Towards Mobile Query Auto-Completion: An Efficient Mobile Application-Aware Approach

Aston Zhang*[1], Amit Goyal[2], Ricardo Baeza-Yates[2]
Yi Chang[2], Jiawei Han[1], Carl A. Gunter[1], Hongbo Deng[2]
[1]University of Illinois at Urbana-Champaign, IL, USA, [2]Yahoo Labs, CA, USA
{lzhang74, hanj, cgunter}@illinois.edu, rbaeza@acm.org,
{goyal, yichang, hbdeng}@yahoo-inc.com

ABSTRACT

We study the new mobile query auto-completion (QAC) problem to exploit mobile devices' exclusive signals, such as those related to mobile applications (apps). We propose AppAware, a novel QAC model using installed app and recently opened app signals to suggest queries for matching input prefixes on mobile devices. To overcome the challenge of noisy and voluminous signals, AppAware optimizes composite objectives with a lighter processing cost at a linear rate of convergence. We conduct experiments on a large commercial data set of mobile queries and apps. Installed app and recently opened app signals consistently and significantly boost the accuracy of various baseline QAC models on mobile devices.

CCS Concepts

•**Information systems** → **Query intent; Query suggestion; Query reformulation;**

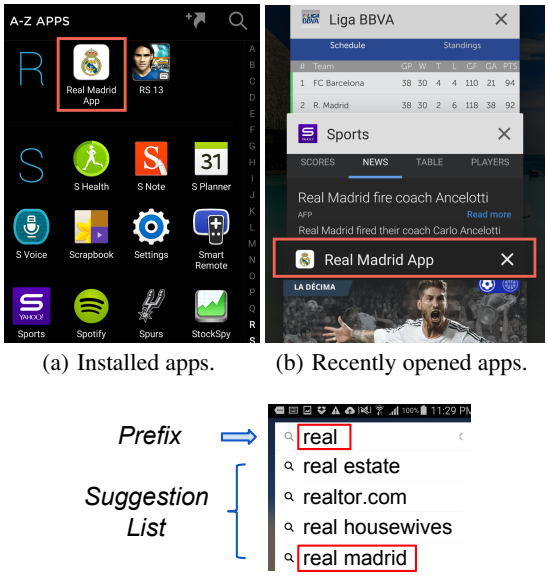
Keywords

Query Auto-Completion; Mobile Application; Mobile Device

1. INTRODUCTION

Query auto-completion (QAC) facilitates user query compositions by suggesting queries given prefixes. Figure 1(c) depicts an example of QAC on mobile devices. Upon a user's keystroke, QAC displays a *suggestion list* (or *list*) below the current *prefix*. Queries in a suggestion list are called *suggested queries* or *query suggestions*. A user can select to submit a suggested query or type to submit a query without selecting any suggestion.

Baeza-Yates *et al.* found that Japan Yahoo Search users generally typed longer queries on mobile devices than desktops to avoid having to query again as mobile internet was slower in 2007 [1]. A report from Microsoft Bing also observed that English queries are generally longer from mobile users than desktop users, and believed that "query auto-suggestion plays an important role" [36]. We further discover that in 2014, global users of Yahoo Search on mobile devices saved more than 60% of the keystrokes when submitting English queries by selecting QAC suggestions. In comparison with such keystroke saving on desktops (~50%) [43], users tend to rely on mobile QAC more heavily. It is probably due to the inconvenience of typing on mobile devices as revealed by Google Search [21]. In fact, users can type 21 words per minute on mobile devices but more than 60 words per minute on desktops [13]. Thus, QAC is even more important to mobile users than desktop users.



(a) Installed apps. (b) Recently opened apps.

(c) Mobile query auto-completion.

Figure 1: A commercial mobile QAC. The *Real Madrid* app is installed and recently opened. Given prefix "real", popular queries on real estate ("real estate" and "realtor.com") are suggested at higher positions than query "real madrid".

