Hodge Decomposition of Paired Comparison Flows in Click-through Data

Zhanglong JI[1], Yang AN[1], Ying CHEN[1], Yuan YAO[1], Jun XU[2], Hang LI[2]
[1]School of Mathematical Sciences, Peking University
[2]Microsoft Research - Asia, Beijing
Correspondence Email: YY <yuany@math.pku.edu.cn>

Abstract - **Hodge theory, as a powerful technique which bridges over analysis, geometry and topology, was recently introduced to data analysis in metric spaces [1] and statistical ranking [2]. When applied to ranking problem, a combinatorial version of Hodge Theory leads to an orthogonal decomposition of pairwise comparison flows into three components: one is the gradient of a global score function which can be used to rate the candidates, the other two describe the inconsistency of pairwise comparison data, locally and globally respectively. Through this decomposition, one not only obtains a global ranking score, but also gets two measurements on how consistent are the raters in pairwise comparisons.**

Pairwise comparison models for user's clicks have been made popular due to Joachims' work [3]: the user's clicks, can derive implicit pairwise comparisons between the clicked and the skipped webpages, which is shown to reflect user's preference over documents and the relevance of query vs. document. In this report, we study an application of combinatorial Hodge theory to the pairwise comparison models for click data based on Joachims' work. Our purpose is to investigate both the global ranking of the relevant documents and the nature of inconsistency within user's preference. Our studies show that it is not a proper way to directly apply Joachim's models on typical industrial data collected from a commercial search engine, which tends to revert the order of search engines and thus leaves a larger deviation from Human Rating Scores even than random ranking. Most of the inconsistency in user's preference is of local scale, consisting of those webpages targeted by different interest groups. Such a result suggests to explore novel models to exploit click data for web search ranking.

Key Word: **Hodge Theory, Search Ranking, Pairwise Ranking**

I. INTRODUCTION

The central task of commercial search engines is to rank the documents according to the relevance to the queries issued by users. There have been lots of algorithms based on features of the document and between documents and queries. But these algorithms are supervised learning algorithms, and they need Human Rating Scores, which are both expensive and sparse. To exploit the information in clicked data which are cheap and of large amount, Jochaims [3-5] suggests pairwise comparison models for implicit feedbacks from user's click with a ranking SVM. This starts the effort to mine information in the user behavior.

One fact that must be considered when building a model based on clicks is position bias, put forward by Jochiams [4], which says that the users tend to click the documents that are ranked higher. This bias is not related to the quality of the documents, because this phenomenon persists in another experiment which puts the document in inverse order [4]. The eye-tracking experiment in the same paper shows that the user's eyes mainly focus on the first documents, which can explain why those documents are clicked more often.

The position bias is the main concern and difficulty in search ranking with click data, but it also hints that users may examine the document from the top to the bottom. Based on this assumption, there is high probability that if a user skips some documents to clicks on the document ranked lower, he in fact conducts a comparison among these documents. With different interpretation of users' preference, several models of pairwise comparisons can be made, and the application of Hodge Theory can be hosted here.

Hodge Theory was developed by W.V.D. Hodge as an extension of the theory of the Laplacian on domains of Euclidean space or on a manifold. It established some beautiful connections between topology, geometry and analysis through Laplacian operators. Recently Hodge theory has been studied in the setting of data analysis in metric spaces [1] and in particular, statistical ranking [2] which enables a decomposition of pairwise comparison matrices into three orthogonal components. The first one is the gradient of a function, which can be regarded as the score of each candidate to be ranked. The other two parts characterize the inconsistency of users' behavior in a local and global way, respectively.

In this paper we report a preliminary study of applying Hodge Theory to Joachim's pairwise comparison models for click data, with data from industrial search engines. The experimental result shows that first of all, Joachim's pairwise comparison models, although successful in his specific setting, are not consistent to human rating scores (HRS) in our cases, even further from that than random ranking. A discussion will be given at the end of the paper. Second the global inconsistency is generally close to zero, which indicates the absence of large global cycles in data of high frequency. Third, the majority of inconsistency is due to local or triangular inconsistency, which is not affected by queries' frequency or number of documents related. The statistical stability of Hodge decomposition is also shown through Bootstrap

experiments. These results motivate novel models for click data in search ranking.

The whole paper is organized as follows: Section 2 will introduce some background of web search. Basic notations, definitions, concepts and theorems of combinatorial Hodge Theory will be introduced in Section 3. Section 4 will give a detailed algebraic way to decompose a pairwise comparison matrix. In Section 5, the pairwise comparison models by Jochaims are formulated and a matrix representation of their decompositions is given. In Section 6, we will briefly compute the complexity of our model. The experimental result will be in Section 7, and a discussion will be in Section 8 with future directions.

II. Web Search

When a user issues a query, for example, *google*, search engine searches the whole web, ranks the documents (or urls) by their relevance to the query, and return the most relevant ones to the user. This is called web search. Search companies do label some query-document pairs manually. Supervised learning technique can thus be exploited to learn to rank based on labels and features for query-document pairs. Rank learning algorithms can be roughly classified into two categories: pair-wise [8-9] and list-wise [6-7] where both classification and regression schemes are found helpful. But this approach is too expensive and inefficient: every rating must be paid, and there are hundreds of millions of queries a search engine receives everyday and billions of web pages on the whole web. To address this challenge, many algorithms have been suggested.

