A Theoretical Analysis of Query Selection for Collaborative Filtering

Wee Sun Lee[1] and Philip M. Long[2]

[1] Department of Computer Science
National University of Singapore
Singapore 117543, Republic of Singapore
[2] Genome Institute of Singapore
1 Research Link
IMA Building
National University of Singapore
Singapore 117604, Republic of Singapore

Abstract. We consider the problem of determining which of a set of experts has tastes most similar to a given user by asking the user questions about his likes and dislikes. We describe a simple and fast algorithm for a theoretical model of this problem with a provable approximation guarantee, and prove that solving the problem exactly is NP-Hard.

1 Introduction

Recommender systems (also known as collaborative £ltering systems) use the opinions of past users to make recommendations to new users. The design of many such systems is based on the assumption that people with similar opinions about some things are likely to have similar opinions about others (see [10, 4]). The user is typically asked to rate a few items before any new item is recommended. Once a suf£cient number of items have been rated, the system can use those ratings to estimate which previous users of the system are most similar to the current user overall. The opinions of these previous users can then used to generate recommendations; methods based on weighted majority prediction [8] and correlation coef£cients [9] usually work quite well for this.

In this paper, we investigate a different aspect of the problem: how to select the initial items for the user to rate. These items are not presented as recommendations, but are asked only for the purpose of learning about the user. Since these are troublesome to the user, a high priority must be placed on asking few of these questions. (This is in contrast to the work on "approximate nearest neighbor searching" [3, 7, 5], where all the components of the point being searched are assumed to be given.) If later questions are decided based on the answers to earlier questions, the questions must also be generated in real time.

We allow the ratings to come from any £nite set Y, and assume that the algorithm is given an integer-valued loss function ℓ on Y to measure the distance between different ratings. We require that the loss function ℓ be a metric, that is, it satis£es the properties:

$\ell(x, y) \geq 0$, $\ell(x, y) = 0$ if and only if $x = y$, $\ell(x, y) = \ell(y, x)$ and $\ell(x, y) \leq \ell(x, z) + \ell(z, y)$ for any $z \in Y$. Common loss functions that satisfy these properties include the $0 - 1$ loss and the absolute loss. The distance between users will then be measured by the sum, over all items, of the loss between their ratings on a given item.

The emphasize the role that they play, we refer to the previous users as *experts*; our approximation bounds will be in terms of the number of such experts. Therefore, it may be worthwhile to cluster the previous users in a preprocessing step, and use the cluster centers as the experts.

Before proceeding to the general case, we illustrate our techniques in a highly idealized setting. We assume that there are only two possible ratings, that the distance between ratings is 1 if they are different and 0 if they are the same, and that some expert agrees with the user on all items. In this case, the problem can be described in terms of the *membership query model* [1].

In the membership query model [1], the learning algorithm is trying to learn an unknown $\{0, 1\}$-valued function f (called the "target") chosen from a known concept class F. The algorithm is allowed to ask the value of $f(x)$ for domain elements x of its choosing, and must eventually halt and output the identity of f.

In the idealized case described above, the problem of £nding the perfect expert can be viewed as the problem of learning using membership queries. The different items would be the domain X, the likes and dislikes of the user is the function f to be learned, and asking the user its opinion about an item can be interpreted as a membership query. The experts are then the concept class F. Viewed this way, the problem we are faced with is that of, given a concept class F as input, designing a membership query algorithm for F.

We begin by showing that the very simple and fast "query-by-committee" [11] algorithm, which maintains a list of possible targets, and chooses the query for which the remaining possibilities are most evenly divided, learns any class F with an approximately optimal number of membership queries in the worst case. Speci£cally, if $\mathrm{opt}(F)$ is the optimal worst-case bound on the number of membership queries for learning arbitrary elements of F, then the query-by-committee algorithm learns F while making at most $\mathrm{opt}(F)(\ln(|F|/\mathrm{opt}(F)) + 1) + 1$ queries. We also show that it is NP-Hard to design a polynomial-time algorithm that, given F as input and an membership oracle for an element f of F, is guaranteed to learn f using $\mathrm{opt}(F)$ queries.

