HodgeRank with Information Maximization for Crowdsourced Pairwise Ranking Aggregation

Qianqian Xu[1,3], Jiechao Xiong[2,3], Xi Chen[4], Qingming Huang[5,6], Yuan Yao[7,3,✉]
[1] SKLOIS, Institute of Information Engineering, CAS, Beijing, China, [2] Tencent AI Lab, Shenzhen, China
[3] BICMR and School of Mathematical Sciences, Peking University, Beijing, China
[4] Department of IOMS, Stern School of Business, New York University, USA
[5] University of Chinese Academy of Sciences, Beijing, China, [6] IIP., ICT., CAS, Beijing, China
[7,✉] Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong
xuqianqian@iie.ac.cn, jcxiong@tencent.com, xichen@nyu.edu, qmhuang@ucas.ac.cn, yuany@ust.hk ✉

Abstract

Recently, crowdsourcing has emerged as an effective paradigm for human-powered large scale problem solving in various domains. However, task requester usually has a limited amount of budget, thus it is desirable to have a policy to wisely allocate the budget to achieve better quality. In this paper, we study the principle of information maximization for active sampling strategies in the framework of HodgeRank, an approach based on Hodge Decomposition of pairwise ranking data with multiple workers. The principle exhibits two scenarios of active sampling: Fisher information maximization that leads to unsupervised sampling based on a sequential maximization of graph algebraic connectivity without considering labels; and Bayesian information maximization that selects samples with the largest information gain from prior to posterior, which gives a supervised sampling involving the labels collected. Experiments show that the proposed methods boost the sampling efficiency as compared to traditional sampling schemes and are thus valuable to practical crowdsourcing experiments.

Introduction

The emergence of online paid crowdsourcing platforms, like Amazon Mechanical Turk, presents us new possibilities to distribute tasks to human workers around the world, on-demand and at scale. Recently, there arises a plethora of pairwise comparison data in crowdsourcing experiments on Internet (Liu 2011; Xu et al. 2016; Chen et al. 2013; Fu et al. 2014; Chen, Lin, and Zhou 2015), where the comparisons can be modeled as oriented edges of an underlying graph. As online workers can come and complete tasks posted by a company, and work for as long or as little as they wish, the data we collected are highly imbalanced where different alternatives might receive different number of comparisons, and incomplete with large amount of missing values. To analyze the imbalanced and incomplete data efficiently, the newly proposed Hodge theoretic approach (Jiang et al. 2011) provides us a simple yet powerful tool.

HodgeRank, introduced by (Jiang et al. 2011), is an application of combinatorial Hodge theory to the preference or rank aggregation from pairwise comparison data. In an analog to Fourier decomposition in signal processing, Hodge decomposition of pairwise comparison data splits the aggregated global ranking and conflict of interests into different orthogonal components. It not only generalizes the classical Borda count in social choice theory to determine a global ranking from pairwise comparison data under various statistical models, but also measures the conflicts of interests (i.e., inconsistency) in the pairwise comparison data. The inconsistency shows the validity of the ranking obtained and can be further studied in terms of its geometric scale, namely whether the inconsistency in the ranking data arises locally or globally.

A fundamental problem in crowdsourcing ranking is the *sampling* strategy, which is crucial to collect data efficiently. Typically, there are two ways to design sampling schemes: *random sampling* and *active sampling*. Random sampling is a basic type of sampling and the principle of random sampling is that every item has the same probability of being chosen at any stage during the sampling process. The most important benefit of random sampling over active methods is its simplicity which allows flexibility and generality to diverse situations. However, this non-selective manner does not sufficiently use the information of past labeled pairs, thus potentially increases the costs in applications. This motivates us to investigate efficient schemes for *active sampling*.

In this paper, we present a principle of active sampling based on *information maximization* in the framework of HodgeRank. Roughly speaking, Fisher's information maximization with HodgeRank leads to a scheme of unsupervised active sampling which does not depend on actual observed labels (i.e., a fixed sampling strategy before the data is observed). Since this sampling scheme does not need the feedback from the worker, it is fast and efficient. Besides, it is insensitive to outliers. On the other hand, a Bayesian information maximization equips us a supervised active sampling scheme that relies on the history of pairwise comparison data. By exploiting additional information in labels,

supervised sampling often exhibits better performances than unsupervised active sampling and random sampling. However as the supervised sampling is sensitive to outliers, while reliability/quality of each worker is heterogeneous and unknown in advance, we find that unsupervised active sampling is sometimes more efficient than supervised sampling when the latter selects outlier samples at the initial stage. Experimental results on both simulated examples and real-world data support the efficiency improvements of active sampling compared against passive random sampling.

Our contributions in this work are threefold:

1. A new version of *Hodge decomposition* of pairwise comparison data with multiple voters is presented. Within this framework, two schemes of information maximization, Fisher and Bayesian that lead to unsupervised and supervised sampling respectively, are systematically investigated.

2. Closed form update and a fast *online algorithm* are derived for *supervised sampling* with Bayesian information maximization for HodgeRank, which is shown faster and more accurate than the state-of-the-art method Crowd-BT (Chen et al. 2013).

3. These schemes exhibit better sampling efficiency than random sampling as well as a better *loop-free* control in clique complex of paired comparisons, thus reduce the possibility of causing voting chaos by harmonic ranking (Saari 2001) (i.e., the phenomenon that the inconsistency of preference data may lead to totally different aggregate orders using different methods).

Hodge-theoretic approach to ranking

Before introducing our active sampling schemes, we will first propose a new version of Hodge decomposition of pairwise labels to ranking.

From Borda count to HodgeRank

In crowdsourced pairwise comparison experiments, let V be the set of candidates and $|V| = n$. A voter (or worker) $\alpha \in A$ provides his/her preference for a pair of candidates $(i, j) \in V \times V$, $y_{ij}^{\alpha} : A \times V \times V \to \mathbb{R}$ such that $y_{ij}^{\alpha} = -y_{ji}^{\alpha}$, where $y_{ij}^{\alpha} > 0$ if α prefers i to j and $y_{ij}^{\alpha} \leq 0$ otherwise. The simplest setting is the binary choice, where

$$y_{ij}^{\alpha} = \begin{cases} 1 & \text{if } \alpha \text{ prefers } i \text{ to } j, \\ -1 & \text{otherwise.} \end{cases} \tag{1}$$

Such pairwise comparison data can be represented by a graph $G = (V, E)$, where $(i, j) \in E$ is an oriented edge when i and j are effectively compared by some voters. Associate each $(i, j) \in E$ a Euclidean space $\mathbb{R}^{|A_{ij}|}$ where A_{ij} denotes the voters who compared i and j. Now define $\mathcal{Y} := \otimes_{(i,j)\in E}\mathbb{R}^{|A_{ij}|}$, a Euclidean space with standard basis e_{ij}^{α}. In other words, for every pair of candidates, a vector space representing preferences of multiple voters or workers is attached to the corresponding graph edge, therefore $\mathcal{Y}$ can be viewed as a vector bundle or sheaf on the edge space E.

