Cohomology and Hodge Theory

Maneet Mahal

December 1 2017

Abstract

This is an expository paper on Hodge Theory and Cohomology and its applications based on "Hodge Laplacians on Graphs" paper by Lek-Heng Lim. This paper examines a ranking method called HodgeRank, that analyzes pairwise rankings represented as edge flows on a graph using discrete or Combinatorial Hodge Theory.

Introduction

The problem of ranking in various contexts has become increasingly important in machine learning. A fundamental problem here is to globally rank a set of alternatives based on scores given by users. In all these ranking problems, graph structures naturally arise from pairwise comparisons. The incompleteness and imbalance of the data are then presented in the (edge) sparsity structure and (vertex) degree distribution of pairwise comparison graphs. The dataset used in this paper is a subset of ratings given by Amazon users to products such as games/toys. The output is a list of products ranked from top to bottom based on the algorithm called "HodgeRanking". In section 1, we discuss the basics of a Cohomology group represented in the form of a quotient vector space as discussed in Lim's paper. We also describe a harmonic representative of this cohomology class. This is followed by a discussion on a form of decomposition of a matrix known as Fredholm Alternative. Next, in section 2 we discuss the basics of Cohomology through graph theory as discussed in Hatcher's book "Algebraic Topology". We start by an example of a system of trails on a mountain as Hatcher stated in his book. We map this to a discussion on co-chain complexes followed by coboundary operators that ultimately help in a thorough understanding of the Hodge Decomposition which is discussed in the next section - section 3. In section 4 we describe the algorithm and methodology of HodgeRanking and its application. We finally conclude the discussion with our observations and results after implementing the code based on Hodge Theory. I would like to thank Kelly Spendlove for guiding me through this project. I would also like to thank the DRP team at Rutgers for giving me this opportunity to participate in the program.

1 Cohomology Group

We have a linear map that is represented as:

$$\mathbb{R}^p \xrightarrow{B} \mathbb{R}^n \xrightarrow{A} \mathbb{R}^m$$

where we have two matrices A $\in \mathbb{R}^{m*n}$ and B $\in \mathbb{R}^{n*p}$ satisfying the property:

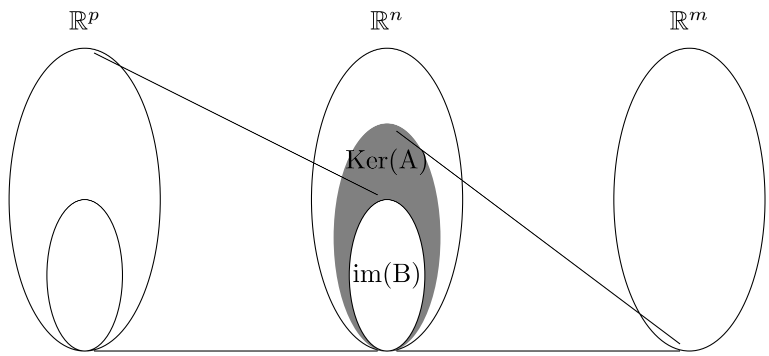
$$AB = 0$$

This means that every element in the image of B is also in kernel of A. This is equivalent to saying:

$$im(B) \subset ker(A)$$

The cohomology group with respect to A and B is the quotient vector space

$$ker(A)/im(B)$$



The shaded region in the diagram above represents the cohomology group with respect to A and B. This group is abelian under addition, and its elements are called cohomology classes. Lek-Heng Lim has described various other ways to represent this cohomology group in his paper:

Let $x \in ker(A)$. Then pick a harmonic representative x+H $\in (x + im(B))$ such that $x + H$ is orthogonal to every vector in im(B).

Here, "x + im(B)" is the cohomology class. So, x+H $\in im(B)^{\perp}$

But we know

$$im(B)^{\perp} = ker(B^{*}) \tag{1}$$

Proof. Let $x \in im(B)^{\perp}$. *Then* $\forall y \in \mathbb{R}^p$ $\langle x, By\rangle = \langle B^{*}x, y\rangle$= 0 Therefore $B^{*}x = 0$ *and so* $im(B)^{\perp}\subset ker(B^{*})$. Let $x \in ker(B^{*})$. Then $B^{*}x = 0$. *So* $\forall$ $y\in \mathbb{R}^p$ $\langle B^{*}x, y\rangle = \langle x, By\rangle$=0 $\implies$ x $\in im(B)^{\perp}$. So $ker(B^{*}) \subset im(B)^{\perp}$. Therefore, $im(B)^{\perp} = ker(B^{*})$. ☐