Next, we look at the more general case. To study this case, we use a variant of the membership query model similar to that proposed by Angluin, Krikis, Sloan and Turán [2]. Here, the range of the target f (our model of the user) and the functions in F (the experts) is an arbitrary £nite set Y. As mentioned above, the algorithm is given an integer-valued metric ℓ on $Y \times Y$, and the distance between functions f and g is measured by $\sum_{x \in X} \ell(f(x), g(x))$. The target function f is not necessarily in F, but the algorithm is given a parameter η such that there is a function g in F at a distance at most η from f. The algorithm must output some element of F within distance η (there may be more than one). Let us refer to the optimal worst-case bound on the number of queries for this model by $\mathrm{opt}(F, \eta)$.

The algorithm we analyze for this problem also maintains a list of elements of F that are "alive"; here, these are elements that might possibly be within distance η of the

target function f. Loosely speaking, it repeatedly chooses a domain element for which any response will discredit the remaining possibilities in total by a large amount.

To analyze this algorithm, we make use of a quantity that we call the η-degree of F. In the recommender system application, this can be interpreted as a measure of the diversity of opinion among the experts; for example, if any possible target f is at a distance at most η from a unique element of F, then the η-degree of F is 0. The motivation for this measure is strongest if we imagine that F is the result of a clustering preprocessing step. Note that, informally, if F consists of the centers of tight clusters, and users typically belong to one such cluster, being much closer to one element of F than to any other should often be expected in practice. Tight clustering is the implicit assumption underlying the design of many collaborative £ltering systems.

One can view the de£nition of η-degree as follows: imagine centering balls of radius η at the elements of F, and constructing a graph where the vertices are these balls, and there are edges between pairs of vertices that overlap. The η-degree of F is the edge degree of that graph.

Our generalization of the query-by-committee algorithm is guaranteed to £nd an element of F within distance η after at most

$$2\mathrm{opt}(F,\eta)\ln\frac{|F|}{1+\deg(F,\eta)}+\eta(1+\deg(F,\eta))$$

queries. Thus, if each possible target is within distance η of a unique element of F, $2\mathrm{opt}(F,\eta)\ln|F|+\eta$ queries suf£ce.

2 Membership queries

Fix some £nite domain X. For some function f from X to $\{0,1\}$, a membership oracle for f, when queried about an element x of X, returns $f(x)$. For an algorithm A with access to a membership oracle for f, let $Q(A,f)$ be the number of queries asked by A before it outputs the identity of f. For a class F of functions from X to $\{0,1\}$, let $Q(A,F)$ be the maximum of $Q(A,f)$ over all $f\in F$. Let $\mathrm{opt}(F)$ be the minimum of $Q(A,F)$ over all algorithms A (note that there is no limitation on the time taken by A).

In this section, we show that there is an algorithm that takes F as input, and, given access to an oracle for an arbitrary element f of F, learns f with a nearly optimal number of queries in polynomial time.

The algorithm analyzed in this section is the "query-by-committee" [11] algorithm. (Our analysis of it builds on Johnson's analysis of his approximation algorithm for Set Cover [6].) It maintains a list of the elements of F consistent with the answers received so far, and asks the query x that divides the elements the most evenly, i.e. for which the number of "alive" functions g for which $g(x)=1$ and the number for which $g(x)=0$ are as close as possible. After receiving $f(x)$, those possibilities that are inconsistent with this value are deleted, and the algorithm continues. When only one possibility remains, the algorithm halts and outputs it.

The key lemma in our analysis is the following.

Lemma 1. *For any domain X, and any £nite set F of at least two functions from X to $\{0,1\}$, there is an x for which* $\min_{y\in\{0,1\}}|\{f\in F:f(x)=y\}|\geq(|F|-1)/\mathrm{opt}(F)$.

Proof: Let A be an optimal membership query algorithm for F. Assume for contradiction that for all $x \in X$, either $|\{f \in F : f(x) = 1\}| < (|F| - 1)/\mathrm{opt}(F)$, or $|\{f \in F : f(x) = 0\}| < (|F| - 1)/\mathrm{opt}(F)$.

Our strategy will be to use the fact that any possible query that A could ask has an answer that eliminates few possibilities to argue that after asking a certain number of queries, A cannot know the function to be learned. We will design an adversary that repeatedly gives the answer to A that eliminates the fewest possibilities.