Statistical rank aggregation problem is to look for some global rating score from such kind of pairwise comparison data. One of the well-known methods for this purpose is the Borda count in social choice theory (Jiang et al. 2011), in which the candidate that has the most pairwise comparisons in favour of it from all voters will be ranked first, and so on. However, Borda count requires the data to be complete and balanced. To adapt to new features in modern datasets, i.e. incomplete and imbalanced, the following least squares problem generalizes the classical Borda count to scenarios from complete to incomplete voting,

$$\min_{x} \|y - D_0 x\|_2^2 \tag{2}$$

where $x \in \mathcal{X} := \mathbb{R}^{|V|}$ is a global rating score, $D_0 : \mathcal{X} \to \mathcal{Y}$ is a finite difference (coboundary) operator defined by $(D_0 x)(\alpha, i, j) = x_i - x_j$. In other words, here we are looking for a universal rating model independent to α, whose pairwise difference approximates the voter's data in least squares. We note that multiple models are possible if one hopes to group voters or pursue personalized ratings by extending the treatment in this paper.

Assume that G is connected, then solutions of (2) satisfy the following graph Laplacian equation which can be solved in nearly linear computational complexity (Spielman and Teng 2004; Cohen et al. 2014)

$$D_0^T D_0 x = D_0^T y \tag{3}$$

where $L = D_0^T D_0$ is the weighted graph Laplacian defined by $L(i, j) = -m_{ij}$ ($m_{ij} = |A_{ij}|$) for $i \neq j$ and $L(i, i) = \sum_{j:(i,j)\in E} m_{ij}$. The minimal norm least squares estimator is given by $\hat{x} = L^{\dagger} D_0^T y$ where $L^{\dagger}$ is the Moore-Penrose inverse of L.

A new version of Hodge decomposition

With the aid of combinatorial Hodge theory, the residue of (2) can be further decomposed adaptive to the topology of clique complex $\chi_G = (V, E, T)$, where $T = \{(i, j, k) : (i, j) \in E, (j, k) \in E, (k, i) \in E\}$ collects the oriented triangles (3-cliques) of G. To see this, define $\mathcal{Z} = \mathbb{R}^{|T|}$ and the triangular curl (trace) operator $D_1 : \mathcal{Y} \to \mathcal{Z}$ by $(D_1 y)(i, j, k) = \frac{1}{m_{ij}} \sum_{\alpha} y_{ij}^{\alpha} + \frac{1}{m_{jk}} \sum_{\alpha} y_{jk}^{\alpha} + \frac{1}{m_{ki}} \sum_{\alpha} y_{ki}^{\alpha}$. Plugging in the definition of D_0, it is easy to see $(D_1(D_0 x))(i, j, k) = (x_i - x_j) + (x_j - x_k) + (x_k - x_i) = 0$. In the following, we extend the existing HodgeRank methodology from simple graph with skew-symmetric preference to multiple digraphs with any preference, which potentially allows to model different users' behaviour. In particular, the existing Hodge decomposition (Jiang et al. 2011) only considers the simple graph, which allows only one (oriented) edge between two nodes where pairwise comparisons are aggregated as a mean flow on the edge. However, in crowdsourcing applications, each pair is labeled by multiple workers. Therefore, there will be multiple inconsistent edges (edges in different directions) for each pair of nodes. Also the pairwise comparison data may

not be skew-symmetric, for example home advantage of sports games. To meet this challenge, we need to extend existing theory to the following new version of Hodge decomposition theorem adapted to the multi-worker scenario.

Theorem 1 (Hodge Decomposition Theorem) *Consider chain map*

$$\mathcal{X} \xrightarrow{D_0} \mathcal{Y} \xrightarrow{D_1} \mathcal{Z}$$

with the property $D_1 \circ D_0 = 0$. Then for any $y \in \mathcal{Y}$, the following orthogonal decomposition holds

$$y = b + u + D_0 x + D_1^T z + w, \tag{4}$$

$$w \in \ker(D_0^T) \cap \ker(D_1),$$

where b is the symmetric part of y, i.e. $b_{ij}^\alpha = b_{ji}^\alpha = (y_{ij}^\alpha + y_{ji}^\alpha)/2$, which captures the position bias of pairwise comparison on edge (α, i, j). The other four are skew-symmetric. u is a universal kernel satisfying $\sum_\alpha u_{ij}^\alpha = 0, \forall (i,j) \in E$ indicating all pairwise comparisons are completely in tie, x is a global rating score, z captures mean triangular cycles and w is called harmonic ranking containing long cycles irreducible to triangular ones.

The proof is provided in the supplementary materials. In fact all the components except b and $D_0 x$ are of cyclic rankings, where the universal kernel u as a complete tie is bi-cyclic for every edge $(i,j) \in E$. By adding 3-cliques or triangular faces to G, the clique complex χ_G thus enables us to separate the triangle cycles $D_1^T z$ from the cyclic rankings. Similarly one can define dimension-2 faces of more nodes, such as quadrangular faces etc., to form a *cell complex* to separate high order cycles via Hodge decomposition. Here we choose clique complex χ_G for simplicity. The remaining harmonic ranking w is generically some long cycle involving all the candidates in comparison, therefore it is the source of voting or ranking chaos (Saari 2001) (a.k.a. fixed tournament issue in computer science), i.e., any candidate i can be the final winner by removing some pairwise comparisons containing the opponent j who beats i in such comparisons. Fortunately harmonic ranking can be avoided by controlling the topology of underlying simplicial complex χ_G; in fact Hodge theory tells us that harmonic ranking will vanish if the clique complex χ_G (or cell complex in general) is loop-free, i.e., its first Betti number being zero. In this case, the harmonic ranking component will be decomposed into local cycles such as triangular cycles. Therefore in applications it is desired to have the simplicial complex χ_G loop free, which is studied later in this paper with active sampling schemes. For this celebrated decomposition, the approach above is often called *HodgeRank* in literature.