Typically, a user favors and submits a query if it reflects the user's query intent in a query composition. Predicting query intents is nontrivial. Most of the recently proposed QAC models rank a list of suggested queries for each prefix according to relevance scores based on various signals, such as popularity-based QAC (historical query frequency count signals) [3], time-based QAC (time signals) [35, 37], context-based QAC (user previous query signals) [3], personalized QAC (user profile signals) [34], or time-and-context-based QAC (both time and user previous query signals) [6]. Note that the aforementioned signals are available on both desktops and mobile devices. Are there any useful signals exclusively exploitable on mobile devices for mobile QAC? Let us look at a few examples.

A mobile application is hereinafter referred to as a *mobile app* or simply as an *app*. Consider a fan of the Real Madrid Football Club

*Part of the work was completed at Yahoo Labs.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2882977.

who installs the *Real Madrid* app on the smart phone. The user opens this app and after a while wants to query "real madrid" to learn more of this club on the web with a popularity-based QAC [3]. When the user types "real", real estate-related queries, such as "real estate" and "realtor.com", are ranked at the top in the suggestion list because they are most popular in historical query logs. Figure 1 displays the user's installed apps, recently opened apps, and a commercial search engine QAC on the same mobile device. Here the user may have implicitly provided the query preference via the installed football club's app. Besides, the user's query intent may also be implied by the recently opened app if the subsequent query interest arises from the app opening. In a large commercial data set, we observe that on mobile devices and matching certain prefixes, users that install the *NBA* app may submit more queries related to basketball teams, and users may query lyrics more often after opening a music app (§2). Being aware of app installation and opening on mobile devices, can QAC be more accurate on mobile devices? Our work answers this question affirmatively.

New Problem, New Challenge. To the best of our knowledge, no existing QAC employs mobile devices' exclusive signals. Hence, our goal is to study the new *mobile QAC* problem: QAC using mobile devices' exclusive signals. We refer to QAC that does not employ any signal exclusive to mobile devices as *Standard QAC*, such as QAC based on popularity and time. Mobile app-related signals are exclusive to mobile devices [2]. The sets of all available applications on desktops and mobile devices are different; even for desktop and mobile versions of the related applications, their contents or interfaces generally differ [17]. Although whether desktop applications can improve QAC is also an open question, we study mobile QAC by exploiting mobile devices' exclusive signals from installed mobile apps and recently opened mobile apps. This is motivated by the importance of mobile QAC.

We model the query–app relationships and the order of recently opened apps before query submissions. It is challenging because such signals are noisy and voluminous. In many cases, a certain installed app may not indicate a higher likelihood of a certain query submission. Besides, even though a certain app opening (*Real Madrid* app) may suggest a higher chance of a certain query ("real madrid"), when another app such as *Realtor.com* is opened more recently before a query, the less recently opened app (*Real Madrid* app) may be less relevant to the query intent. Moreover, even for 1,000 queries and 100 apps, potentially there can be voluminously 100,000 query–app relationship pairs to process.

Our Approach. We go beyond Standard QAC by exploiting signals exclusive to mobile devices. To solve the mobile QAC problem, we propose AppAware, a novel model to employ installed app and recently opened app signals. AppAware reuses the relevance scores of queries from Standard QAC to pre-index top queries. In a single query composition, AppAware re-ranks these top queries based on installed app and recently opened app signals. For these signals, AppAware captures relationships between different mobile queries and apps, and the order of recency for opened apps before query submissions.

To overcome the challenge of noisy and voluminous signals, AppAware optimizes a convex composite objective function by single-stage random coordinate descent with mini-batches. The composite objectives include filtering out noisy signals. When processing voluminous signals, the algorithm has a lighter processing cost at each iteration than either full proximal gradient descent or the gradient update with respect to all coordinates. Importantly, while enjoying a lighter processing cost for voluminous signals and capable of noisy signal filtering, our algorithm converges to the global optimum at a linear rate with a theoretical guarantee.

We make the following contributions:

- We jointly study mobile queries and apps from commercial products (§2). Specifically, we find that going beyond Standard QAC by exploiting installed app and recently opened app signals for mobile QAC is useful. For example, recently opened app signals abound on mobile devices before query submissions.
- We propose a novel AppAware model that exploits installed app and recently opened app signals to solve the mobile QAC problem (§3). To overcome the challenge of noisy and voluminous signals, AppAware optimizes composite objectives by an algorithm using single-stage random coordinate descent with mini-batches. We prove that our algorithm converges to the global optimum at a linear rate with a theoretical guarantee.
- We conduct comprehensive experiments (§4). Among many findings, we show that installed app and recently opened app signals consistently and significantly boost the accuracy of various investigated Standard QAC models on mobile devices.