Therefore, from (1) and (2), x+H $\in ker(B^{*})$

Here, B^{*} represents B^{T} since we are working over $\mathbb{R}$ with the standard L^2 inner product on our spaces. This means that we pick x+H $\in ker(A) \cap ker(B^{*})$

Theorem 1.

$$ker(A^{*}A + BB^{*}) = ker(A) \cap ker(B^{*})$$

Proof. Let $x \in ker(A) \cap ker(B^{*})$. Then $Ax = 0$ *and* $B^{*}x = 0$. So $(A^{*}A + BB^{*})x = A^{*}Ax + BB^{*}x = A^{*}(0) + B(0) = 0$. So, $x \in ker(A^{*}A + BB^{*})$. $\Rightarrow$

$ker(A) \cap ker(B^*) \subset ker(A^*A + BB^*)$. Now let $x \in ker(A^*A + BB^*)$. Then $(A^*A + BB^*)x = 0. \Rightarrow A^*Ax = -BB^*x$. Multiplying by A on the left side of both sides of the equation, we get: $AA^*Ax = -ABB^*x = 0$ since $AB = 0$.
$\Rightarrow A^*Ax \in ker(A) = ker(A^*A)$ (proof in * below).
Thus $A^*Ax \in ker(A^*A). \Rightarrow A^*Ax = 0 \Rightarrow x \in ker(A^*A) = ker(A) \Rightarrow x \in ker(A)$. Similarly, if we multiply our equation $A^*Ax = -BB^*x$ by B^* on the left side of both sides, we get $x \in ker(B^*)$ and therefore, $x \in ker(A) \cap ker(B^*)$.
Thus $ker(A^*A + BB^*) \subset ker(A) \cap ker(B^*)$.
It clearly follows that

$$ker(A^*A + BB^*) = ker(A) \cap ker(B^*) \tag{2}$$

□

Proof. *Proof for $ker(A) = ker(A^*A)$: If $x \in ker(A) \Rightarrow Ax = 0$. So $A^*Ax = A^*(0) = 0$. So $x \in ker(A^*A) \Rightarrow ker(A) \subset ker(A^*A)$.
If $x \in ker(A^*A) \Rightarrow A^*Ax = 0. \Rightarrow Ax \in ker(A^*)$. It is also true that $Ax \in im(A)$. Therefore, $Ax \in ker(A^*) \cap im(A)$. But, $ker(A^*) \cap im(A) = 0$ (since $\mathbb{R}^m = ker(A^*) \oplus im(A)$ by decomposition of $\mathbb{R}^m$ by Fredholm Alternative). This means $Ax = 0$. So $x \in ker(A) \Rightarrow ker(A^*A) \subset ker(A)$.
Therefore, $ker(A) = ker(A^*A)$. □

The harmonic representative, $x + H$, is a solution to the Laplace equation

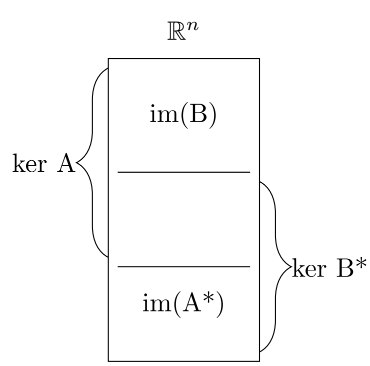
$$(A^*A + BB^*)x = 0$$

By Fredholm Alternative, the decomposition associated with the four fundamental subspaces of a matrix A $\in \mathbb{R}^{m*n}$ we have:

$$\mathbb{R}^n = ker(A) \oplus im(A^*) \tag{3}$$

$$\mathbb{R}^n = im(B) \oplus ker(B^*) \tag{4}$$

This also means that $ker(A) \cap im(A^*) = 0$ and $ker(B^*) \cap im(B) = 0$



As we can see from the diagram above, $\mathbb{R}^n$ can be decomposed based on Fredholm Alternative, and we have:

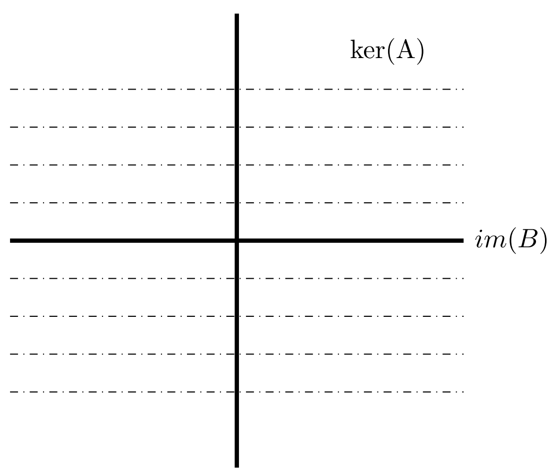
$$ker(A) \cap ker(B^*) = ker(A^*A + BB^*)$$