As implicit feedback from the users are both informative, cheap and in huge amount, and computers have gained the ability to analyze them, researchers have turned to click log for help. Joachims [5] first utilized implicit feedback in web search and webpage ranking, he chose pair-wise comparison method to train a RankSVM. RankSVM converts the ranking problem into a classification problem based on the comparison between document pairs. This algorithm uses the features of documents as predictors and preference shown by clicks as responses, then solves the problem under the framework of SVM.

However, in such a work we are not clear that how consistently the click data can be used to approximate the users' preference behavior. So learning from preference models derived from clicks might be misleading. This motivates us to explore this issue by Hodge theory, which can measure whether user behavior on some queries is consistent.

III. Combinatorial Hodge Theory

Hodge theory, which is a powerful technique in geometry and topology, was developed by W.V.D. Hodge in the 1930s as an extension of the theory of the Laplacian on domains of Euclidean space or on a manifold. In this paper, we will pay close attention to its combinatorial version in k-dimensional simplicial complex and its relevance in ranking [2]. Although simplicial complex of dimension higher than 2 does not make any contribution to our study so far, it is promising that higher-dimensional simplicial complices would play an important role in our future studies.

A. Definitions to Claim

Definition 3.1 *A **simplicial complex** $K=(V, \Sigma)$ is a vertex set $V=\{1, \ldots, n\}$ together with a collection Σ of subsets of V that is closed in the sense of inclusion. The elements in Σ are called **simplices**. For $k \le n$, a **k-simplex** is a $(k+1)$-element subset of V and we use Σ_k to denote all k-simplices in Σ.*

Definition 3.2 *Given any undirected graph $G=(V, E)$, one obtains a $(k-1)$-dimensional simplicial complex $K_G^k=(V, \Sigma_{k-1})$ called the **k-clique complex** of G by setting $\Sigma=\{j\text{-clique of } G \mid j= 1,\ldots, k\}$. The k-clique complex of G where k is maximal is just called the* **clique complex** *of G and denoted K_G.*

Definition 3.3 *Let K be a simplicial complex and recall that Σ_k denotes the set of all k-simplices in Σ. A k-dimensional **cochain** is a function $f : \Sigma_k \to R$ that is alternating on each of the k-simplex, i.e.*

$$f(i_{\sigma(0)}, \ldots, i_{\sigma(k)}) = sign(\sigma) f(i_0, \ldots i_k)$$

for all $\{i_0, \ldots, i_k\} \in \Sigma_k$ and all σ from the permutation group on $k+1$ elements. The set of all k-cochains on K is denoted $C^k(K,R)$.

For simplicity, we will often use C^k instead when there is no cause for confusion. We note that the k-cochain space C^k can be given a choice of inner product $< , >_k$.

Definition 3.4 *The kth **coboundary** operator $\delta_k : C^k(K,R) \to C^{k+1}(K,R)$ is the linear map that takes a k-cochain f to a $(k+1)$-cochain $\delta_k f$ defined by*

$$(\delta_k f)(i_0, i_1, \ldots, i_{k+1}) = \sum_{j=0}^{k+1} (-1)^j f(i_0, \ldots, i_{j-1}, i_{j+1}, \ldots, i_{k+1})$$

for all $\{i_0, i_1, \ldots, i_{k+1}\} \in \Sigma_{k+1}$ and all $f \in C^k$.

We may construct the formal adjoint operator of the coboundary map, $\delta_k^*: C^{k+1}(K,R) \to C^k(K,R)$ by

$$< \delta_k f_k, g_{k+1} >_{k+1} = < f_k, \delta_k^* g_{k+1} >_k$$

where $f_k \in C^k$ and $g_{k+1} \in C^{k+1}$.

Definition 3.5 *The **combinatorial divergence** operator $div : C^1(K,R) \to C^0(K,R)$ is the adjoint of δ_0, i.e.*

$$\mathrm{div} = -\delta_0^*$$

For convenience, we may usually drop the adjective 'combinatorial' from 'combinatorial divergence' when there is no risk of confusion.

Definition 3.6 *Let K be a simplicial complex. The k-dimensional* ***combinatorial Laplacian*** *is the operator Δ_k : $C^k(K,R) \to C^k(K,R)$ defined by*

$$\Delta_k = \delta_k^* \delta_k + \delta_{k-1}\delta_{k-1}^*$$

B. Theorems to Apply

The main theorem in combinatorial Hodge theory is the Hodge decomposition theorem, which holds in general for any simplicial complex and for any dimension k. We will mainly concern ourselves with studying a special case k=1, also called Helmholtz decomposition theorem. In this section, we will state these two theorems in detail and one may refer to [2] for a basic proof to the Hodge decomposition theorem.

Theorem 3.7 (Closedness). $\delta_{k+1}\delta_k = 0$

Theorem 3.8 (Hodge Decomposition Theorem). *$C^k(K,R)$ has an orthogonal decomposition*

$$\mathrm{C}^k(\mathrm{K},\mathrm{R}) = \mathrm{im}(\delta_{k-1}) \oplus \ker(\Delta_k) \oplus \mathrm{im}(\delta_k^*)$$

Furthermore,

$$\ker(\Delta_k) = \ker(\delta_k) \cap \ker(\delta_{k-1}^*)$$

Theorem 3.9 (Helmholtz Decomposition Theorem). *Let G be an undirected, unweighted graph and recall that K_G denotes its clique complex. $C^1(K_G, R)$ admits an orthogonal decomposition*

$$\mathrm{C}^1(\mathrm{K_G},\mathrm{R}) = \mathrm{im}(\delta_0) \oplus \ker(\Delta_1) \oplus \mathrm{im}(\delta_1^*)$$