When the preference data y is skew-symmetric, the bias term b vanishes, there only exists a global rating score and cyclic rankings. Cyclic rankings part mainly consists of noise and outliers, where outliers have much larger magnitudes than normal noise. So a sparse approximation of the cyclic rankings for pairwise comparison data can be used to detect outliers. In a mathematical way, suppose **Proj** is the projection operator to the cyclic ranking space, then $\mathbf{Proj}(\gamma)$ with a sparse outlier vector γ is desired to approximate $\mathbf{Proj}(y)$. One popular method is LASSO as following:

$$\min_\gamma \|\mathbf{Proj}(y) - \mathbf{Proj}(\gamma)\|_2^2 + \lambda \|\gamma\|_1$$

Further more, the term b models the user-position bias in the preference. It means on the edge (α, i, j) and (α, j, i), there is a bias caused by various reasons, such as which one is on the offensive. While in most crowdsourcing problems, we believe there should not have a such term unless the worker is careless. So this term can be used to model the workers' behavior. In formulation, we can add an intercept term into (2):

$$\min_x \|y - b - D_0 x\|_2^2 \tag{5}$$

where b is a piecewise constant intercept depending on worker α only: $b_{ij}^\alpha = constant_\alpha, \forall i, j$. Such an intercept term can be seen as a mean effect of the position bias for each worker. The bigger its magnitude is, the more careless the worker is. Generally, this term can be any piecewise constant vector which models different group effect of bias. This potentially allows to model different workers' behavior.

Statistical models under HodgeRank

HodgeRank provides a unified framework to incorporate various statistical models, such as Uniform model, Thurstone-Mosteller model, Bradley-Terry model, and especially Mosteller's Angular Transform model which is essentially the only model having the asymptotic variance stabilization property. These are all generalized linear models for binary voting. In fact, generalized linear models assume that the probability of pairwise preference is fully decided by a linear function as follows

$$\pi_{ij} = \mathbf{Prob}\{i \succ j\} = \Phi(x_i^* - x_j^*),\ x^* \in \mathcal{X} \tag{6}$$

where $\Phi : \mathbb{R} \to [0,1]$ can be chosen as any symmetric cumulated distributed function. In a reverse direction, if an empirical preference probability $\hat{\pi}_{ij}$ is observed in experiments, one can map $\hat{\pi}$ to a skew-symmetric pairwise comparison data by the inverse of Φ, $\hat{y}_{ij} = \Phi^{-1}(\hat{\pi}_{ij})$. Then solving the HodgeRank problem (2) is actually solving the weighted least squares problem for this generalized linear model. Different choices of Φ lead to different generalized linear models, e.g. $\Phi(t) = e^t/(1+e^t)$ gives Bradley-Terry model and $\Phi(t) = (\sin(t)+1)/2$ gives Mosteller's Angular Transform model.

Information Maximization for Sampling in HodgeRank

Our principle for active sampling is *information maximization*. Depending on the scenarios in application, the definition of *information* varies. There are often two ways to design active sampling strategies depending

on available information: (1) unsupervised active sampling without considering the actual labels collected, where we use Fisher information to maximize algebraic connectivity in graph theory; (2) supervised active sampling with label information, where we exploit a Bayesian approach to maximize expected information. In the following, we will first introduce the unsupervised active sampling, followed by the supervised active sampling. After that, an online algorithm of supervised active sampling will be detailed. Finally, we discuss the online tracking of topology evolutions of the sampling schemes.

Fisher information maximization: unsupervised sampling

In case that the cyclic rankings in (4) are caused by Gaussian noise, i.e. $u+D_1^T z+w=\epsilon$ where $\epsilon \sim \mathcal{N}(0,\Sigma_\epsilon)$, the least squares problem (2) is equivalent to the following Maximum Likelihood problem:

$$\max_x \frac{(2\pi)^{-m/2}}{\det(\Sigma_\epsilon)} \exp\left(-\frac{1}{2}(y-D_0x)^T\Sigma_\epsilon^{-1}(y-D_0x)\right),$$

where Σ_ϵ is the covariance matrix of the noise, $m = \sum_{(i,j)\in E} m_{ij}$. In applications without a priori knowledge about noise, we often assume the noise is independent and has unknown but fixed variance σ_ϵ^2, i.e. $\Sigma_\epsilon = \sigma_\epsilon I_m$. So HodgeRank here is equivalent to solve the Fisher's Maximum Likelihood with Gaussian noise. Now we are ready to present a sampling strategy based on Fisher information maximization principle.

Fisher Information Maximization: The log-likelihood is

$$l(x) = -m\log(\sqrt{2\pi}\sigma_\epsilon) - \frac{1}{2}(y-D_0x)^T\Sigma_\epsilon^{-1}(y-D_0x).$$

So the Fisher Information is given as

$$I := -E\frac{\partial^2 l}{\partial x^2} = D_0^T\Sigma_\epsilon^{-1}D_0 = L/\sigma_\epsilon^2. \tag{7}$$

where $L = D_0^T D_0$ is the weighted graph Laplacian.

Given a sequence of samples $\{\alpha_t, i_t, j_t\}_{t\in N}$ (edges), the graph Laplacian can be defined recursively as $L_t = L_{t-1} + d_t^T d_t$, where $d_t : \mathcal{X} \to \mathcal{Y}$ is defined by $(d_t x)(\alpha_t, i_t, j_t) = x_{i_t} - x_{j_t}$ and 0 otherwise. Our purpose is to maximize the Fisher information given history via

$$\max_{(\alpha_t,i_t,j_t)} f(L_t) \tag{8}$$

where $f : S_+^n \to R$ is a concave function w.r.t the weights on edges. Since it is desired that the result does not depend on the index V, f has to be permutation invariant. A stronger requirement is orthogonal invariant $f(L) = f(O^T LO)$ for any orthogonal matrix O, which implies that $f(L) = g(\lambda_2,\ldots,\lambda_n)$, $0=\lambda_1\le\lambda_2\le\cdots\le\lambda_n$ are the eigenvalues of L (Chandrasekaran, Pablo, and Willsky 2012). Note that it does not involve sampling labels and is thus an unsupervised active sampling scheme.

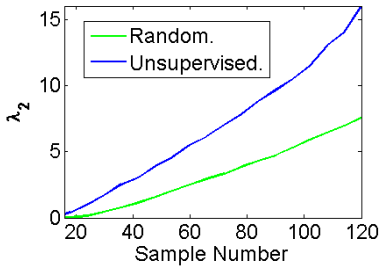


Figure 1: Fiedler value comparison of unsupervised active sampling vs. random sampling.