This represents our cohomology group, $ker(A)/im(B)$.
Also, it is clear from the representation that $im(B) \subset ker(A)$ and $im(A^*) \subset ker(B^*))$.
Therefore decomposition of $\mathbb{R}^n$ can be represented as:

$$\mathbb{R}^n = im(A^*) \oplus ker(A^*A + BB^*) \oplus im(B) \tag{5}$$

Proof for equation (6): From equation (4), we have $\mathbb{R}^n = im(A^*) \oplus ker(A)$
$= im(A^*) \oplus (ker(A) \cap im(B)) \oplus (ker(A) \cap (ker(B^*)))$ (based on the fact that we can write a set A as $(A \cap B) \cup (A \cap B^c)$ and from (5), we can see that $im(B) \cap ker(B^*) = 0$). Therefore, $\mathbb{R}^n = im(A^*) \oplus im(B) \oplus ker(A^*A + BB^*)$ (since $ker(A) \cap im(B) = im(B)$ and $ker(A) \cap ker(B^*) = ker(A^*A + BB^*)$ from equation (3)). $\square$

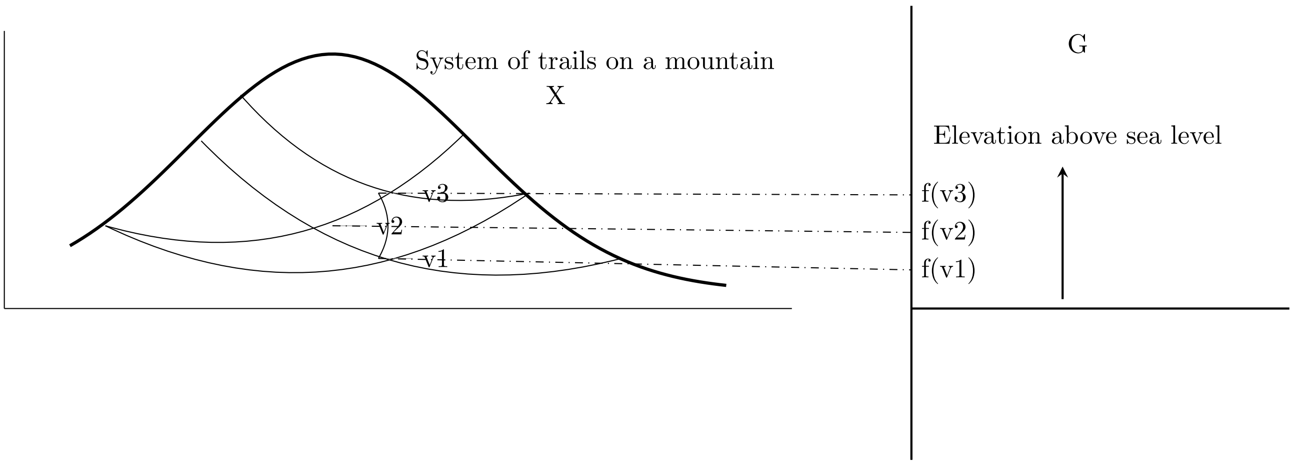
Another way to look at our cohomology group is as follows:



In the representation above, each coset of im(B), drawn in the form of horizontal line is a cohomology group (our quotient group). We pick our harmonic representative $x + H$ wherever the cosets of im(B) intersect on y axis. This is the representative we had picked in our discussion earlier and it is clearly orthogonal to every vector in im(B).

2 The Idea of Cohomology through Graph Theory

Hatcher states in his book "Algebraic Topology", a simple case of a system of trails on a mountain. Upon visualizing this, we get a mapping between junctions on trails of the mountain (vertices) and functions that represent the elevation of those points above sea level. Here, a junction refers to a point of intersection between two trails.



In the above example, we define f to be a function from X to G that maps a junction to a point that represents its elevation above sea level. We also define δf to be the function from X to G that measures the net elevation change along the trail from one junction to next. Here, X represents a 1-dimensional Δ-complex, i.e. an oriented graph and G is an abelian group. The set of all functions from X to G is denoted by $\Delta^0(X;G)$ and is also an abelian group. Therefore, $f \in \Delta^0(X;G)$. Similarly, the set of all functions assigning an element of G to each edge of X forms a group $\Delta^1(X;G)$ which is also abelian. In this case, $\delta f \in \Delta^1(X;G)$.