Furthermore,

$$\ker(\Delta_1) = \ker(\delta_1) \cap \ker(\delta_0^*)$$

IV. Application of Helmholtz Decomposition Theorem in Rank Learning

A. Pairwise Rank Learning Problem

Let $\mathrm{V} = \{1,2,\dots,\mathrm{n}\}$ be a set of alternatives to be ranked and $\Lambda = \{1,2,\dots,\mathrm{m}\}$ be a set of voters. We will first define the weight function $\omega: \Lambda \times \mathrm{V} \times \mathrm{V} \to [0,\infty)$as the indicator function

$$\omega_{ij}^{\alpha} = \omega(\alpha,i,j)$$

$$= \begin{cases} 1 & \text{if voter } \alpha \text{ made a paiwise comparison between } \{i,j\}. \\ 0 & \text{otherwise.} \end{cases}$$

For each voter $\alpha \in \Lambda$, we use the pairwise ranking matrix $\mathrm{Y}^{\alpha} \in \mathrm{R}^{n*n}$, satisfying

$$\mathrm{Y}_{ij}^{\alpha} = -\mathrm{Y}_{ji}^{\alpha} > 0$$

to quantify the degree of preference of the *i*th alternative over the *j*th alternative of voter α. In this paper, we only consider the case that

$$\mathrm{Y}_{ij}^{\alpha} = \begin{cases} 1 & \text{if voter } \alpha \text{ prefers } i \text{ to } j \\ -1 & \text{if voter } \alpha \text{ prefers } j \text{ to } i \\ 0 & \text{otherwise} \end{cases}$$

Thus, we can define the skew- symmetric original comparison matrix C^* to be

$$\mathrm{c}_{ij}^* = \sum\nolimits_{\propto} \omega_{ij}^{\alpha} \mathrm{Y}_{ij}^{\alpha}$$

And the final comparison matrix C by

$$c_{ij} = f\left(c_{ij}^*, \omega_{ij}\right),$$

where $\omega_{ij} = \sum_{\alpha} \omega_{ij}^{\alpha}$ and f indicates a function with c_{ij}^* and ω_{ij} to be its variables. The detailed method to construct such pairwise comparison matrices C^* and C from raw data will be performed in Section 5B.

We can also present the comparison with a graph. Let G=(V,E) be an undirected graph with vertex set V and edge set

$$\mathrm{E} = \{\{i,j\} \in \binom{\mathrm{V}}{2} | \omega_{ij} > 0\},$$

where $\binom{\mathrm{V}}{k}$ stands for set of all k-element subset of V. Furthermore, we can also associate weights on the edges of G as capacity ω_{ij}. Such G is called a pairwise comparison graph.

We can then define edge flows on G, i.e. a function X: $\mathrm{V} \times \mathrm{V} \to \mathrm{R}$ that satisfies

$$\begin{cases} \mathrm{X}(i,j) = -\mathrm{X}(j,i) & \{i,j\} \in \mathrm{E}, \\ \mathrm{X}(i,j) = 0 & \text{otherwise.} \end{cases}$$

In particular, an edge flow of the form $\mathrm{X}_{ij} = \mathrm{s}_j - \mathrm{s}_i$ is called a gradient flow, for it can be regarded as the gradient of a function $\mathrm{s}: \mathrm{V} \to \mathrm{R}$, which will be called a potential function. A collection of all gradient flows is defined as the model class and denoted μ_{G}. A potential function is also called a score function or utility function, representing the scores of each alternative in V. From s, we can obtain the global ranking of the elements in V, on the basis of the rule that $i \gtrsim j$ iff $\mathrm{s}_i > \mathrm{s}_j$, where the former operator "$\gtrsim$" shows that alternative i is preferred to j.

Generally, our goal for rank learning problem is to minimize a sum-of-squares loss function over a model class μ_{G}, which can be then rewritten as a weighted l_2-minimization on a pairwise comparison graph, as is shown below:

$$(1) \qquad \min_{\mathrm{X} \in \mu_{\mathrm{G}}} \sum_{\{i,j\} \in E} \left(X_{ij} - c_{ij}\right)^2.$$

Besides getting global rankings from raw data, we also need a statistical measurement to quantify the consistency or reliability of the pairwise rankings.

Similar to edge flow, we define the triangular flow on G, i.e. a function $\Phi: V \times V \times V \to R$ that satisfies that an odd permutation of the arguments of Φ changes its sign while an even permutation does not. Let

$$T(E) = \{\{i,j,k\} \in \binom{V}{3} | \{i,j\},\{j,k\},\{k,i\} \in E\}$$

be a collection of triangles with every edge in E. Then we can define the curl operator that maps edge flows to triangular flows by

$$(\text{curl } X)(i,j,k) = \begin{cases} X_{ij} + X_{jk} + X_{ki} & \text{if } \{i,j,k\} \in T(E), \\ 0 & \text{otherwise.} \end{cases}$$

To measure the triangular inconsistencies, we will finally define that an edge flow is called globally consistent if it is a gradient flow and locally consistent if it is curl-free on every triangular in T(E).

B. Relevance of Hodge Theory in rank learning

In our data ranking problem, it suffices to consider cases k=0, 1, 2. Let G=(V,E) be a pairwise comparison graph and we will give special attention to a combinatorial object of the form (V, E, T(E)).