Among various choices of f, a popular one is $f(L_t) = \lambda_2(L_t)$, where $\lambda_2(L_t)$ is the smallest nonzero eigenvalue (a.k.a. algebraic connectivity or Fiedler value) of L_t, which corresponds to "E-optimal" in experimental design (Osting, Brune, and Osher 2014). Despite that (8) is a convex optimization problem with respect to real-valued graph weights, the optimization over integral weights is still NP-hard and a greedy algorithm (Ghosh and Boyd 2006) can be used as a first-order approximation

$$\begin{aligned}\max \lambda_2(L_t) &\approx \max[\lambda_2(L_{t-1}) + \|d_t v_2(L_{t-1})\|^2] \\ &= \lambda_2(L_{t-1}) + \max(v_2(i_t) - v_2(j_t))^2,\end{aligned}$$

where v_2 is the second nonzero eigenvector or Fiedler vector of L_{t-1}. Figure 1 shows Fiedler value plots of two sampling schemes, where unsupervised active sampling above effectively raises the Fiedler value curve than random sampling.

While the unsupervised sampling process only depends on L_t, label information is collected for the computation of HodgeRank global ranking estimator $\hat{x}^t = L_t^\dagger (D_0^t)^T y^t$, where $D_0^t = D_0^{t-1} + d_t$ and $y^t = y^{t-1} + y_{i_t j_t}^{\alpha_t} e_{i_t j_t}^{\alpha_t}$.

Algorithm 1: Unsupervised active sampling algorithm.

Input: An initial graph Laplacian L_0 defined on the graph of n nodes.

for $t = 1,\ldots,T$ **do**
- Compute the second eigenvector v_2 of L_{t-1}.;
- Select the pair (i_t, j_t) which maximizes $(v_2(i_t) - v_2(j_t))^2$.;
- Draw a sample on the edge (i_t, j_t) with voter α_t.;
- Update graph Laplacian L_t.;

end

Output: Sampling sequence $\{\alpha_t, i_t, j_t\}_{t\in N}$.

Bayesian information maximization: supervised sampling

Since the least squares problem (2) is invariant under the shift of x, a small amount of regularization is al-

ways preferred. Therefore in practice (2) is understood as the minimal norm least squares solution, or the ridge regularization,

$$\min_x \|y - D_0 x\|_2^2 + \gamma \|x\|_2^2. \tag{9}$$

Regularization on x means a prior distribution assumption on x. So (9) is equivalent to

$$\max_x \exp\left(-\frac{\|y - D_0 x\|_2^2}{2\sigma_\epsilon^2} - \frac{\|x\|_2^2}{2\sigma_x^2}\right), \tag{10}$$

when $\sigma_\epsilon^2/\sigma_x^2 = \gamma$. So regularized HodgeRank is equivalent to the Maximum A Posterior (MAP) estimator when both the likelihood and prior are Gaussian distributions.

With such a Bayesian perspective, a natural scheme for active sampling is based on the maximization of expected information gain (EIG) or Kullback-Leibler divergence from prior to posterior. In each step, the most informative triplet (object i, object j, annotator α) is added based on the largest KL-divergence between posterior and prior. The maximization of EIG has been a popular criterion in active sampling (Settles 2009) and applied to some specific pairwise comparison models (e.g. (Chen et al. 2013) applied EIG to Bradley-Terry model with Gaussian prior and (Pfeiffer et al. 2012) to Thurstone-Mosteller model). Combining the EIG criterion with the ℓ_2-regularized HodgeRank formulation in (10), we obtain a simple closed form update for the posterior for general models, which leads to a fast online algorithm.

Bayesian information maximization: Let $P^t(x|y^t)$ be the posterior of x given data y^t. So given present data y^t, we choose a new pair to maximize the expected information gain (EIG) of a new pair (i, j):

$$(i^*, j^*) = \arg\max_{(i,j)} EIG_{(i,j)} \tag{11}$$

where

$$EIG_{(i,j)} := E_{y_{ij}^{t+1}|y^t} KL(P^{t+1}|P^t) \tag{12}$$

and the KL-divergence

$$KL(P^{t+1}|P^t) := \int P^{t+1}(x|y^{t+1}) \ln \frac{P^{t+1}(x|y^{t+1})}{P^t(x|y^t)} dx$$

Once an optimal pair (i^*, j^*) is determined from (11), we assign this pair to a random voter $\alpha \in A$ and then collect the corresponding label for the next update.

In the l_2-regularized HodgeRank setting, such a optimization problem in (11) can be greatly simplified.

Proposition 1 *When both the likelihood and prior are Gaussian distributions, then posterior $P^t(x|y^t)$ is also Gaussian.*

$$x|y^t \sim N(\mu^t, \sigma_\epsilon^2 \Sigma^t)$$

$$\mu^t = (L_t + \gamma I)^{-1} (D_0^t)^T y^t, \Sigma^t = (L_t + \gamma I)^{-1}.$$

Thus

$$\begin{aligned} & 2KL(P^{t+1}|P^t) && (13) \\ = \; & \frac{1}{\sigma_\epsilon^2}(\mu^t - \mu^{t+1})^T (L_t + \gamma I)(\mu^t - \mu^{t+1}) - n \\ & + tr((L_t + \gamma I)(L_{t+1} + \gamma I)^{-1}) \\ & + \ln \frac{\det(L_{t+1} + \gamma I)}{\det(L_t + \gamma I)} && (14) \end{aligned}$$

and the posterior $y_{ij}^{t+1}|y^t \sim N(a, b)$ with $a = \mu_i^t - \mu_j^t, b = (\Sigma_{ii}^t + \Sigma_{jj}^t - 2\Sigma_{ij}^t + 1)\sigma_\epsilon^2$.

Remark 1 *Note the first term of $KL(P^{t+1}|P^t)$ is l_2 distance of gradient flow between μ^t and μ^{t+1} if $\gamma = 0$. The unknown parameter σ_ϵ needs a roughly estimation. In binary comparison data, $\sigma_\epsilon = 1$ is good enough. Given the history D_0^t, y^t and the new edge (i, j), μ^{t+1} is only a function of y_{ij}^{t+1}, so does $KL(P^{t+1}|P^t)$.*

Generally, the posterior of y_{ij}^{t+1}

$$p(y_{ij}^{t+1}|y^t) = \int p(y_{ij}^{t+1}|x) P^t(x|y^t) dx$$

can be approximated by $p(y_{ij}^{t+1}|\hat{x}^t)$, where $\hat{x}^t$ is the HodgeRank estimator μ^t. In practice, we receive binary comparison data $y_{ij}^\alpha \in \{\pm 1\}$, hence we can adopt generalized additive models $\hat{\pi}(y_{ij}^\alpha = 1) = \Phi(\hat{x}_i - \hat{x}_j)$ to compute it explicitly.