Then, the map $\delta : \Delta^0(X;G) \to \Delta^1(X;G)$ is a homomorphism whose value on an oriented edge $[v_0, v_1]$ is the difference $\delta(v_1) - \delta(v_0)$. The cohomology groups are $H^0(X;G)$ and $H^1(X;G)$ where $H^0(X;G) = \ker \delta \subset \Delta^0(X;G)$ and $H^1(X;G) = \Delta^1(X;G)$ / Im δ. If we move up a dimension, taking X to be a 2-dimensional Δ-complex, we can define $\Delta^2(X;G)$ to be the set of functions from the 2-simplices of X to G. Then, the map $\delta : \Delta^1(X;G) \to \Delta^2(X;G)$ is a homomorphism defined by $\delta\delta f([v_0, v_1, v_2]) = \delta f([v_0, v_1]) + \delta f([v_1, v_2]) + \delta f([v_0, v_2])$.

The two homomorphisms $\Delta^0(X;G) \xrightarrow{\delta} \Delta^1(X;G) \xrightarrow{\delta} \Delta^2(X;G)$ form a chain complex. For $f \in \Delta^0(X;G)$, we have $\delta\delta f = (f(v_1) - f(v_0)) + (f(v_2) - f(v_1)) -$

$(f(v_2) - f(v_0)) = 0$.

For G = (V,E), an undirected graph (where V = (1,....,n) is a finite set of vertices and E $\subseteq \binom{V}{2}$ is the set of edges) the set of k-cliques $K_k(G) \subseteq \binom{V}{k}$ is defined by $(i_1, ...i_{k)} \in K_k(G)$ iff $(i_p, i_q) \in E \ \forall \ 1 \leq p < q \leq k$. In our example above, the vertices v1, v2 and v3 form a 3-clique, i.e. a triangle shape. Evidently, the set comprising of all cliques of a graph G is a simplicial complex and is called the clique complex of graph G.

Now given a graph G = (V,E), we define real valued functions on its vertices $f : V \to \mathbb{R}$ and alternating functions on E (the set of edges or 2-cliques) and T (the set of triangles or 3-cliques) respectively as follows:

$\chi : V \times V \to \mathbb{R}$ such that:

$$\begin{cases} \chi(i,j) = -\chi(j,i) & \forall (i,j) \in E \\ \chi(i,j) = 0 & (i,j) \notin E \end{cases}$$

$\phi : V \times V \times V \to \mathbb{R}$ such that:

$$\begin{cases} \phi(i,j,k) = \phi(j,k,i) = \phi(k,i,j) = -\phi(j,i,k) = -\phi(i,k,j) = -\phi(k,j,i) & \forall (i,j,k) \in T \\ \phi(i,j,k) = 0 & (i,j,k) \notin T \end{cases}$$

An alternating function described above is one where permutation of its arguments has the effect of multiplying by the sign of the permutation. Referring back to the "trails on a mountain" example from Hatcher's book, the functions $f, \chi \ \phi$ represent 0-, 1- and 2- cochains respectively.

Coboundary Operators

The gradient is the linear operator defined as follows:
$grad : L^2(V) \to L^2(E)$ such that:

$$\begin{cases} (grad\ f)(i,j) = f(j) - f(i) & \forall (i,j) \in E \\ (grad\ f)(i,j) = 0 & (i,j) \notin E \end{cases}$$

The is the linear operator defined as follows:
$curl : L^2(E) \to L^2(T)$ such that:

$$\begin{cases} (curl\ \chi)(i,j,k) = \chi(i,j) + \chi(j,k) + \chi(k,i) & \forall (i,j,k) \in T \\ (curl\ \chi)(i,j,k) = 0 & (i,j,k) \notin T \end{cases}$$

The divergence is the linear operator defined as follows:
$div : L^2(E) \to L^2(V)$ such that:

$$(div\ \chi)(i) = \textstyle\sum_{j=1}^{n} \frac{w_{ij}}{w_i} \chi(i,j) \ \forall i \in V$$

Here, the weights w_i, w_{ij} are any positive values invariant under arbitrary permutation of indices. Since we are looking at the standard l^2 inner product space, we can assume w_{ij} and w_i to be of constant value 1.

