, E. Khabiri, and J. Caverlee. Ranking comments on the social web. In *ICCSE*, 2009.
- [14] M. Hu, A. Sun, and E. P. Lim. Comments-oriented document summarization: understanding documents with readers’ feedback. In *SIGIR*, New York, NY, USA, 2008.
- [15] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Learning to rank with combinatorial hodge theory. *CoRR abs/0811.1067*.
- [16] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Math. Program.*, March 2011.
- [17] Y.-T. Liu, T.-Y. Liu, T. Qin, Z.-M. Ma, and H. Li. Supervised rank aggregation. WWW, 2007.
- [18] E. Menchen-Trevino. Blogger motivations: Power, pull, and positive feedback. *Internet Research 6.0*, 2005.
- [19] G. Mishne. Blocking blog spam with language model disagreement. In *AIRWeb*, 2005.
- [20] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *WWW Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006.
- [21] A. Schuth, M. Marx, and M. de Rijke. Extracting the discussion structure in comments on news-articles. In *WIDM*, 2007.
- [22] S. Siersdorfer, S. Chelaru, W. Nejdl, and J. San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. WWW, 2010.
- [23] K. M. Svore, M. Volkovs, and C. J. C. Burges. Learning to rank with multiple objective functions. In *WWW*, 2011.
- [24] T. Witschge. (In)difference Online. *PhD thesis, ASCoR*, 2007.