According to definition 3.1 and definition 3.2 above, (V, E, T(E)) is called a 2-dimensional simplicial complex or the 3-clique complex of G, denoted K_G^3, and $\Sigma_0 = V$, $\Sigma_1 = E$, $\Sigma_2 = T(E)$, $\Sigma = V \cup E \cup T(E)$.

Potential functions (score/utility functions), edge flows (pairwise rankings), triangular flows(triplewise rankings), gradient(global ranking), curl(local inconsistency) are all special instances of those calculus on a simplicial complex, introduced in definition 3.3 to definition 3.5.

We point out that C^0 is the space of potential functions (score/utility functions), C^1 is the space of edge flows(pairwise rankings), and C^2 is the space of triangular flows(triplewise rankings) as special cases of cochains.

For coboundary maps, we have, in particular, δ_0=grad, i.e.

$$(\delta_0 s)(i,j) = s_j - s_i$$

where $s: V \to R$, and δ_1=curl, i.e.

$$(\delta_1 X)(i,j,k) = X_{ij} + X_{jk} + X_{ki}$$

where X stands for edge flows on G.

In conclusion, the relationships between combinatorial gradient, curl, and divergence are given by

$$(\text{grad } s)(i,j) = (\delta_0 s)(i,j) = s_j - s_i$$
$$(\text{curl } X)(i,j,k) = (\delta_1 X)(i,j,k) = X_{ij} + X_{jk} + X_{ki}$$
$$(\text{div } X)(i) = -(\delta_0^* X)(i) = \sum_{\{i,j\} \in E} X_{ij}$$

with respect to the inner products on both C^0 and C^1 to be the unweighted Euclidean inner products,

$$< r, s >= \sum_{i=1}^{n} r_i s_i$$

$$< X, Y > = \sum_{\{i,j\} \in E} X_{ij} Y_{ij}$$

for all $r, s \in C^0$ and all $X, Y \in C^1$.

To better understand the application of Helmholtz decomposition theorem in rank learning, we need to clarify the ranking theoretic interpretations of each subspace in the theorem.

(1) $\text{im}(\delta_0) = \text{im}(\text{grad})$ denotes the subspace of pairwise rankings that are globally consistent or acyclic.
(2) $\ker(\delta_1) = \ker(\text{curl})$ denotes the subspace of curl-free pairwise rankings and they are precisely locally consistent. According to the Closedness theorem, we have im(grad) is a subset of ker(curl), and its orthogonal complement in ker(curl) is $\ker(\Delta_1)$ discussed below.
(3) $\ker(\delta_0^*) = \ker(\text{div})$ denotes the subspace of divergence-free pairwise rankings, i.e. for each alternative $i \in V$, whose total out-flow equals its total in-flow. So such rankings may be considered to be inconsistent or cyclic.
(4) $\ker(\Delta_1) = \ker(\delta_1) \cap \ker(\delta_0^*)$ denotes the subspace of pairwise rankings that are both curl-free and divergence-free. Thus this subspace comprises only locally but not globally consistent pairwise rankings.
(5) $\text{im}(\delta_1^*) = \text{im}(\text{curl}^*)$ denotes the subspace of locally cyclic pairwise rankings. By the Closedness theorem, im(curl*) is a subspace of ker(div), and the orthogonal complement of im(curl*) in ker(div) is $\ker(\Delta_1)$ discussed above.

C. Application of Helmholtz Decomposition Theorem

In order to discuss the solutions and residuals of our optimization problem (1) in the Hodge theoretic framework, we need to restate the inner products of C^0mentioned in Section 4B.

The optimization problem (1) is then equivalent to the following equation

$$\min_{s \in C^0} \|\delta_0 s - C\|^2 = \min_{s \in C^0} < \delta_0 s - C, \delta_0 s - C >,$$

which is an l_2- projection of an edge flow representing a pairwise ranking onto im(grad) or $\text{im}(\delta_0)$, for the Helmholtz decomposition assures that the three subspaces $\text{im}(\delta_0)$, $\ker(\Delta_1)$ and $\text{im}(\delta_1^*)$ are orthogonal with respect to the inner products on C^0 and C^1.

Then the condition for a stationary point gives the normal equation

$$\delta_0^* \delta_0 s = \delta_0^* C.$$

By substituting $\Delta_0 = \delta_0^* \delta_0$ and $\text{div} = -\delta_0^*$, we get the following theorem.

Theorem 4.1 *Solutions of (8) satisfy the following normal equation*

$$\Delta_0 s = -\mathrm{div}\, C.$$

The minimum norm solution is then given by

$$s^* = -\Delta_0^{+} \mathrm{div}\, C,$$

where + indicates a Moore-Penrose inverse.

Furthermore, Hodge theory provides us with information about the l_2-norm of the least squares residual, which represents the validity of the global ranking of s*. If the residual is small, then the global ordering obtained is expected to be a majority consensus and we may conclude that s* gives a reasonably reliable ranking of the alternatives. On the other hand, if the residual is large, we may see that it is hard to assign any reliable ranking to it and it may need to be labeled manually.