Therefore, we can look at our mapping as follows, where the gradient and curl are the coboundary operators:

$$L^2(V) \underset{div}{\overset{grad}{\rightleftarrows}} L^2(E) \overset{curl}{\rightarrow} L^2(T)$$

The operator $\Delta_0 : L^2(V) \to L^2(V)$ is the graph Laplacian defined by $\Delta_0 = -div\ grad$ and operator $\Delta_1 : L^2(E) \to L^2(E)$ is the graph Helmholtzian defined by $\Delta_1 = -grad\ div + curl^*\ curl$. The graph Laplacian and Helmholtzian are special cases of Hodge Laplacians. Moreover, from our previous setting of matrices, setting A = curl and B = grad gives us curl grad = 0. We can also see that $grad^*\ grad : L^2(V) \to L^2(V)$ and $curl^*\ curl : L^2(E) \to L^2(E)$, which are vertex Laplacian and edge Laplacian respectively.

3 Hodge Decomposition

The Hodge decomposition of $L^2(V)$ is given by:

$$L^2(V) = ker(\Delta_0) \oplus im(div)$$

This can be derived from the fact that $L^2(V) = ker(grad) \oplus\ im(div)$ by Fredholm alternative and that $ker(\Delta_0) = ker(grad)$ (since $\Delta_0 = grad^* grad$ and we proved earlier that $ker(grad^* grad) = ker(grad)$).

The Helmholtz decomposition of $L^2(E)$ is given by:

$$L^2(E) = im(curl^*) \oplus ker(\Delta_1) \oplus im(grad) \tag{6}$$

The H^0 Cohomology:

The Hodge Laplacians assemble into a morphism of cochain complexes, as depicted below:

$$\begin{tikzcd}
0 \arrow[r, "0"] & Ker(\Delta_0) \arrow[r, "0"] \arrow[d] & Ker(\Delta_1) \arrow[r, "0"] \arrow[d] & Ker(\Delta_2) \arrow[r] \arrow[d] & \dots \\
0 \arrow[r, "0"] & L^2(V) \arrow[r, "grad"] \arrow[d, "\Delta_0"] & L^2(E) \arrow[r, "curl"] \arrow[d, "\Delta_1"] & L^2(T) \arrow[r] \arrow[d, "\Delta_2"] & \dots \\
0 \arrow[r, "0"] & L^2(V) \arrow[r, "grad"] & L^2(E) \arrow[r, "curl"] & L^2(T) \arrow[r] & \dots
\end{tikzcd}$$

In our example above, the H^0 Cohomology is the quotient vector space $Ker(grad)/im(0)$ which is just $Ker(grad)$. The dimension of H^0 is the number of the connected components. The H^1 Cohomology is the quotient vector space $Ker(curl)/im(grad)$.

The following two examples explain the concept of dimension of kernel, i.e. dim of $H^k(G)$ for kth order:
Let G be a graph with two vertices v1 and v2 that are not connected by any edge as shown below:

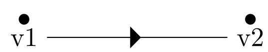
$$0 \rightarrow L^2(V) \underset{div}{\overset{grad}{\rightleftarrows}} L^2(E) \tag{7}$$

This can be represented as:

$$0 \rightarrow < \lambda_1 f1, \lambda_2 f2 > \rightarrow 0$$

Clearly, G has a two-dimensional kernel since $\Delta_0 = -div * grad = 0$.

Now we have another graph, say G' in which the vertices are connected by an edge 'e' with edgeflow as shown below:



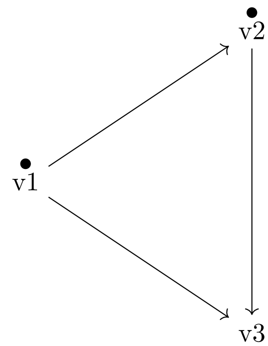
Then the equation (8) can be represented as:

$$0 \rightarrow < f1, f2 > \rightarrow < g >$$

Here, g represents the map $g : e \rightarrow 1$ such that g is the basis for the vector space $\chi : E \rightarrow \mathbb{R}$. Now, in order to find the kernel of the map from $L^2(V) \rightarrow L^2(E)$, we define $f1 : v1 \rightarrow 1; v2 \rightarrow 0$ and $f2 : v1 \rightarrow 0; v2 \rightarrow 1$, where f1 and f2 are functions from $V \rightarrow \mathbb{R}$. Then, $grad(f1+f2)(e) = (f1+f2)(v2) - (f1+f2)(v1) = 1 - 1 = 0$. Therefore, (f1 + f2) is in the kernel, and thus, by Rank Nullity, the dimension of kernel is 1 (which just consists of this element).