Let $F_0 = F$ (in general, F_t will be the possibilities remaining after t queries have been asked). Let x_1 be the £rst query asked by A. Choose y_1 to minimize $|\{f \in F : f(x_1) = y_1\}|$. Let $F_1 = \{f \in F : f(x_1) = y_1\}$. Then, by assumption, $|F_1| - 1 > |F| - 1 - (|F| - 1)/\mathrm{opt}(F)$.

Continuing, let each x_t be the tth query asked, and choose y_t to minimize $|\{f \in F : f(x_t) = y_t\}|$, and let $F_t = \{f \in F : f(x_1) = y_1, ..., f(x_t) = y_t\}$. For each such t, $|F_t| - 1 > |F_{t-1}| - 1 - (|F| - 1)/\mathrm{opt}(F)$. Telescoping,

$$|F_{\mathrm{opt}(F)}| - 1 > (1 - \mathrm{opt}(F)/\mathrm{opt}(F))(|F| - 1) = 0. \tag{1}$$

Thus, after $\mathrm{opt}(F)$ queries, there is more than one element of F consistent with the information received by A, a contradiction. □

Theorem 1. *For any £nite set X, and any £nite set F of at least two functions from X to $\{0, 1\}$, the query-by-committee algorithm, given F and a membership oracle for any arbitrary $f \in F$, outputs f after asking at most*

$$\mathrm{opt}(F)\left(1 + \ln\frac{|F| - 1}{\mathrm{opt}(F)}\right) + 1$$

queries.

Proof: Choose F, and a target $f \in F$. Suppose the query-by-committee algorithm asks T queries before learning f, and for each $0 \le t \le T$, let F_t be the set of functions in F consistent with the information received after the £rst t queries. Lemma 1 implies that for all $t \le T$,

$$\begin{aligned} |F_t| - 1 &\le (1 - 1/\mathrm{opt}(F_{t-1}))(|F_{t-1}| - 1) \\ &\le (1 - 1/\mathrm{opt}(F))(|F_{t-1}| - 1). \end{aligned} \tag{2}$$

We also have

$$|F_t| \le |F_{t-1}| - 1. \tag{3}$$

Let S be largest index for which $|F_S| - 1 \ge \mathrm{opt}(F)$. Then (2) implies

$$\left(1 - \frac{1}{\mathrm{opt}(F)}\right)^S (|F| - 1) \ge \mathrm{opt}(F)$$

and, applying the fact that $\forall x, 1 - x \le e^{-x}$ and solving for S, we get

$$S \le \mathrm{opt}(F) \ln\frac{|F| - 1}{\mathrm{opt}(F)}.$$

Also, (3), together with the fact that $|F_{T-1}| > 1$, implies that

$$T - S < \mathrm{opt}(F) + 1,$$

completing the proof. □

2.1 Hardness result

We now show that the problem of, given F, learning an arbitrary element of f with $\mathrm{opt}(F)$ membership queries is NP-Hard. Since this is a special case of our model of the query selection problem for collaborative £ltering, this problem is NP-hard also. Our proof is via a reduction from the set covering problem.

An instance (X', F') of the *set covering problem* consists of a £nite set X' and a family F' of subsets of X' such that every element of X' belongs to at least one subset of F'. The problem is to £nd the minimum-sized subset $C \subseteq F'$, such that every element of X' belongs to at least one subset of C.

For any instance of the set covering problem where $|X'| = n$ and $|F'| = m$, we will construct an optimal query problem whose solution will give the solution to the set covering problem. We £rst describe a closely related optimal query problem. An instance of the optimal query problem can be given as an $|F| \times |X|$ matrix. We construct an initial matrix M of size $n \times m$, where an element of X' in the set covering problem corresponds to an element in F of our optimal query problem while an element of F' in the set covering problem corresponds to an element of X in our optimal query problem. For each subset $s \in F'$, the ith entry in the corresponding column of matrix M is set to 1 if the element of X' corresponding to the ith row is a member of s, otherwise it is set to 0. Assume that we augment the matrix M with an all zero row which we set as the target for a query algorithm. Each query corresponds to a member of F' and will eliminate all the rows with 1's at that column. When the target is identi£ed, all the rows except the all zero row will have been eliminated. The subset of F' corresponding to the subset of queries will thus form a cover for X'.