Theorem 4.2 *The residual* $R^* = C - \delta_0 s^*$ *is divergence-free, i.e.* $div\ R^* = 0$. *Moreover, it has a further orthogonal decomposition*

$$R^* = \mathrm{proj}_{\mathrm{im}(\mathrm{curl}^*)} C + \mathrm{proj}_{\ker(\Delta_1)} C,$$

where $\mathrm{proj}_{\mathrm{im}(\mathrm{curl}^*)} C$ *is a local cyclic ranking accounting for local inconsistencies and* $\mathrm{proj}_{\ker(\Delta_1)} C$ *is a harmonic ranking accounting for global inconsistencies. In particular, the first projection is given by*

$$\mathrm{proj}_{\mathrm{im}(\mathrm{curl}^*)} C = \mathrm{curl}^*(\mathrm{curlcurl}^*)^{+} \mathrm{curl}(C).$$

V. MODELS

A. Pair-wise comparison of documents

First of all, information of comparisons of documents should be extracted from search engine's log file. How to get them given logs of users' clicks on pages? Since the users examine the documents from the top to the bottom of the search result block according to our hypothesis, it is reasonable to assume that users' clicks represent their preference for the clicked documents to the others as follows: here we use five methods on the pair-wise comparison from[14]: clicked $\tilde{\succ}$ skipped above, clicked $\tilde{\succ}$ former clicked, clicked $\tilde{\succ}$ skipped above + next, last click $\tilde{\succ}$ the previous and clicked $\tilde{\succ}$ the unclicked previous.

a) Clicked $\tilde{\succ}$ skipped above

This model is mainly based on the fact that users tend to read urls from top to bottom, according to the result of an eye-tracking study mentioned before. This means that if the *i*-th document is clicked, then it is preferred by the user to all the documents presented before it (from the first to the (*i*-1)-th document). There is neither comparison between this document and the documents after it, or comparison between it and the clicked documents before it.

b) Clicked $\tilde{\succ}$ former clicked

This means that if the *i*-th document is clicked, then it is better than the clicked documents before it. This is because users continue searching when they are not satisfied with the former clicked documents, and only more relevant documents can attract their attentions. There is no comparison between the clicked and non-clicked documents, and among the non-clicked documents, there are no comparison, either.

c) Clicked $\tilde{\succ}$ skipped above + next

This means that except the comparison mentioned in the first method, the clicked document is also better than the document next to it. This is because people make the decision to click the result after evaluating not only the higher-ranked results, but the result right after it as well.

d) Last click $\tilde{\succ}$ the previous

This means that comparison only happens between the last click and the document before it. When this method is applied, each page provides at most one comparison, so the comparison is very sparse compared with the former methods.

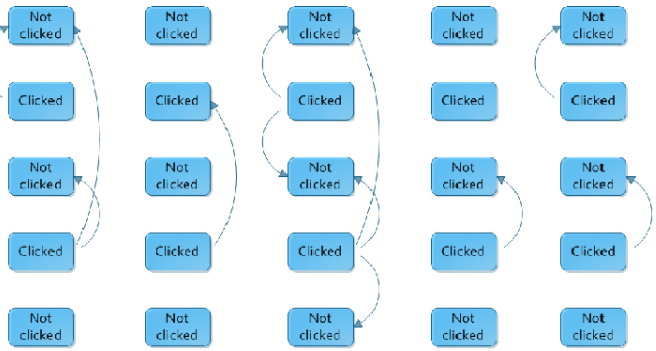

Figure 1 Ways to construct pairwise comparison model

e) Clicked $\tilde{\succ}$ unclicked previous

This is also a method which produces very sparse comparison matrix. It believes that comparison occurs only between the clicked document and the document before it.

The operator "$\tilde{\succ}$" indicates that the left one is preferred by the user according to the amount of useful information on this page. Figure 1 shows those five methods of pairwise comparisons introduced above on a single page. This simplified page has five documents in a row, and the second and the fourth are clicked, while the others are not. We use arrows to present the preference: if an arrow is from document d_1 to document d_2, then d_1 is thought to be superior to d_2.

B. Form of Comparison Matrix

After the extraction of information on users' preference from a large number of log files, the next step is to use them to form an original comparison matrix $C^* = \{c_{ij}^*\}_{i,j=1,2,\ldots,n}$. In this matrix, c_{ij}^* denotes the times that document *i* is preferred to document *j* minus the times document *j* is preferred to document *i*. When calculating c_{ij}^*, we need to go though all log files and once there is a page indicating users' preference for document *i* to document *j* based on one of the five methods discussed above, c_{ij}^* should be added by 1 and c_{ji}^* should be added by -1.

If all the documents presented are regarded as vertices, and two vertices are connected when they are compared at least once, all the documents and their connections will form a graph. Only those components having connections with many other vertices should be paid attention to, and vertices that have no edges should be left out. In our experiment, we only study the largest connected subgraph of the whole graph. And for most situations, this subgraph and the original graph is the same.

After this dealing with all log file, an original comparison matrix C^* will be obtained. The number of its rows and columns should be the same as the number of documents shown in the logs. Then some modifications will be done to form the final comparison matrix $\mathrm{C} = \{\mathrm{c}_{ij}\}_{i,j=1,2,\ldots,\mathrm{n}}$ and there are three methods to accomplish such modification: Thurstone-Mosteller Model, Uniform Model and Bradley-Terry Model.

Before introducing these three models in detail, It should be pointed out that all these models are built on the probability $\mathrm{P}(i > j)$, and c_{ij} should be a monotone increasing function of $\mathrm{P}(i > j)$. Here $\mathrm{P}(i > j)$ is defined as $\frac{1}{2}\left(\frac{\mathrm{c}_{ij}^*}{\omega_{ij}} + 1\right)$. From the large number theorem, this is reasonable when *i* and *j* are compared many times. For the first and third model below, a smoother can be used to avoid $\mathrm{P}(i > j) = 0$ or 1.