2. MOBILE QUERY AND APPLICATION

We jointly study mobile query logs and mobile app logs from commercial products at a large scale and discuss our observations.

Terminology. In general, *mobile devices* (*devices*) are handheld computing devices with an operating system where various types of mobile apps can run. Below are other used terms.

Query composition (*Composition*): The duration of composing and submitting a single query. It starts from the keystroke of a new query's first character, or from the keystroke starting to edit a previous query. It ends when a query is submitted. A composition contains information on all keystrokes (with the timestamp of the first keystroke), submitted query, installed apps at the first keystroke time, and recently opened apps with timestamps.

Before query: Before the first keystroke of a query composition.

Mobile log data set: Our jointly collected data set of mobile query logs and mobile app logs from Yahoo. It contains 823,421 compositions sampled from 5 months in 2015. In one composition, all keystrokes (with the timestamp of the first keystroke), the submitted query, installed apps at the first keystroke time, and recently opened apps with timestamps are collected.

Example 1 (Mobile Query and Installed App). Users install apps on mobile devices. Some apps may reflect users' interests or preferences in sports, business, and other fields. Users' interests or preferences exhibited from their installed apps may be relevant to their query intents. Table 1 compares top queries (with percentage) prefixed by "chicago" from all users' mobile devices in the mobile log data set where the *NBA* app is installed (left) or not (right). Among all the mobile queries prefixed by "chicago" submitted from devices installing the *NBA* app, 24% are "chicago bulls" followed by "chicago bears" with a sharp fall in its percentage. However, "chicago bulls" is not among the top 5 mobile queries prefixed by "chicago" on devices without installing the *NBA* app. So, installing the *NBA* app may exhibit users' interests in NBA basketball teams, such as Chicago Bulls (not Chicago Bears). Since the top 4 queries on the left column of Table 1 are sport teams, an NBA fan may generally submit more sports-related queries.

Example 2 (Mobile Query and Recently Opened App). Users open apps to perform activities, such as listening to music. After users open apps, the subsequent query intents may arise from the performed activities through those apps. Table 2 compares top queries (with percentage) prefixed by "sugar" from all users' mobile devices in the mobile log data set where the *Spotify Music* app is opened within 30 minutes before queries (left) or not (right). Four of five top queries on the left column of Table 2 are related

Table 1: Top queries (with percentage) prefixed by "chicago" from all users' mobile devices where the *NBA* app is installed (left) or not (right).

chicago bulls	24%	chicago tribune	11%
chicago bears	12%	chicago weather	10%
chicago cubs	10%	chicago bears	9%
chicago blackhawks	9%	chicago craiglist	9%
chicago tribune	7%	chicago cubs	8%

Table 2: Top queries (with percentage) prefixed by "sugar" from all users' mobile devices where the *Spotify Music* app is opened within 30 minutes before queries (left) or not (right).

sugar maroon 5 lyrics	22%	sugar cookie recipe	13%
sugar lyrics maroon 5	18%	sugar glider	11%
sugar lyrics	14%	sugar bowl	10%
sugar maroon 5	13%	sugar maroon 5 lyrics	10%
sugar daddy	9%	sugar sugar	9%

to the song Sugar by the pop rock band Maroon 5. So, users may tend to search for music-related items, such as lyrics, after opening music apps on mobile devices.

Abundance of Signals. From the two examples above, signals of installed apps and recently opened apps may be useful for boosting the accuracy of mobile QAC. We proceed to study the existence of such app-related signals. The Yahoo Aviate team reported mobile app installation and opening statistics in Table 3. On average, there are 95 installed apps on each mobile device and they are opened 100 times every day. Some apps are opened more than once in a day and on average 35 unique apps are opened per day.