However, our optimal query algorithm is guaranteed to give only the optimal *worst case* number of queries which does not necessarily correspond to the smallest cover. In order to ensure that the optimal *worst case* queries gives us the optimal cover, we solve the optimal query problem for an augmented matrix M'. We £rst augment the matrix M with n columns that are zero in every entry. Then we augment the augmented matrix with $n+1$ rows. The last row of the matrix consist of all zero elements while the $n+i$th row ($i = 1, \ldots, n$) consist of all zeros except for element $m + i$ which is set to 1. Call the doubly augmented matrix M'. We will show that if an optimal query algorithm for the matrix M' uses at most $n + q$ queries, then the optimal cover for the corresponding covering problem has size q.

The all zero row will be used as the target of an optimal query algorithm. Each query corresponding to the one of the £rst m columns corresponds to a member of F'. Hence the subset of F' corresponding to queries from the £rst m columns will form a cover of X'. We call this subset the cover *generated* by the query algorithm. Note that the rows $n + 1$ to $2n$ cannot be eliminated by any query except the query to the column where they have entry 1 and that such a query will eliminate only one row. Hence to uniquely identify the target, n of the queries must be to the last n columns. We now need to show that the cover generated by the optimal query algorithm is an optimal cover.

Lemma 2. *The cover generated by an optimal query algorithm for matrix M' is a cover of the smallest size.*

Proof: Let A be an optimal query algorithm. Assume that the cover generated by A is not a cover of the smallest size. Hence it is possible to reduce the number of queries needed to identify the all zero row target by generating a smaller cover. Let B be an algorithm that uses the fewest number of queries to identify the all zero row target. We transform B into another algorithm B'. The algorithm B' has the property that any query on the last n columns always happens after the queries to the £rst m columns. This can be done by delaying the queries on the last n columns while retaining the ordering of the other queries. Since a query to column $m+i$ can eliminate only row $n+i$, the effect of the delays is to potentially reduce the number of queries required for the £rst n rows while potentially increasing the number of queries required for the other rows. The number of queries required for the all zero rows remain the same.

The algorithm B' will take the optimal number of queries for identifying the all zero row target and no more than n queries to identify any of the £rst n rows. This is because rows $n+1$ to $2n+1$ are identically zero when restricted to the £rst m columns, giving effectively a matrix with $n+1$ distinct rows. At least $n+1$ queries is needed by any algorithm to identify the all zero row target and the all zero row target always takes more queries than rows $n+1$ to $2n$. Hence, algorithm B' has better worst case performance than algorithm A contradicting the optimality of algorithm A. □

3 General Case

Choose a £nite nonempty set Y, a positive integer M, and a metric ℓ mapping $Y \times Y$ to the nonnegative integers.

For functions f and g from X to Y de£ne the distance $d(f,g)$ between f and g by

$$d(f,g) = \sum_{x \in X} \ell(f(x), g(x)).$$

Let $F^{\oplus \eta}$ consist of all $g : X \to \{0,1\}$ such that there is an $f \in F$ for which $d(f,g) \leq \eta$.

In this model, the adversary picks a function f from $F^{\oplus \eta}$ (which we will call the target), and provides an evaluation oracle for f to the algorithm; this oracle responds to a query of x with $f(x)$. The algorithm then must output $h \in F$ such that $d(f,h) \leq \eta$ (there may be more than one such possibility). The worst case number of queries for an algorithm A in this setting is $Q(A,F,\eta)$, and the optimal number of queries is $\mathrm{opt}(F,\eta)$.

For a function $f : X \to Y$, and $U \subseteq X$, denote the restriction of f to U by $f_{|U}$.

For a set F of functions from X to Y, de£ne the η-degree of F, denoted by $\deg(F,\eta)$, to be the maximum, over all $g \in F$, of $|\{f \in F : 0 < d(f,g) \leq 2\eta\}|$. For $\mu : F \to \mathbf{Z}^+$, de£ne $\phi(F,\mu)$ to be the maximum, over all $g \in F$, of

$$\sum_{f \in F : d(f,g) \leq \mu(f) + \mu(g)} \mu(f).$$

For technical reasons, we will consider a related model. In this model, instead of η, the learning algorithm is given a priori a function $\mu : F \to \mathbf{Z}^+$ called a *quota function*,

and access to an evaluation oracle for some $f : X \to Y$ such that there is an $g \in F$ with $d(f, g) \leq \mu(g)$. The algorithm then must output an $h \in F$ such that $d(f, h) \leq \mu(h)$. Let $\mathrm{opt}(F, \mu)$ be the optimal worst-case number of queries for learning in this model.