Vertex-Edge Incidence Matrix of a Graph

This matrix is used to calculate the eigenvalues and eigenvectors of graph Laplacians and graph Helmholtzians. The following graph of a single 3-clique is a very basic example:



Given an undirected graph as above, we label its vertices and edges arbitrarily and assign arbitrary directions to the edges. Then we write down a matrix whose columns are indexed by the vertices and rows are indexed by the edges. The (i, j)th entry of this matrix is +1 if jth edge points to the ith vertex, -1 if jth edge points out of the ith vertex and 0 otherwise. This matrix is the gradient operator, grad (known as vertex-edge incidence matrix of the graph). The edges are labeled as: e(v1,v2) = a, e(v2,v3) = b and e(v1, v3) = c. The matrix we get as a result is as follows:

$$M = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} v1 & v2 & v3 \end{array} \\ \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix} \end{array}$$

$$Graph\ Laplacian : M^*M = \begin{array}{c} \\ v1 \\ v2 \\ v3 \end{array} \begin{array}{c} \begin{array}{ccc} v1 & v2 & v3 \end{array} \\ \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \end{array}$$

The eigenvalues of the above matrix are: 3, 0

To calculate the graph Helmholtzian (the curl), we assume clockwise orientation, and we assign +1 if the edge is in the same orientation, otherwise -1. Going this way, we get the following matrix:

$$N = T \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} \end{array}$$

$$Graph\ Helmholtzian : MM^* + N^*N = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \end{array}$$

4 Algorithm and Methodology

Pairwise Ranking

Let V be a set of alternatives that need to be ranked and $E \subseteq \binom{V}{2}$ is the set of pairs of alternatives that need to be compared. Then $\chi(i,j)$ is the degree of preference of an alternative i over j, i.e. the amount that i is favored over j, and therefore, $\chi(j,i) = -\chi(i,j)$ is the amount that j is disfavored over i. In this case, the resulting edge flow $\chi \in L^2(E)$ is the pairwise ranking. Equation (7) gives us the Helmholtz Decomposition. This can be represented as:
$\chi = grad\ f +$ (x+H) $+ curl^*(\phi)$
This basically means: "pairwise ranking = consistent + globally inconsistent + locally inconsistent", where inconsistency refers to situations in which there is a circular preference such as $i > j > k > i$. The local inconsistent component $curl\phi$ measures the inconsistencies among items ranked close together while the global inconsistent component xH measures inconsistencies among items ranked far apart.

HodgeRank

In order to apply HodgeRank to a dataset, we need the data in form of pairwise comparisons, that is, each user would have compared some pairs of alternatives. These could either be ordinal pairwise comparisons (A is preferred to B), or cardinal pairwise comparisons (A is preferred to B by some value). The technique of "HodgeRank" is to rank data that may be incomplete or imbalanced. In HodgeRank, the vertices of the graph in question represent the alternatives to be ranked. A user's preferences are quantified and aggregated in the form of an edge flow. An orthogonal decomposition of the edge flow based on Hodge Theory is derived. This decomposition, called the Hodge or Helmholtz decomposition, consists of three components: a gradient flow that is globally acyclic, a harmonic flow that is locally acyclic but globally cyclic, and a curl flow that is locally cyclic. The gradient flow component induces a global ranking of the alternatives which can be easily computed via a linear least squares problem. The l2-norm of the least squares residual, which represents the contribution from the sum of the remaining curl flow and harmonic flow components, quantifies the validity of the global ranking induced by the gradient flow component. If this residual is small, then the gradient flow represents most of the variation in the underlying data and therefore the global ranking obtained from it is expected to be a majority consensus. However, if the residual is large, then the underlying data is has cyclic inconsistencies and one may not assign any reasonable global ranking to it. The model that is used reduces rank aggregation to a linear least squares regression.
Let $X : V \times V \to R$ be a pairwise ranking edge flow on a pairwise comparison graph $G = (V, E)$. X is called "consistent" on $i, j, k \in T(E)$ if it is curl-free on $i, j, k, i.e. (curlX)(i, j, k) = X_{ij} + X_{jk} + X_{ki} = 0$.
X is called "globally consistent" if it is a gradient of a score function, i.e. X =

grad s for some $s : V \to R$.
X is called "locally consistent" if it is curl-free on every triangle in T (E), i.e. every 3-clique of G.
Clearly any gradient flow must be curl-free everywhere, i.e. curl grad = 0. Therefore, global consistency implies local consistency.