To further investigate opened app signals, there are two interesting open questions: do users open apps before query submissions within a short time? If so, how many unique apps do they open? To answer these questions, we jointly study mobile queries and apps. Figure 2(a) shows the percentage of mobile queries that have non-zero recently opened apps (at least one app is opened within a given time before queries). Specifically, 84.9% of mobile queries belong to the cases where at least one app is opened within 30 minutes before queries. Figure 2(b) shows the average count of unique recently opened apps within a given time before queries (compositions that have no recently opened apps within the time are excluded). Among those 84.9% queries, on average 4.0 unique apps are opened within 30 minutes before queries. Recently opened app signals abound on mobile devices before query submissions.

Recall §1 that mobile QAC is important. Given the observations that app-related signals may imply users' query intents and the abundance of such signals, it is appealing to exploit them for mobile QAC. We propose and discuss an app-aware approach to exploit such signals for mobile QAC in §3.

3. APPLICATION-AWARE APPROACH

For mobile QAC, we propose the AppAware model to exploit installed app and recently opened app signals on mobile devices.

3.1 Design Overview

Before detailing the problem and method, we describe the high-level design of AppAware to rank suggested queries for a given prefix on mobile devices. AppAware has two stages: pre-indexing and re-ranking. A toy example of two suggestions "real estate" and "real madrid" matching prefix "real" is used to describe the idea.

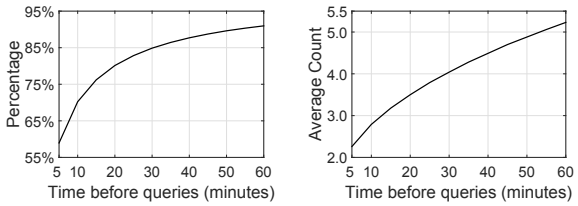
In the pre-indexing stage, given an input prefix, top N query suggestions with the highest relevance scores of Standard QAC are pre-indexed: a higher score gives a higher position. For prefix "real", the top 2 queries "real estate" and "real madrid" are pre-indexed by Standard QAC based on the historical query frequency counts. In the re-ranking stage, AppAware re-ranks these top N queries based on installed app and recently opened app signals in the same query composition. To illustrate, given prefix "real", the pre-indexed queries "real estate" and "real madrid" are instantly fetched. If a user's preference for "real madrid" to "real estate" is inferred from signals of the installed and recently opened *Real Madrid* app, AppAware updates the ranking scores of the two queries. The top 2 queries "real estate" and "real madrid" are re-ranked. With re-ranking, "real madrid" is now at Position 1, higher than the more popular query "real estate".

The number of the pre-indexed top queries N can be set to a small positive integer in a production. Given various display sizes of mobile devices, a smaller number of top queries may be suggested. For a small constant value N, sorting N queries based on the updated ranking scores can be achieved in a constant time [8].

AppAware is designed to reuse existing Standard QAC research in computing the relevance score of a query. It can be available via an existing Standard QAC model, such as a popularity-based QAC. However, AppAware is not constrained to use any certain relevance score: in §4 we evaluate several different relevance scores with different parameter settings in these scores.

Table 3: Mobile app installation and opening statistics according to the Yahoo Aviate team.

Description	Average count
Installed apps per mobile device	95
App opening per day	100
Unique apps that are opened per day	35



(a) Mobile queries (b) Recently opened apps

Figure 2: Recently opened app signals abound on mobile devices before queries. The left figure shows the percentage of mobile queries that have non-zero recently opened apps (at least one app is opened within a given time before queries). The right figure shows the average count of unique recently opened apps within a given time before queries (compositions that have no recently opened apps within the given time are not counted).