Algorithm Our algorithm works in time polynomial in $|X|$, $|Y|$, and $|F|$. (Note that the latter is signi£cantly less than $|F^{\oplus \eta}|$.)

Our algorithm (let's call it B_F), is de£ned recursively as follows. Suppose at some point in time B_F has previously asked queries $x_1, ..., x_{t-1}$, which were answered with $y_1, ..., y_{t-1}$ respectively. Let

$$F_t = \{f_{|_{X - \{x_1, ..., x_{t-1}\}}} : f \in F, \sum_{s<t} \ell(f(x_s), y_s) \leq \eta\}.$$

Informally F_t consists of the restrictions of those elements of F that are "still alive" to the unexplored portion of X. If $|F_t| = 1$, it halts, and outputs an extension h of the single element of F_t to all of X that minimizes $\sum_{s<t} \ell(h(x_s), y_s)$. Otherwise, it chooses x_t from $X - \{x_1, ..., x_{t-1}\}$ in order to maximize

$$\min_{y \in Y} \sum_{f \in F_t} \ell(f(x_t), y_t).$$

The following is the main lemma in our analysis of this algorithm.

Lemma 3. *Choose a £nite X, a £nite Y, a £nite set F of at least 2 functions from X to Y, and $\mu : F \to \mathbf{Z}^+$. There is an $x \in X$ for which*

$$\min_{y \in Y} \sum_{f \in F} \ell(f(x_t), y_t) \geq \frac{1}{\mathrm{opt}(F, \mu)} \left(\left(\sum_{f \in F} (1 + \mu(f)) \right) - \phi(F, \mu) \right)$$

Proof: Let A be an optimal membership query algorithm for learning F with a quota function μ in the model of this section. Let $T = \mathrm{opt}(F, \mu)$. Assume without loss of generality that A always asks exactly T queries before halting and outputting a function in F. Assume for contradiction that

$$\forall x, \exists y, \sum_{f \in F} \ell(f(x), y) < \frac{1}{T} \left(\left(\sum_{f \in F} (1 + \mu(f)) \right) - \phi(F, \mu) \right). \tag{4}$$

Generate $(x_1, y_1), ..., (x_T, y_T)$ recursively as follows. For each t, let x_t be A's query when its previous queries $x_1, ..., x_{t-1}$ were answered with $y_1, ..., y_{t-1}$ respectively. Choose

$$y_t = \mathrm{argmin}_u \sum_{f \in F} \ell(f(x_t), u). \tag{5}$$

First, we claim that $(x_1, y_1), ..., (x_T, y_T)$ are "legal", in the sense that there is at least one potential target function f such that $f(x_1) = y_1, ..., f(x_T) = y_T$ for which

there is a $g \in F$ with $d(f,g) \leq \mu(g)$. To see this, note that (4) and (5) imply

$$\begin{aligned}\sum_{g\in F}\sum_{t=1}^{T}\ell(g(x_t),y_t) &= \sum_{t=1}^{T}\sum_{g\in F}\ell(g(x_t),y_t)\\ &< \left(\sum_{g\in F}(1+\mu(g))\right) - \phi(F,\mu) \qquad (6)\\ &\leq \left(\sum_{g\in F}(1+\mu(g))\right).\end{aligned}$$

Thus, there is a $g \in F$ such that $\sum_{t=1}^{T}\ell(g(x_t),y_t) \leq \mu(g)$. So if f is de£ned by $f(x_1) = y_1, ..., f(x_T) = y_T$ and $f(x) = g(x)$ for $x \notin \{x_1, ..., x_T\}$, f satis£es the requirements of a target function.