(1) Thurstone-Mosteller Model

$c_{ij} = erfinv(P(i > j) - 1/2)$, and erfinv is the inverse of error function. This model is on the assumption that the score of every document to a single user obeys independent normal distributions with the same variance.

(2) Uniform Model

$c_{ij} = P(i > j) - P(j > i)$, and this is a simple and intuitive model.

(3) Bradley-Terry Model

$c_{ij} = \log(P(i > j)) - \log(P(j > i))$, and ε is a smoother.

All these three models fit the requirement that c_{ij} is a monotone increasing function of $\mathrm{P}(i > j)$. Besides, they are all even functions, making the comparison matrix C a skew-symmetric matrix.

C. Decomposition of Comparison Matrix

In Section 3 and Section 4, we take a deep look at Hodge theory and its relevance in rank learning problem basically from the linear algebra view. In this section, we will rely on a comparatively simple matrix view of Hodge theory to find a practical way to solve our problem, at the sacrifice of losing some important geometric insights.

After the comparison matrix C is obtained, it can be decomposed into three orthogonal components as described in Section 3C.

$$C = C_1 + C_2 + C_3$$

while C_1 corresponds to the $\mathrm{im}(\delta_0)$, C_2 corresponds to the $\ker(\Delta_1)$, and C_3 the $\mathrm{im}(\delta_1^*)$. The decomposition is restricted to the positions where *i* and *j* have been compared.

Since the three subspaces are orthogonal with respect to C^0 and C^1 inner products, the decomposition can simply be completed by projecting the matrix C into the three subspaces. Then a mathematical solution to this decomposition comes:

To calculate the first part, we should find a vector $\mathrm{S} = \{s_i\}_{i=1,2,\ldots,n}$ satisfying

$$\mathrm{S} = \mathrm{argmin}\|\delta_0 S - C\|^2$$

Or represented by matrix view:

$$\mathrm{S} = \mathrm{argmin}\sum(s_i - s_j - c_{ij})^2$$

Let $|V|$ be the number of documents related to the matrix, or vertices in the subgraph, $|E|$ be the number of compared pairs of documents, or edges in the graph. D_0 is a matrix with $|E|$ rows and $|V|$ columns, each row corresponds to an edge and each column to a vertex. The entry of D_0 at the *i*-th row and *j*-th column is 0 if the *i*-th edge doesn't have connection with the *j*-th vertex, and as for the two vertices that the *i*-th edge connects, the one with smaller index, let's call it j_1 for convenience, the j_1-th entry in this row is 1, and the other is -1. w is a sparse matrix of C, i.e. a vector whose length is $|E|$. Each entry of w corresponds to an edge: If the edge is between *i*-th vertex and *j*-th vertex, and $i < j$, then $w = c_{ij}$. With these parameters, S is the solution of the following formula:

$$\mathrm{S} = \mathrm{argmin}(D_0\mathrm{S} - w)^2$$

Since S is the coefficients of projection of w on the space spanned by columns of D_0, S is also the solution of

$$\mathrm{S} = \mathrm{argmin}(D_0' D_0\mathrm{S} - D_0' w)^2$$

Further,

$$D_0' D_0\mathrm{S} = D_0' w$$

After calculating the *Moore-Penrose inverse* (or Pseudo-inverse, or Generalized inverse) of $D_0' D_0$, the value of S is got, and so is C_1:

$$c_{ij}^1 = s_j - s_i$$

Thus $C_2 + C_3 = C - C_1$, and C_3 can be calculated with the similar method. Since C_3 is the projection of C on the space $\mathrm{im}(\delta_1^*)$. There hold:

$$C_3 = \delta_1^* T,$$

where T is a vector with a length of $|Tri|$ satisfying

$$\mathrm{T} = \mathrm{argmin}\|\delta_1^* T - C\|^2$$

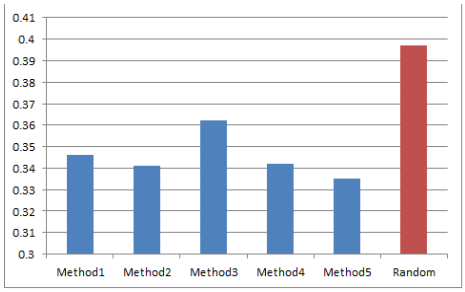

Figure 2 NDCG of five pair-wise comparison models in Section 5A, compared with that of the random model

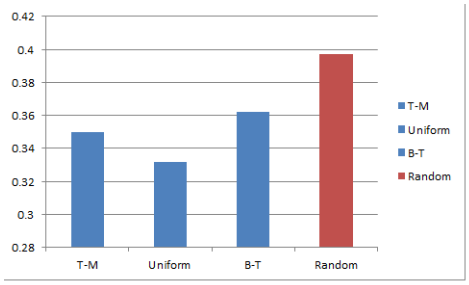

Figure 3 NDCG of the three models and the random model

Then we have metrical representation:

$$\mathrm{T} = \operatorname{argmin}(D_1 T - w)^2 = \operatorname{argmin}(D_1' D_1 T - w_1)^2$$

And $w_1 = D_1' w$.

Tri is the set of all triangles (a set of three vertices among which each pair is connected), and D_1 is a matrix with $|\mathrm{V}|$ rows and $|\mathrm{Tri}|$ columns. If a triangle j consists of three edges $i_1 < i_2 < i_3$, then the j-th column of D_1 has only three non-zero entries, 1 at the i_1-th and i_2-th positions respectively, and -1 at the i_3-th. The j-th entry of w_1 is $w_{i_1} + w_{i_2} - w_{i_3}$. Applying the Moore-Penrose inverse again, we can get the value of T.