3.2 Problem Formulation

Recall §2 that a query composition contains information on all keystrokes (with the timestamp of the first keystroke), the submitted query, installed apps at the first keystroke time, and recently opened apps with timestamps. We assume that signals are the same at all the keystrokes of the same composition. To keep notations unclogged, an AppAware output depends on signals of a certain composition rather than an explicit keystroke of this composition. During composition c, AppAware suggests a ranked list of queries matching a given prefix in query set $\mathcal{Q}$ according to ranking scores determined by a probabilistic model. The probabilistic model is

Table 4: Main notations

Symbol	Description
$a \in \mathcal{A}$	App and app set.
$q \in \mathcal{Q}$	Query and query set
$c \in \mathcal{C}$	Composition and composition set
$q^{(c)}$	Submitted query in composition c
$\mathcal{A}^{(c)}$	Set of installed apps on the device of composition c
$\widetilde{\mathcal{A}}^{(c)}$	Set of recently opened apps in composition c
$\widetilde{a}_k^{(c)}$	k^{th} most recently opened app in composition c
$s(q, c)$	Relevance score of query q that matches a given prefix in composition c
$p(q, c)$	Preference for query q in composition c
$\mathcal{Q}^{(c)}$	Set of top N queries ranked by $s(q, c)$
$\mathbf{w}$	Signal parameter vector
x, y	Signals of installed apps and recently opened apps

based on a combination of the relevance score and app-related signal score on mobile devices. For query q that matches a prefix in composition c, the relevance score of q is denoted as $s(q, c)$. In a composition, installed app and recently opened app signals are represented by x and y. The app-related signal score is based on x and y, and their associated signal parameters β. A collection of β form the signal parameter vector $\mathbf{w}$. This is for indexing convenience in our technical discussions: subscripts of β correspond to queries, apps, and recency orders (§3.3), while subscripts of w locate elements in vector $\mathbf{w}$ (§3.4 and §3.5). The goal is to compute $\mathbf{w}$ by an optimization algorithm. Table 4 briefly summarizes the main notations. Some of them are described in §3.3.

3.3 Likelihood Function

To compute the signal parameter vector $\mathbf{w}$, we need a likelihood function integrating signals and $\mathbf{w}$.

As discussed in §2, installed apps may reflect users' interests or preferences. However, even if two different users both install the same app, their interests or preferences related to that app may still be at different levels. For example, one may like the app, while the other may dislike it but forget to remove it. We cannot directly observe these and we resort to the opening frequency of apps. Intuitively, more frequently opened apps may be more likely related to users' interests or preferences. For example, consider one user who opens the *Real Madrid* app every day and the other who almost never opens it after installation. The former user is more likely interested in the Real Madrid football club than the latter. Besides, suppose that different users install the same app of the same level of interests at different time. A user more likely has a higher app opening frequency aggregated from a longer app installation history. In light of this, daily opening frequency can be used for comparison. An installed app signal $x(a, c)$ with respect to app a in composition c is the average daily opening frequency of app a on the mobile device of composition c.

Note that recently opened apps in a composition are already opened by users. Recall the assumption that app openings may reflect users' interests or preferences related to the apps, signals of recently opened apps are directly built in relation to submitted queries in the same composition. So, a recently opened app signal $y(q, a)$ with respect to query q and app a is computed based on the training data set. It is the proportion of the count of q to the count of all queries for all compositions where a is a recently opened app.

Let $\mathcal{A}^{(c)}$ be the set of installed apps on the device of composition c, and $\widetilde{\mathcal{A}}^{(c)} = \{\widetilde{a}_1^{(c)}, \widetilde{a}_2^{(c)}, \ldots\}$ of size $|\widetilde{\mathcal{A}}^{(c)}|$ be the set of unique recently opened apps in composition c, where $\widetilde{a}_k^{(c)}$ is the k^{th} most recently opened app in c. If an app is opened more than once in the same composition, only the most recent one is included in $\widetilde{\mathcal{A}}^{(c)}$. We model preference $p(q, c)$ for query q in composition c by a generalized additive model [14]:

$$p(q, c) = s(q, c) + \sum_{a \in \mathcal{A}^{(c)}} \beta_{q,a} \log\left[1 + x(a, c)\right] + \sum_{k=1}^{|\widetilde{\mathcal{A}}^{(c)}|} \beta_k y(q, \widetilde{a}_k^{(c)}), \tag{3.1}$$

where $\beta_{q,a}$ and β_k are signal parameters. Note that every $\beta_{q,a}$ corresponds to a query–app pair for all $q \in \mathcal{Q}$ and $a \in \mathcal{A}$, where $\mathcal{Q}$ and $\mathcal{A}$ are the sets of queries and apps in the training data set. Signal parameter β_k is only related to recency order k for app opening in any composition. Values of signals x and y are pre-computed in parallel and stored distributively in a Hadoop MapReduce framework. Such values are directly fetched in training and testing without re-computing. The logarithm transformation of daily opening frequency in (3.1) is to dampen the effect of a higher frequency.