Suppose A outputs h. Loosely speaking, any $f \in F$ that is too far from h had better be eliminated as a possible target by $(x_1, y_1), ..., (x_T, y_T)$, since otherwise an adversary could choose f as a target. Speci£cally, for any $f \in F$ such that $d(h,f) > \mu(h)+\mu(f)$, it must be the case that $\sum_{t=1}^{T}\ell(f(x_t),y_t) > \mu(f)$, since otherwise, an adversary could modify f to get a target function with distance at most $\mu(f)$ from f, and therefore distance greater than $\mu(h)$ from h. Thus

$$\begin{aligned}\sum_{f\in F}\sum_{t=1}^{T}\ell(f(x_t),y_t) &\geq \sum_{f\in F: d(f,h)>\mu(f)+\mu(h)}(1+\mu(f))\\ &\geq \left(\sum_{f\in F}(1+\mu(f))\right) - \sum_{f\in F: d(f,h)\leq\mu(f)+\mu(h)}\mu(f)\\ &\geq \left(\sum_{f\in F}(1+\mu(f))\right) - \phi(F,\mu),\end{aligned}$$

contradicting (6) and completing the proof. □

Now we're ready for our theorem about B_F.

Theorem 2. *Choose X, a set F of functions from X to Y, and an integer $\eta \geq 1$. Then $Q(B_F, F, \eta) \leq 2\mathrm{opt}(F,\eta)\ln(|F|/(1+\deg(F,\eta))) + \eta(1+\deg(F,\eta))$.*

Proof: Consider a run of algorithm B_F in which it asks queries $x_1, ..., x_T$, which are answered with $y_1, ..., y_T$. For each t, let

$$F_t = \{f_{X-\{x_1,...,x_{t-1}\}} : f \in F, \sum_{S<t}\ell(f(x_s),y_s) \leq \eta\}.$$

For each t, de£ne $\mu_t : F_t \to \mathbf{Z}^+$ by

$$\mu_t(f) = \eta - \min\Big\{\sum_{s<t}\ell(g(x_s),y_s) : g \in F, g_{|_{X-\{x_1,...,x_{t-1}\}}} = f\Big\}.$$

Informally, $\mu_t(f)$ is the amount of loss left before f is eliminated as a possible target.

We divide our analysis of B_F into two stages. Let

$$S = \max \left\{ t : \sum_{f \in F_t} (1 + \mu_t(f)) \geq 2\eta(1 + \deg(F, \eta)) \right\}.$$

Choose $t \leq S$. By Lemma 3,

$$\begin{aligned}
&\sum_{f \in F_{t+1}} (1 + \mu_{t+1}(f)) \\
&\leq \left(\sum_{f \in F_t} (1 + \mu_t(f)) \right) - \sum_{f \in F_t} \ell(f(x_t), y_t) \\
&\leq \left(\sum_{f \in F_t} (1 + \mu_t(f)) \right) - \frac{1}{\mathrm{opt}(F_t, \mu_t)} \left(\left(\sum_{f \in F_t} (1 + \mu_t(f)) \right) - \phi(F_t, \mu_t) \right). \quad (7)
\end{aligned}$$

We will now prove that

$$\phi(F_t, \mu_t) \leq \eta(1 + \deg(F, \eta)). \quad (8)$$

For each $f \in F_t$, let $f^E \in F$ be obtained by extending f to X so as to minimize $\sum_{s<t} \ell(f^E(x_s), y_s)$. Recall that

$$\phi(F_t, \mu_t) = \max_{g \in F_t} \sum_{f \in F_t : d(f,g) \leq \mu_t(f) + \mu_t(g)} \mu_t(f).$$

Choose $g_* \in F_t$ achieving this maximum. We have

$$\begin{aligned}
\eta(1 + \deg(F, \eta)) &= \eta \max_{g \in F} |\{f \in F : d(f, g) \leq 2\eta\}| \\
&\geq \eta |\{f \in F : d(f, g_*^E) \leq 2\eta\}| \\
&= \sum_{f \in F : d(f, g_*^E) \leq 2\eta} \eta \\
&\geq \sum_{f \in F_t : d(f^E, g_*^E) \leq 2\eta} \eta. \quad (9)
\end{aligned}$$

For any $f, g \in F_t$,

$$\begin{aligned}
d(f^E, g^E) &= \sum_{x \in X} \ell(f^E(x), g^E(x)) \\
&= d(f, g) + \sum_{s<t} \ell(f^E(x_s), g^E(x_s)) \\
&\leq d(f, g) + \sum_{s<t} \ell(f^E(x_s), y_s) + \ell(g^E(x_s), y_s) \\
&\leq d(f, g) + 2\eta - (\mu_t(f) + \mu_t(g))
\end{aligned}$$

since, by de£nition, for all $f \in F_t$, $\sum_{s<t} \ell(f^E(x_s), y_s) \leq \eta - \mu_t(f)$. Thus (9) implies

$$\begin{aligned}\eta(1+\deg(F,\eta)) &\geq \sum_{f\in F_t : d(f,g_*)\leq \mu_t(f)+\mu_t(g_*)} \eta \\ &\geq \sum_{f\in F_t : d(f,g_*)\leq \mu_t(f)+\mu_t(g_*)} \mu_t(f),\end{aligned}$$

proving (8).