Such a Bayesian information maximization approach relies on actual labels collected in history, as sampling process depends on y^t through μ^t. Hence it is a supervised active sampling scheme, in contrast to the previous one.

Online supervised active sampling algorithm

To update the posterior parameters efficiently, we would like to introduce an accelerating method using Sherman-Morrison-Woodbury formula (Bartlett and Maurice 1951). In active sampling scheme, the Bayesian information maximization approach needs to compute EIG for $\binom{n}{2}$ times to choose one pair. And each EIG consists of the computation of inverting an $n \times n$ matrix, which costs $O(n^3)$ and is especially expensive for large scale data. But notice that L_{t+1} and L_t only differs by a symmetric rank-1 matrix, Sherman-Morrison-Woodbury formula can be applied to greatly accelerate the sampling procedure.

Denote $L_{t,\gamma} = L_t + \gamma I$, so $L_{t+1,\gamma} = L_{t,\gamma} + d_{t+1}^T d_{t+1}$, then Sherman-Morrison-Woodbury formula can be rewritten as follows:

$$L_{t+1,\gamma}^{-1} = L_{t,\gamma}^{-1} - \frac{L_{t,\gamma}^{-1} d_{t+1}^T d_{t+1} L_{t,\gamma}^{-1}}{1 + d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T} \tag{15}$$

Proposition 2 *Using the Sherman-Morrison-Woodbury formula, Eq (13) can be further simplified*

as

$$KL(P^{t+1}|P^t) \quad (16)$$
$$= \frac{1}{2}[\frac{1}{\sigma_\epsilon^2}(\frac{y_{ij}^{t+1} - d_{t+1}\mu^t}{1+C})^2 C + \ln(1-C) - \frac{C}{1+C}]$$

where $C = d_{t+1}L_{t,\gamma}^{-1}d_{t+1}^T$ *and*

$$\mu^{t+1} = \mu^t + \frac{y_{ij}^{t+1} - d_{t+1}\mu^t}{1+C} L_{t,\gamma}^{-1} d_{t+1}^T. \quad (17)$$

Now for each pair of nodes (i, j), we only need to compute $d_{t+1}L_{t,\gamma}^{-1}d_{t+1}^T$ and $d_{t+1}\mu^t$. Since d_{t+1} has the form of $e_i - e_j$, so it only costs $O(1)$ which is much cheaper than the original $O(n^3)$. The explicit formula of KL-divergence (15) makes the computation of EIG easy to vectorize, especially useful in MATLAB. Also note that if we can store the matrix $L_{t,\gamma}^{-1}$, (14) provides the formula to update $L_{t,\gamma}^{-1}$ and (16) provides the update of score function μ^t. Combining these two posterior update rules, the entire online active algorithm is presented in Algorithm 2.

Algorithm 2: Online supervised active sampling algorithm for binary comparison data.

Input: Prior distribution parameters $\gamma, \mu^0, L_{0,\gamma}^{-1}$.

for $t = 0, 1, \ldots, T-1$ **do**

  For each pair (i, j), compute the expected information gain in Eq. (12) and Eq. (15) using $\sigma_\epsilon = 1$;

  Select the pair (i^*, j^*) which has maximal EIG.;

  Draw a sample on the edge (i^*, j^*) from a randomly chosen voter α_t and observe the next label $y_{i^*j^*}^{t+1}$.;

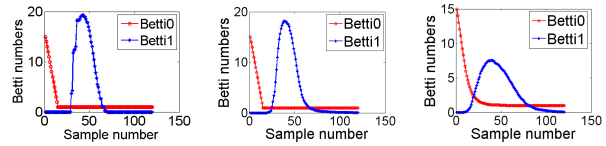
  Update posterior parameters according to (14) and (16).;

end

Output: Ranking score function μ^T.

Online tracking of topology evolution

In HodgeRank, two topological properties of clique complex χ_G have to be considered which are obstructions for obtaining global ranking and harmonic ranking. First of all, a global ranking score can be obtained, up to a translation, only if the graph G is connected, so one needs to check the number of connected components as the zero-th Betti number β_0. Even more importantly, the voting chaos indicated by harmonic ranking w in (4) vanishes if the clique complex is loop-free, so it is necessary to check the number of loops as the first Betti number β_1. Given a stream of paired comparisons, persistent homology (Edelsbrunner, Letscher, and Zomorodian 2002; Carlsson 2009) is in fact an online algorithm to check topology evolution when simplices (e.g. nodes, edges, and triangles) enter in a sequential way such that the subset inclusion order is respected. Here we just discuss in brief the application of persistent homology to monitor the number of connected components (β_0) and loops (β_1) in three different sampling settings.



(a) Unsupervised. (b) Supervised. (c) Random.

Figure 2: Average Betti numbers for three sampling schemes.

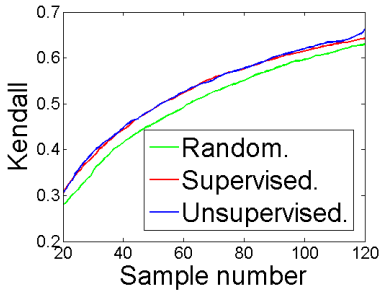


Figure 3: The mean Kendall's τ between ground-truth and HodgeRank estimator for three sampling schemes.

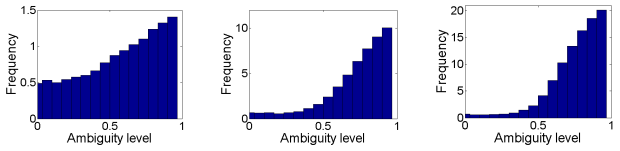
Assume that the nodes come in a certain order (e.g., production time, or all created in the same time), after that pairs of edges are presented to us one by one guided by the corresponding sampling scheme. A triangle $\{i, j, k\}$ is created whenever all the three associated edges appeared. Persistent homology may return the evolution of the number of connected components (β_0) and the number of independent loops (β_1) at each time when a new node/edge/triangle is born. The expected β_0 and β_1 (with 100 graphs) computed by Javaplex (Sexton and Johansson 2009) for $n = 16$ of three sampling schemes are plotted in Figure 2. It is easy to see that both unsupervised & supervised active sampling schemes narrow the nonzero region of β_1, which indicates that these two active sampling schemes both enlarge the loop-free regions thus reduce the chance of harmonic ranking or voting chaos.

Experiments

In this section, we study examples with both simulated and real-world data to illustrate the validity of the proposed two schemes of active sampling.

Simulated data

In this experiment, we use simulated data to illustrate the performance differences among unsupervised & supervised active sampling, and random sampling. We first randomly create a global ranking score x as the



(a) T=K=120 (b) T=5K=600 (c) T=10K=1200

Figure 4: Sampling counts for pairs with different levels of ambiguity in supervised active sampling.