The HodgeRank gets rid of "imbalanced" ratings resulting from a simple average method. For example, if a product A gets 1 5-star rating and product B gets 100 5-star ratings but one single 1-star rating, ranking by average ratings would rank A ahead of B which does not make sense. The methodology is as follows: if a user has assigned items i and j cardinal scores s_i, s_j respectively, the degree of preference for item i over item j is $s_i - s_j$.
We could ignore the cardinal nature of the dataset and just use its ordinal information to construct a binary pairwise ranking, however, through this process we would lose valuable information, for example, a 5-star versus 2-star comparison is indistinguishable from a 4-star versus 2-star comparison when one only takes ordinal information into account.
We have $V = (1, ..., n)$ and $\wedge = (\alpha 1, ..., \alpha m)$ as the set of users and alternatives, respectively. We can form the weight matrix and pairwise comparison matrix from the data. The weight matrix is the symmetric n-by-n matrix defined by:

$$\begin{cases} [W_{ij}^{\alpha}] = 1 \ if \ \alpha \ made \ a \ pairwise \ comparison \\ [W_{ij}^{\alpha}] = 0 \ otherwise \end{cases}$$

The pairwise comparison matrix is given by:

$$\begin{cases} [Y_{ij}^{\alpha}] = degree \ of \ preference \ of \ i \ over \ j \ if \ [W_{ij}^{\alpha}] = 1 \\ [Y_{ij}^{\alpha}] = 0 \ if \ [W_{ij}^{\alpha}] = 0 \end{cases}$$

The matrix representing the graph Laplacian is given by:

$$\begin{cases} [\Delta_0]_{ij} = -w_{ij} \ if \ (i,j) \in E \\ [\Delta_0]_{ij} = \sum_k \ w_{ik} \ if \ i = j \\ [\Delta_0]_{ij} = 0 \ otherwise \end{cases}$$

The code calculates rank score through least squares as follows:
Given the weight matrix W, and Pairwise Comparison matrix Y, first we calculate the graph laplacian: $\Delta_0 = -W + diag(sum(W))$

Then we calculate the divergence: $divY = diag(Y * W^T)$

Then we calculate the rank score by using Moore-Penrose Pseudoinverse:
$r = pinv(\Delta_0) + divY$

The cyclicity ratio (a crude comparison of gradient and residual) tells us how confident we can be in our rankings. It is given as follows:

$C_r = \parallel R \parallel / \parallel grad(s) \parallel$
Here, grad (s) captures the component of our data that contains no cardinal or ordinal inconsistencies. R is the the least squares residual which is given by subtracting off the gradient component:
$R = Y - grad(s)$
If this ratio is less than 1, that means the gradient makes up a much larger portion of our data, and therefore, we have a reliable set of rankings.

In order to see how HodgeRank can be used in practice, we used several examples of NCAA Division I Basketball and NCAA Division I Football games from Sizemore's paper to assign a global ranking to each team. We compared our algorithm to his results and found similar results.[**Sizemore**]

Once the code was validated, we worked with a subset of Amazon dataset to rank each product (in toys/games category). The dataset consisted of user IDs (user that ranked a particular product), product IDs (for each product that was ranked by users) and a rating for each product. We computed Weights and Pairwise Comparison Matrices and used the code based on HodgeRank Methodology to compute global ranking for each product.

Conclusion

HodgeRank uses graph theory and topology to determine a global ranking on datasets as described in our paper. Jiang, Lim, Yao and Ye showed in their paper, how HodgeRanking helps in ranking datasets that have inconsistencies and incompleteness in data. We have shown how HodgeRank can be used to rank products listed on Amazon based on ratings by users. Given a set of products and their ratings assigned by various users, we form a Weight matrix (the number of times a product A is compared with product B) and a Pairwise Comparison matrix (summation of differences of rankings between products A and B each time they are compared divided by number of times they are compared). Then we solve the least squares problem to find the l2 projection of our pairwise comparison data onto the space of gradient flows and solve for a ranking score using Moore-Penrose Pseudoinverse. Based on our observation from Amazon dataset, it seems like some high rated products were being compared mostly with low rated products which gave the high rated products a boost compared to high rated products that were only compared with other high rated products. Therefore, we recommend that a final judgement should be a result of comparing rankings based on average ratings of products versus Hodgerank.

Appendix

Code

The code implemented in R Studio is as follows:
$library(MASS)$
$W = Weights$
$Y = PairwiseComp$
$vectornames = col\ names(Y)$
$\Delta_0 = -W + diag(rowSums(W))$
$\Delta_0 = as.matrix(\Delta_0)$
$Y = as.matrix(Y)$
$W^T = t(as.matrix(W))$
$divY = diag(Y * W^T)$
$r = ginv(\Delta_0) * divY$
$output = data.frame(vectornames, r)$
$outputsorted = output[with(output, order(r)),]$
$write.csv(outputsorted, file =' ranking.csv')$

In the above code, W represents the Weight matrix and Y represents Pairwise Comparison matrix.