Putting (8) together with (7), we have

$$\begin{aligned}&\sum_{f\in F_{t+1}} (1+\mu_{t+1}(f)) \\ &\leq \left(\sum_{f\in F_t}(1+\mu_t(f))\right) - \frac{1}{2\mathrm{opt}(F_t,\mu_t)} \sum_{f\in F_t}(1+\mu_t(f)) \\ &= \left(1 - \frac{1}{2\mathrm{opt}(F_t,\mu_t)}\right) \sum_{f\in F_t}(1+\mu_t(f)) \\ &\leq \left(1 - \frac{1}{2\mathrm{opt}(F,k)}\right) \sum_{f\in F_t}(1+\mu_t(f)).\end{aligned}$$

Thus,

$$\sum_{f\in F_S}(1+\mu_S(f)) \leq \left(1 - \frac{1}{2\mathrm{opt}(F,\eta)}\right)^S |F|(\eta+1).$$

But, by de£nition, $\sum_{f\in F_S}(1+\mu_S(f)) \geq 2\eta(1+\deg(F,\eta))$, and so

$$\begin{aligned}2\eta(1+\deg(F,\eta)) &\leq \left(1 - \frac{1}{2\mathrm{opt}(F,\eta)}\right)^S |F|(\eta+1) \\ 2\eta(1+\deg(F,\eta)) &\leq \exp\left(-\frac{S}{2\mathrm{opt}(F,\eta)}\right)|F|(\eta+1) \\ \ln 2 + \ln\eta + \ln(1+\deg(F,\eta)) &\leq -\frac{S}{2\mathrm{opt}(F,\eta)} + \ln|F| + \ln(\eta+1) \\ S &\leq 2\mathrm{opt}(F,\eta)\ln(|F|/(1+\deg(F,\eta))).\end{aligned}$$

For all $t \leq T$, since $|F_t| > 1$, and $\ell(u,v) \geq 1$ for $u \neq v$,

$$\sum_{f\in F_{t+1}}(1+\mu_{t+1}(f)) \leq \left(\sum_{f\in F_t}(1+\mu_t(f))\right) - 1. \tag{10}$$

Since $\sum_{f\in F_{S+1}}(1+\mu_{S+1}(f)) < 2\eta(1+\deg(F,\eta))$ and $\sum_{f\in F_T}(1+\mu_T(f)) \geq 1$, (10) implies that $T - S \leq 2\eta(1+\deg(F,\eta))$. This completes the proof. □

4 Acknowledgements

We gratefully acknowledge the support of National University of Singapore Academic Research Fund grant R252–000–070–107.

References

1. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
2. D. Angluin, M. Krikis, R. H. Sloan, and G. Turán. Malicious omissions and errors in answers to membership queries. *Machine Learning*, 28:211–255, 1997.
3. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 573–582, 1994.
4. J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative ltering. *Proceedings of the Fourteenth Conference on Uncertainty in Articial Intelligence*, pages 43–52, 1998.
5. P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pages 604–613, 1998.
6. D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.
7. E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Ef£cient search for approximate nearest neighbor in high dimensional spaces. *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pages 614–623, 1998.
8. Atsuyoshi Nakamura and Naoki Abe. Collaborative £ltering using weighted majority pr ediction algorithms. In *Proceedings of the Fifteenth International Confere nce on Machine Learning*, 1998.
9. P. Resnick, N. Iacovou, M. Suchak, P. Ber gstrom, and J. Riedl. Grouplens: An open architecture for collaborative £ltering of netnews. In *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, 1994.
10. P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40:56–58, 1997.
11. H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Proceedings of the 1992 Workshop on Computational Learning Theory*, pages 287–294, 1992.