Table 1: Computational complexity (s) comparison on simulated data.

n	16	20	24	28	32	100
Offline Sup.	25.22	81.65	225.54	691.29	1718.34	>7200
Online Sup.	0.10	0.17	0.26	0.38	0.50	15.93
Unsup.	0.75	1.14	4.27	6.73	9.65	310.58

ground-truth, uniformly distributed on [0,1] for n candidates. Then we sample pairs from this graph using these three sampling schemes. The pairwise comparisons are generated by uniform model, i.e. $y_{ij}^{\alpha} = 1$ with probability $(x_i - x_j + 1)/2$, $y_{ij}^{\alpha} = -1$ otherwise. Averagely, there are $30\% - 35\%$ comparisons are in the wrong direction, $(x_i - x_j)y_{ij}^{\alpha} < 0$. The experiments are repeated 1000 times and ensemble statistics for the HodgeRank estimator are recorded.

- **Kendall's τ comparison.** First, we adopt the Kendall rank correlation (τ) coefficient (Kendall and Maurice 1948) to measure the rank correlation between ground-truth and HodgeRank estimator of these three sampling schemes. Figure 3 shows the mean Kendall's τ associated with these three sampling schemes for $n = 16$ (chosen to be consistent with the first two real-world datasets considered later). The x-axes of the graphs are the number of samples added, taken to be greater than $\frac{\log n}{n}$ percentage so that the random graph is connected with high probability. From these experimental results, we observe that both active sampling schemes, with a similar performance, show better efficiency than random sampling with higher Kendall's τ.
- **Computational cost.** Table 1 shows the computational complexity achieved by online/offline algorithms of supervised active sampling and unsupervised active sampling. The total number of edges added is $\binom{n}{2}$ and the value in this table represents the average time (s) needed of 100 runs for different n. All computation is done using MATLAB R2014a, on a Mac Pro desktop PC, with 2.8 GHz Intel Core i7-4558u, and 16 GB memory. It is easy to see that online supervised algorithm is faster than unsupervised active sampling. Besides, it can achieve up to hundreds of times faster than offline supervised algorithm, with exactly the same performances. And more importantly, as n increases, such a benefit is increasing, which further implies its advantage in dealing with large-scale data.
- **Budget Level.** Next, we would like to investigate how the total budget is allocated among pairs with different levels of ambiguity in supervised active sampling scheme. In particular, we first randomly create a global ranking score as the ground truth, uniformly distributed on [0, 1] for n candidates, corresponding to $K = \binom{n}{2}$ pairs with different ambiguity levels (i.e., 1-abs (ground-truth score differences). In this experiment, $n = 16$, we vary the total budget $T = K, 5K, 10K$, and report the number of times that each pair is sampled on average over 100 runs. The results are presented in Figure 4. It is easy to see that more ambiguous pairs with *Ambiguity Level* close to 1 in general receive more labels than those simple pairs close to 0. This is consistent with practical applications, in which we should not spend too much budget on those easy pairs, since they can be decided based on the common knowledge and majority voting, excessive efforts will not bring much additional information.

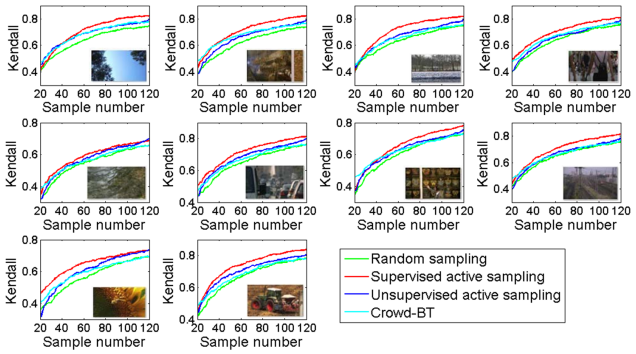


Figure 5: Experimental results of four sampling schemes for 10 reference videos in LIVE database.

Real-world data

The first example gives a comparison of these three sampling schemes on a complete & balanced video quality assessment (VQA) dataset (Xu et al. 2011). It contains 38,400 paired comparisons of the LIVE dataset (LIV 2008) from 209 random observers. As there is no ground-truth scores available, results obtained from all the paired comparisons are treated as the ground-truth. To ensure the statistical stability, for each of the 10 reference videos, we sample using each of the three methods for 100 times. For comparison, we also conduct experiments with the state-of-the-art method Crowd-BT (Chen et al. 2013). Figure 5 shows the results and these different reference videos exhibit similar observations. Consistent with the simulated data, the proposed unsupervised/supervised active sampling performs better than random sampling scheme in the prediction of global ranking scores, and the performance of supervised active sampling is slightly better than unsupervised active sampling with higher kendall's τ. Moreover, our supervised active sampling consistently manages to improve the kendall's τ of Crowd-BT by roughly 5%.

The second example shows the sampling results on an

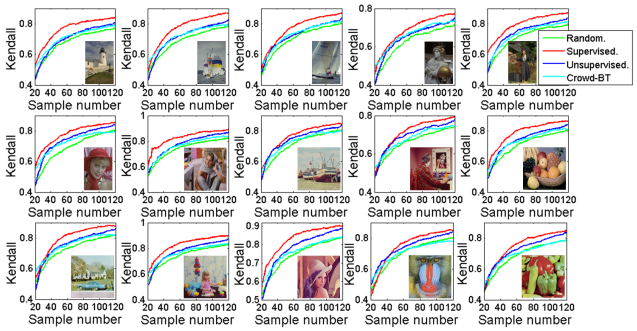


Figure 6: Experimental results of four sampling schemes for 15 reference images in LIVE and IVC databases.

imbalanced dataset for image quality assessment (IQA), which contains 43,266 paired comparisons of 15 reference images (LIV 2008)(IVC 2005) from 328 random observers on Internet. As this dataset is relatively large and edges occurred on each paired comparison graph with 16 nodes are dense, all the 15 graphs are also complete graph, though possibly imbalanced. Figure 6 shows mean Kendall's τ of 100 runs, and similarly for all reference images active sampling schemes show better performance than random sampling. Besides, our proposed supervised active sampling also performs better than Crowd-BT.

In the third example, we test our method to the task of ranking documents by their reading difficulty. This dataset (Chen et al. 2013) is composed of 491 documents. Using the **CrowdFlower** crowdsourcing platform, 624 distinct annotators from the United States and Canada provide us a total of 12,728 pairwise comparisons. For better visualization, we only present the mean Kendall's τ of 100 runs for the first 4,000 pairs in Figure 7. As captured in the figure, the proposed supervised active strategy significantly outperforms the random strategy. We also compare our method with Crowd-BT and it is easy to see that our method also improves over the Crowd-BT method's performance.