Data

The subset of dataset we used is as follows:

User	Product	Rating
A31POTIYCKSZ9G	B000OMIA9W	5
A38IEZF0P3ZUQJ	B000OMZQA8	5
A38IEZF0P3ZUQJ	B000OMZQCQ	5
A2YIICW5DQ4MLK	B000ONVDFE	5
A2YIICW5DQ4MLK	B000ONXC6C	5
A10YMFWZY0SVS4	B000OO2B8G	3
A10YMFWZY0SVS4	B000OO2B90	4
AJGU56YG8G1DQ	B000OO4B8O	5
A24H96VU06OQCA	B000OOGQKK	4
A28UPG3EM8Q1IR	B000OOGQL4	5
A3O78MS3X2R118	B000OOGQL4	3
A24H96VU06OQCA	B000OOGQL4	5
A28UPG3EM8Q1IR	B000OOHUD2	5
A3O78MS3X2R118	B000OOHUD2	3
A3TVB8HUP3L9PK	B000RAW5LU	5
A33FU5ZYL4XNAQ	B000RAW5MY	5
AABPZUR8TZRDK	B000RAY4MI	5
A3TVB8HUP3L9PK	B000RAY4MI	4
A3TVB8HUP3L9PK	B000RAZJD6	3
AABPZUR8TZRDK	B000RB104C	5
A33FU5ZYL4XNAQ	B000RB104W	5
A1H41IC29WQVA6	B000RC0MC2	5
A1EKOIPHEFGJWU	B000RC0MC2	5
A1H41IC29WQVA6	B000RC0NU8	5
A1EKOIPHEFGJWU	B000RC0NU8	5
A1H41IC29WQVA6	B000RC2OC8	4
A1EKOIPHEFGJWU	B000RC2OC8	5
A1H41IC29WQVA6	B000RC2R9S	5
AJGU56YG8G1DQ	B000RC2VI0	5
AMEVO2LY6VEJA	B000RC2VI0	5
A2NOEX4EJZ8DV3	B000RDI3PY	5
A2NOEX4EJZ8DV3	B000RDJMEU	5
AMEVO2LY6VEJA	191639	5
A31POTIYCKSZ9G	76561046	3
A31POTIYCKSZ9G	B000OMIA9W	5
A38IEZF0P3ZUQJ	B000OMZQA8	5
A38IEZF0P3ZUQJ	B000OMZQCQ	5
AS8IOLI9AC5OX	B0019IAIWW	1
AS8IOLI9AC5OX	B0019IAOSU	4
AJGU56YG8G1DQ	B0019IARK0	5
AJGU56YG8G1DQ	B0019IAZ3Y	5
A4Y5PAH9NPQKJ	B000NW5RWQ	5
A23KVKN2IRPIQL	B000NW66D0	5
A23KVKN2IRPIQL	B000NW66DA	5
A4Y5PAH9NPQKJ	439028485	2

Output

The Ranking (output in descending order) is as follows:

Rank	Product
1	B0019IAOSU
2	B000NW5RWQ
3	B000OMIA9W
4	B000OO2B90
5	B000OOGQL4
6	B000OOHUD2
7	B000RAW5LU
8	B000RB104C
9	B000RC2R9S
10	B000RC0NU8
11	B000RC0MC2
12	B000RAY4MI
13	B000NW66DA
14	B000OMZQCQ
15	B000RDJMEU
16	B000ONVDFE
17	B000ONXC6C
18	B000RDI3PY
19	191639
20	B0019IARK0
21	B000RB104W
22	B000OO4B8O
23	B0019IAZ3Y
24	B000RAW5MY
25	B000RC2VI0
26	B000NW66D0
27	B000OMZQA8
28	B000RC2OC8
29	B000OO2B8G
30	B000OOGQKK
31	B000RAZJD6
32	76561046
33	439028485
34	B0019IAIWW

References

[1] Lek-Heng Lim. *Hodge Laplacians on Graphs.* https://arxiv.org/pdf/1507.05379.pdf

[2] Robert K Sizemore. *HodgeRank: Applying Combinatorial Theory to Sports Ranking.* Masters Thesis, Wake Forest University, 2013

[3] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, Yinyu Ye. *Statistical Ranking and Combinatorial Hodge Theory.* https://arxiv.org/pdf/0811.1067.pdf

[4] Allen Hatcher. *Algebraic Topology.* Cambridge University Press, 2001.

[5] R. He, J. McAuley. *Amazon Product Data.* Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016