In general, the preference model $p(q, c)$ in (3.1) reflects a user's preference for query q in composition c in conjunction with installed app signals and recently opened app signals. The signal parameters $\beta_{q,a}$ and β_k are to be inferred based on maximizing the likelihood of submitted queries, together with those integrated app-related signals observed from the training data set. In order to infer such parameters, we define a likelihood function for a submitted query $q^{(c)}$ in c with a softmax function that represents a smoothed version of the "max" function [5, 41]:

$$\mathbb{P}(q^{(c)} \mid c) = \frac{\exp\left[p(q^{(c)}, c)\right]}{\sum_{q \in \mathcal{Q}^{(c)} \bigcup \{q^{(c)}\}} \exp\left[p(q, c)\right]}, \tag{3.2}$$

where $\mathcal{Q}^{(c)}$ represents the set of top N queries ranked by relevance score $s(q, c)$. Its union with $q^{(c)}$ ensures proper normalization. Likewise, AppAware predicts the likelihood that any query $q' \in \mathcal{Q}^{(c)}$ to be submitted in composition c by

$$\mathbb{P}\left(q' \mid c\right) = \frac{\exp\left[p(q', c)\right]}{\sum_{q \in \mathcal{Q}^{(c)}} \exp\left[p(q, c)\right]}. \tag{3.3}$$

After signal parameters are inferred, in practice, the simpler term $p(q', c)$ in (3.3) is used for re-ranking the pre-indexed query suggestions as described in §3.1. Since query suggestions are pre-indexed by relevance score s, the re-ranking stage of AppAware is determined by app-related signals in composition c, which are captured by the last two terms of (3.1). We emphasize that, the preference model $p(q, c)$ in (3.1) is not constrained to employ any certain relevance score s. We evaluate different settings of s in §4.

Challenges. App-related signals are noisy. On one hand, for many query–app pairs, a certain installed app may not indicate a higher likelihood of a certain query submission. On the other hand, a less recently opened app may be less relevant to the query intent at the time of a query submission. To overcome the challenge of noisy signals, AppAware optimizes composite objectives with filtering out noisy signals. We describe such composite objectives in §3.4.

Besides, app signals are voluminous. Recall that signal parameter $\beta_{q,a}$ captures relationships between every query and installed app in the training data set. The number of such parameters can be as large as the product of unique query count and unique app count (20 million in our experiments) plus the maximum count of unique recently opened apps (48 in our experiments within 30 minutes before queries). Hence, processing with respect to all these parameters simultaneously consumes computational resources heav-

ily. To overcome the challenge of voluminous signals, we describe an algorithm to compute lightly with respect to a random signal parameter at each step in §3.5.

3.4 Composite Objectives

As mentioned in §3.2, for indexing convenience all the signal parameters $\beta_{q,a}$ and β_k from (3.1) in any fixed order constitute the signal parameter vector $\mathbf{w}$. Let w_j be the j^{th} element of vector $\mathbf{w}$ of dimension d. We denote the ℓ_1 and ℓ_2 norms of vector $\mathbf{w}$ as $\|\mathbf{w}\|_1 = \sum_{k=1}^{d} |w_k|$ and $\|\mathbf{w}\|_2 = \left(\sum_{k=1}^{d} w_k^2\right)^{1/2}$.