- **Running cost.** More importantly, our method is much faster than Crowd-BT by orders of magnitude due to closed-form posterior in Proposition 1 and fast online computation in Proposition 2. Table 2 shows the comparable computational cost of these two methods using the same settings with Table 1. It is easy to see that On VQA dataset, for a reference video, 100 runs of Crowd-BT take about 10 minutes on average; while our online supervised algorithm takes only 18 seconds, which is 33 times faster. Besides, our method can achieve nearly 40 times speed-up on IQA dataset and 35 times faster on reading level dataset. In a word, the main advantages of our method lies in its computational efficiency and the ability to handle streaming data.
- **Parameter tuning.** A crucial question here is how to choose γ in supervised active sampling experiments. In practice, for dense graph, we find that γ makes little difference in the experimental results, a smaller γ, say 0.01, or even $1e^{-5}$ is sufficient. However, for sparse graph such as reading level dataset, a bigger γ (i.e., $\gamma = 1$) may produce better performance.

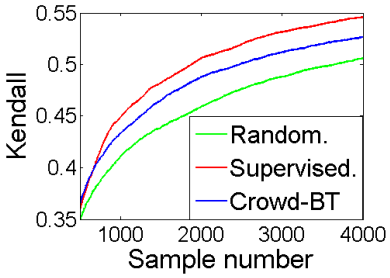


Figure 7: Experimental results of three sampling schemes on reading level dataset.

Table 2: Average running cost (s) of 100 runs on three real-world datasets.

Method	Our supervised method	Crowd-BT
VQA dataset	18	600
IQA dataset	12	480
Reading level dataset	120	4200

Conclusions

In this paper, we proposed a new Hodge decomposition of pairwise comparison data with multiple voters and analyzed two active sampling schemes in this framework. In particular, we showed that: 1) for unsupervised active sampling without considering the actual labels, we can use Fisher information to maximize algebraic connectivity in graph theory; 2) for supervised active sampling with label information, we can exploit a Bayesian approach to maximize expected information gain from prior to posterior. The unsupervised sampling involves the computation of a particular eigenvector of graph Laplacians, the Fiedler vector, which can be precomputed *a priori*; while the supervised sampling benefits from a fast online algorithm using the Sherman-Morrison-Woodbury formula for matrix inverses, which however depends on the label history. Both schemes enable us a more efficient budget control than passive random sampling, tested with both simulated and real-world data, hence provide a helpful tool for researchers who exploit crowdsourced pairwise comparison data.

Acknowledgments

The research of Qianqian Xu was supported by National Key Research and Development Plan (No.2016YFB0800403), National Natural Science Foundation of China (No.U1636214, 61422213, 61672514, 61390514, 61572042), CCF-Tencent Open Research Fund. The research of Xi Chen was supported in part by Google Faculty Research Award and Adobe Data Science Research Award. The research of Qing-

ming Huang was supported in part by National Natural Science Foundation of China: 61332016, U1636214, 61650202 and 61620106009, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. The research of Yuan Yao was supported in part by Hong Kong Research Grant Council (HKRGC) grant 16303817, National Basic Research Program of China (No. 2015CB85600, 2012CB825501), National Natural Science Foundation of China (No. 61370004, 11421110001), as well as awards from Tencent AI Lab, Si Family Foundation, Baidu BDI, and Microsoft Research-Asia.

References

Bartlett, and Maurice, S. 1951. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics* 107–111.

Carlsson, G. 2009. Topology and data. *Bulletin of the American Mathematical Society* 46(2):255–308.

Chandrasekaran, V.; Pablo, A.; and Willsky, A. 2012. Convex graph invariants. *SIAM Review* 54(3):513–541.

Chen, X.; Bennett, P.; Collins-Thompson, K.; and Horvitz, E. 2013. Pairwise ranking aggregation in a crowdsourced setting. In *ACM international conference on Web search and data mining*, 193–202.

Chen, X.; Lin, Q.; and Zhou, D. 2015. Statistical decision making for optimal budget allocation in crowd labeling. *Journal of Machine Learning Research* 16:1–46.

Cohen, M.; Kyng, R.; Miller, G.; Pachocki, J.; Peng, R.; Rao, A.; and Xu, S. 2014. Solving sdd linear systems in nearly m log 1/2 n time. In *ACM Symposium on Theory of Computing*, 343–352.

Edelsbrunner, H.; Letscher, D.; and Zomorodian, A. 2002. Topological persistence and simplification. *Discrete and Computational Geometry* 28(4):511–533.

Fu, Y.; Hospedales, T.; Xiang, T.; Gong, S.; and Yao, Y. 2014. Interestingness prediction by robust learning to rank. In *European Conference on Computer Vision*, 488–503.

Ghosh, A., and Boyd, S. 2006. Growing well-connected graphs. *IEEE Conference on Decision and Control* 6605–6611.

2005. Subjective quality assessment irccyn/ivc database. http://www2.irccyn.ec-nantes.fr/ivcdb/.

Jiang, X.; Lim, L.-H.; Yao, Y.; and Ye, Y. 2011. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming* 127(1):203–244.

Kendall, and Maurice, G. 1948. *Rank Correlation Methods*. Griffin.

Liu, T. 2011. *Learning to Rank for Information Retrieval*. Springer.

2008. LIVE image & video quality assessment database. http://live.ece.utexas.edu/research/quality/.

Osting, B.; Brune, C.; and Osher, S. J. 2014. Optimal data collection for informative rankings expose well-connected graphs. *Journal of Machine Learning Research* 15:2981–3012.

Pfeiffer, T.; Gao, X. A.; Mao, A.; Chen, Y.; and Rand, D. G. 2012. Adaptive polling for information aggregation. In *AAAI*.

Saari, D. 2001. *Chaotic Elections! A mathematician looks at voting*. American Mathematical Society.

Settles, B. 2009. Active learning literature survey. Technical report, University of Wisconsin–Madison.

Sexton, H., and Johansson, M. 2009. JPlex: a java software package for computing the persistent homology of filtered simplicial complexes. `http://comptop.stanford.edu/programs/jplex/`.

Spielman, A., and Teng, S. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *ACM Symposium on Theory of Computing*, 81–90.

Xu, Q.; Jiang, T.; Yao, Y.; Huang, Q.; Yan, B.; and Lin, W. 2011. Random partial paired comparison for subjective video quality assessment via HodgeRank. 393–402. ACM Multimedia.