If the edge between the i-th and j-th vertices responds to the k-th edge, then

$$C_{ij}^3 = (D_1 T)_k.$$

Thus, the value of C_3 is got, and so is C_2:

$$\mathrm{C}_2 = \mathrm{C} - \mathrm{C}_1 - \mathrm{C}_3.$$

VI. COMPUTATIONAL COMPLEXITY

The whole computational complexity is low compared with algorithms such as RankSVM. If there are n pages, m queries, and at most l documents for each query, the time complexity is at most $(n + ml^3)$, and memory required is at most $\mathrm{O}(ml^2)$. The analysis below will give details.

A. Complexity in Forming the Comparison Matrices

When the original comparison matrices are calculated, all the data are only dealt once and the time required for dealing one page doesn't grow with the size of data. So the time require for this step is no longer than $\mathrm{O}(n)$. As for the memory, they only need to remember a comparison matrix for each query, and the matrices have no more than l columns or rows. Thus the step requires memory less than $\mathrm{O}(ml^2)$.

When the original matrices are converted into final ones, there is no additional memory required, and the time required is just $\mathrm{O}(ml^2)$ (each pair in a matrix requires one operation).

B. Complexity in Decomposing the Matrices

Here the decomposition of matrix of each query is independent, so we can simply calculate the complexity of each of them and add them up. The time required by each matrix is $\mathrm{O}(l^3)$, which is the result of calculating pseudo-inverse matrix. This step requires $\mathrm{O}(l^2)$ space to conduct the operation. In total it needs $\mathrm{O}(ml^3)$ time and $\mathrm{O}(l^2)$ space.

After these two steps, to output the result requires $\mathrm{O}(ml)$ time. Then we come to the conclusion that the time required is no more than $\mathrm{O}(n + ml^3)$ and memory requires $\mathrm{O}(ml^2)$. Compared with the iterative algorithms such as RankSVM, it is very competitive.

VII. EXPERIMENT

A. Dataset

Our data come from the log of a search engine. Only the data from the US and whose query is both frequent and has Human Rating Scores (HRS) have been selected. There are about 300 queries and about 1,800,000 pages. They are grouped according to their frequencies as follows:

Frequency	Query	Pages
<5000	182	450735
5000~10000	65	448363
>10000	47	884237
a Total	294	1783335

and according to the total number of documents compared:

Documents	Queries
<25	71
25~40	81
>40	142
Total	294

B. Criteria

NDCG is a criterion to judge whether a kind of ranking is similar to another [13], where in this paper we compare all the methods with HRS. As the first document is of the most

importance and NDCG@1 is used as a common criterion in measuring ranking of web search, we are using it to measure all kinds of models.

C. Baseline

In this experiment, we use the random ranking as a baseline.

D. Result

a) NDCG of five models in the Section 5A (using Bradley Terry Model): From the Figure 2, the third model works best, and we will use it for the rest of our experiment. Surprisingly, even this model cannot compete with random ranking in terms of fitting human rating scores! For this phenomenon, we will try to give a discussion at the end of this paper. Moreover, we note that the frequency and number of documents related to a query does not affect the result significantly. So it is unnecessary to draw their influence here.

b) Comparison of the three models in Section 5B: Figure 3 shows that even the best model, Bradley-Terry Model, is worse than random ranking in consistency with human rating scores.

E. Inconsistency

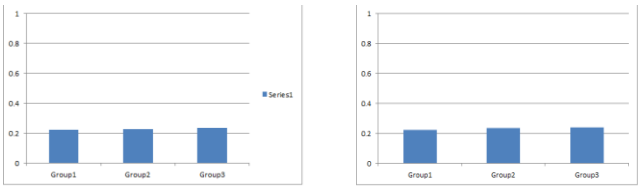

Figure 4&5 Total inconsistency of different groups

As mentioned at the beginning of this paper, the other two parts of Hodge decomposition represent whether the users' behavior is consistent. Large inconsistency means that users have different views on whether one document is more relevant than another, and total inconsistency consists of the local and the global inconsistency.

After examining all the queries, we found that only one of them has a global inconsistency larger than 0. So in the analysis followed, we will only consider local inconsistency measured by triangular curls.

The experimental results in Figure 4 and 5 show that the inconsistency doesn't change with frequency or number of documents of a query at group level.

F. Statistical Stability

Here the query citizen bank is used as a sample to measure the statistical stability of ranking and inconsistency. There are about 30,000 pages in the dataset and bootstrap method is used here. We conducted resampling 1000 times and record the model's performance on the resampled sets.

a) Number of Documents in the Largest Connected Subgraph

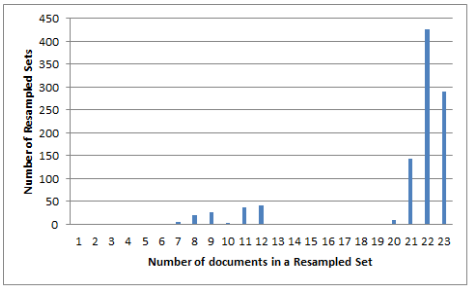

Figure 6 Number of documents in a resampled set

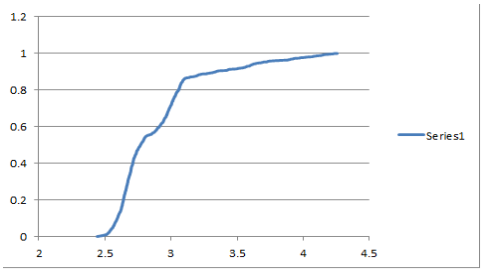

Figure 7 the Cumulated Distribution Function curve

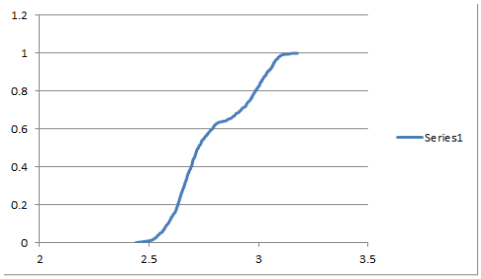

Figure 8 the new CDF curve after removing some sets

In the original dataset, there are 23 documents in the largest component, and Figure 6 is the distribution of numbers of documents in the result of resampled sets. It proves that for most instances (86%), the number of documents in the largest connected component is stable and approximates the largest size.

b) Ranking

Since we are using NDCG@1 to measure the performance of our model, which is only affected by the document ranked best, we also use the highest ranked document to measure the stability of ranking here.

Surprisingly, in the 1000 resampled dataset, the same document is always the highest. This means that our ranking algorithm is very stable. But the distribution of the score of this document is very strange (see the cumulated distribution function in Figure 7). It has a heavy tail, which is partly due to the absence of some documents. If we can get rid of those resampled sets whose largest connected subgraph is very small, the new CDF curve is as the Figure 8. The tail in this graph is much lighter.

c) Inconsistency

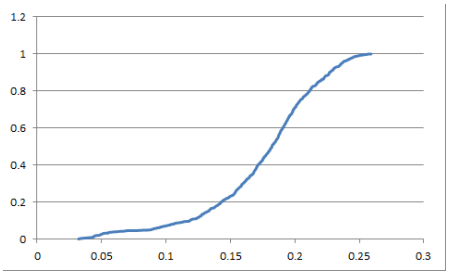

Figure 9 Stability of inconsistency

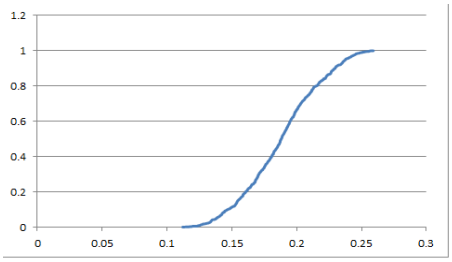

Figure 10 Stability of inconsistency after removing some resampled sets

Inconsistency is also very stable here, which has a standard deviation of 0.045 (its mean is 0.174). Yet it has a very heavy tail in the left: the 95% confidential interval is about [0.05, 0.243] (Figure 9). If we get rid of all the resampled sets that have a small largest connected subgraph, the tail will be much lighter (Figure10). After the filtering of resampled set, the mean is 0.188, the standard deviation of inconsistencies is 0.03 and the interval is [0.132, 0.244].

VIII. Future Work

Above we conducted experiments based on pairwise comparison models by Joachims, whose deviation from HRS is even larger than random rank in a sharp contrast to their original proposal. Why? A closer inspection reveals that the pairwise comparison models above tend to revert the orders of search engine, i.e. move up rapidly the documents that are ranked lower by search engine. In these pairwise comparison models, except for a small number of situations, only the documents ranked lower can be thought *preferred* by the user to the documents ranked higher. The direct consequence of this comparison is that the order of the documents is reversed.

To improve its performance, we may modify the weights in Section 4A. Although it is hard to say a click at the second position means that the second is more preferred to the third, these is a probability that the second is better. If we can consider this fact, and admit that the comparison between a former clicked and latter non-clicked may happen, the model probably work better. We leave this aspect to further pursuit.

References

[1] Laurent Bartholdi, Thomas Schick, Nat Smale, Steve Smale, and Anthony W. Baker, "Hodge theory on metric spaces," arXiv:0912.0284v1 [math.KT], 2009.

[2] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye, "Statistical Ranking and Combinatorial Hodge theory,"-Mathematical Programming, preprint. arXiv:0811.1067v2 [stat.ML], 2010.

[3] Joachims, T. Optimizing search engines using clickthrough data. In SIGKDD 2002.

[4] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, And G. Gay, Accurately interpreting clickthrough data as implicit feedback, ACMTOIS, 25(2), 2007

[5] T. Joachims, L. Granka, Bing Pan, H. Hembrooke, F. Radlinski, G. Gay, Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search, ACM Transactions on Information Systems (TOIS), Vol. 25, No. 2 (April), 2007.

[6] Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. Query-level loss functions for information retrieval. IP&M, 2006

[7] C.J.C. Burges, R. Ragno and Q.V. Le. Learning to Rank with Non-Smooth Cost Functions. Advances in Neural Information Processing Systems, 2006.

[8] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research, 4:933–969, 2003.

[9] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds,M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. ICML 2005.

[10] D. Cossock and T. Zhang. Subset ranking using regression. COLT, 2006.

[11] P. Li, C. Burges, and Q. Wu. Mcrank: Learning to rank using multiple classifications and gradient boosting. NIPS, 2007.

[12] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to Rank using Gradient Descent, in Proceedings of the International Conference on Machine Learning, 2005

[13] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 20(4), 422-446 (2002), 2002.