Xu, Q.; Xiong, J.; Cao, X.; and Yao, Y. 2016. False discovery rate control and statistical quality assessment of annotators in crowdsourced ranking. In *International Conference on Machine Learning*, 1282–1291.

Supplementary Materials

A. Proof of Hodge Decomposition Theorem

Let $b_{ij}^\alpha = b_{ji}^\alpha = (y_{ij}^\alpha + y_{ji}^\alpha)/2$, then $y - b$ is skew-symmetric, and $\langle y - b, b\rangle = 0$. So W.L.O.G, we only need to prove the theorem with skew-symmetric preference y.

Now, consider the following least squares problem for each $(i,j) \in E$,

$$\bar{y}_{ij} = \arg\min_c \sum_\alpha (y_{ij}^\alpha - c)^2.$$

Define $\bar{y} \in \mathcal{Y}$ by $\bar{y}_{ij}^\alpha = \bar{y}_{ij}$, then define

$$u := y - \bar{y}.$$

Clearly u satisfies $\sum_\alpha u_{ij}^\alpha = 0$ and hence $\langle u, \bar{y}\rangle = 0$.

Now consider Hilbert spaces $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ and chain map

$$\mathcal{X} \xrightarrow{D_0} \mathcal{Y} \xrightarrow{D_1} \mathcal{Z}$$

with the property $D_1 \circ D_0 = 0$. Define the product Hilbert space $\mathcal{H} = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ and let Dirac operator $\nabla : \mathcal{H} \to \mathcal{H}$ be

$$\nabla = \begin{pmatrix} 0 & 0 & 0 \\ D_0 & 0 & 0 \\ 0 & D_1 & 0 \end{pmatrix}.$$

Define a Laplacian operator

$$\Delta = (\nabla + \nabla^*)^2 = \mathrm{diag}(D_0^T D_0, D_0 D_0^T + D_1^T D_1, D_1 D_1^T)$$

where $(\cdot)^T$ denotes the adjoint operator. Then by Rank-nullity Theorem, $\mathrm{im}(\nabla) + \ker(\nabla^T) = \mathcal{H}$, in particular the middle space admits the decomposition

$$\begin{aligned}\mathcal{Y} &= \mathrm{im}(D_0) + \ker(D_0^T) \\ &= \mathrm{im}(D_0) + \ker(D_0^T)/\mathrm{im}(D_1^T) + \mathrm{im}(D_1^T), \\ &\qquad \text{since } \mathrm{im}(D_0) \subseteq \ker(D_1), \\ &= \mathrm{im}(D_0) + \ker(D_0^T) \cap \ker(D_1) + \mathrm{im}(D_1^T).\end{aligned}$$

Now apply this decomposition to $\bar{y} = y - u \in \mathcal{Y}$, we have $D_0 x \in \mathrm{im}(D_0)$, $D_1^T z \in \mathrm{im}(D_1^T)$, and $w \in \ker(D_0^T) \cap \ker(D_1)$.

B. Proof of Proposition 1

The posterior distribution of x is proportional to

$$\begin{aligned}&\exp\left(-\frac{\|y - D_0 x\|_2^2}{2\sigma_\epsilon^2} - \frac{\|x\|_2^2}{2\sigma_x^2}\right) \\ =\ &\exp\left(-\frac{\|y - D_0 x\|_2^2 + \gamma\|x\|_2^2}{2\sigma_\epsilon^2}\right) \\ \sim\ &\exp\left(-\frac{(x-\mu^t)^T(L_t + \gamma I)(x - \mu^t)}{2\sigma_\epsilon^2}\right).\end{aligned}$$

So $x|y$ is gaussian distribution with mean $(L_t + \gamma I)^{-1} D_0^T y$ and covariance $\sigma_\epsilon^2 (L_t + \gamma I)^{-1}$.

$$y_{ij}^{t+1} = (x_i - x_j) + \epsilon_{ij}^{t+1}$$

is a linear combination of gaussian variables, so it is also gaussian.

The KL-divergence between two gaussian distributions has an explicit formulation

$$\begin{aligned}&2KL(P^{t+1}|P^t) \\ =\ &(\mu^t - \mu^{t+1})^T(\sigma_\epsilon^2 \Sigma^t)^{-1}(\mu^t - \mu^{t+1}) \\ &+tr((\Sigma^t)^{-1}\Sigma^{t+1}) - \ln\frac{\det(\Sigma^{t+1})}{\det(\Sigma^t)} - n \\ =\ &\frac{1}{\sigma_\epsilon^2}(\mu^t - \mu^{t+1})^T(L_t + \gamma I)(\mu^t - \mu^{t+1}) - n \\ &+tr((L_t + \gamma I)(L_{t+1} + \gamma I)^{-1}) \\ &+ \ln\frac{\det(L_{t+1} + \gamma I)}{\det(L_t + \gamma I)}.\end{aligned}$$

C. Proof of Proposition 2

Note that $\mu^t = L_{t,\gamma}^{-1}(D_0^t)^T y^t$, so

$$\begin{aligned}\mu^{t+1} &= L_{t+1,\gamma}^{-1}(D_0^{t+1})^T y^{t+1} \\ &= (L_{t,\gamma}^{-1} - \frac{L_{t,\gamma}^{-1} d_{t+1}^T d_{t+1} L_{t,\gamma}^{-1}}{1 + d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T}) \\ &\quad \cdot((D_0^t)^T y^t + d_{t+1}^T y_{ij}^{t+1}) \\ &= \mu^t + \frac{y_{ij}^{t+1} - d_{t+1}\mu^t}{1 + d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T} L_{t,\gamma}^{-1} d_{t+1}^T.\end{aligned}$$

Moreover

$$\begin{aligned}tr((L_{t,\gamma})(L_{t+1,\gamma})^{-1}) &= tr(I - \frac{d_{t+1}^T d_{t+1} L_{t,\gamma}^{-1}}{1 + d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T}) \\ &= n - \frac{d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T}{1 + d_{t+1} L_{t,\gamma}^{-1} d_{t+1}^T},\end{aligned}$$

and

$$\begin{aligned}\frac{\det(L_{t+1,\gamma})}{\det(L_{t,\gamma I})} &= \det((L_{t,\gamma})^{-1} L_{t+1,\gamma}) \\ &= \det(I + (L_{t,\gamma})^{-1} d_{t+1}^T d_{t+1}) \\ &= 1 - d_{t+1}(L_{t,\gamma})^{-1} d_{t+1}^T.\end{aligned}$$

Last equation uses $d_{t+1} = e_i - e_j$. Plugging all these identities into Proposition 1, we can get the result.