Signal parameter vector $\mathbf{w}$ is to be inferred based on maximum likelihood. To begin with, we want to maximize the following log-likelihood for the set of compositions $\mathcal{C}$ in the training data set with respect to signal parameters:

$$\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \log \mathbb{P}(q^{(c)} \mid c), \tag{3.4}$$

where $|\mathcal{C}|$ is the size of $\mathcal{C}$ and $\mathbb{P}(q^{(c)} \mid c)$ is defined in (3.2). By (3.2) and (3.4), an unconstrained optimization problem out of minimizing negative log-likelihood with the ℓ_1 and ℓ_2 norms is obtained:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \Big[\log \sum_{q \in \mathcal{Q}^{(c)} \bigcup \{q^{(c)}\}} \exp\big[p(q,c)\big] - p(q^{(c)}, c) \Big] \\ & + \frac{\lambda_2}{2} \|\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1, \end{aligned} \tag{3.5}$$

where λ_2 and λ_1 are regularizer weights of ℓ_2 and ℓ_1 norms. Recall that $\beta_{q,a}$ and β_k of $p(q,c)$ in (3.1) correspond to $\mathbf{w}$. In (3.5), the main purpose of introducing the ℓ_2 norm with $\lambda_2 > 0$ is to guarantee the strong convexity of the objective function in (3.5) excluding the last term. We denote the convexity parameter by μ. The ℓ_1 norm is for filtering out noisy signals, which is discussed in detail in §3.5.1 (Remark 3.1). Rewriting (3.5) in the form of a sum of a finite number of functions gives the composite objective problem:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, F(\mathbf{w}) + R(\mathbf{w}), \tag{3.6}$$

where $F(\mathbf{w}) = (1/|\mathcal{C}|) \sum_{c \in \mathcal{C}} f_c(\mathbf{w})$ and $R(\mathbf{w}) = \sum_{j=1}^{d} r_j(\mathbf{w})$, where $f_c(\mathbf{w}) = \log \sum_{q \in \mathcal{Q}^{(c)} \bigcup \{q^{(c)}\}} \exp\big[p(q,c)\big] - p(q^{(c)}, c) + (\lambda_2/2)\|\mathbf{w}\|_2^2$ and $r_j(\mathbf{w}) = r_j(w_j) = \lambda_1 |w_j|$. Gradient $\nabla F(\mathbf{w})$ is Lipschitz continuous and we denote the Lipschitz constant by L. Same as $F(\mathbf{w})$, which is the objective function in (3.5) excluding the last term, each function $f_c(\mathbf{w})$ is strongly convex with convexity parameter μ. Note that $F(\mathbf{w})$ is a sum of a finite number of strongly convex and smooth functions and $R(\mathbf{w})$ is a general convex function that is non-differentiable. Each element function $f_c(\mathbf{w})$ is a negative log-likelihood function with the ℓ_2 norm for composition c, which is a single element of set $\mathcal{C}$.

3.5 Optimization

There are a few issues with optimizing the composite objectives in (3.6). Due to the large size of the training data set, an algorithm based on proximal stochastic gradient descent is preferred. However, this has a slower sublinear rate of convergence. Recently, Schmidt *et al.* trained conditional random fields using the stochastic average gradient with a faster linear rate of convergence [32]. In fact, there is another linearly-convergent stochastic variance reduced gradient that has multiple stages with two nested for-loops per iteration [20]. Such a multi-stage algorithm requires a pass through the entire data set per iteration, which is computationally expensive especially when the data set is large. In sharp contrast, the gradient update method by Schmidt *et al.* has a simpler single-stage iteration with only one for-loop and avoids the aforementioned computational complexity from a multi-stage algorithm.

We propose an optimization algorithm in §3.5.1 employing the single-stage stochastic average gradient from Schmidt *et al.* [32]. We highlight that their algorithm cannot be directly applied to solve (3.6), and our algorithm is distinct from theirs in two main aspects. First, the noisy signal challenge is addressed by optimizing composite objectives with non-differentiable $R(\mathbf{w})$ (details are in Remark 3.1), which can be solved by our algorithm but not their algorithm. Second, to overcome the voluminous signal challenge, our algorithm updates the gradient with respect to only one coordinate per iteration while their algorithm updates the gradient with respect to all coordinates at each iteration. We theoretically guarantee the linear rate of convergence for our algorithm with different proof techniques from those of Schmidt *et al.*

3.5.1 Algorithm

First, initialize signal parameter vector $\mathbf{w}^{(0)}$ at random. Then, for iteration $t = 1, 2, \ldots$, repeat the following:

I Sample mini-batch $\mathcal{B}$ from $\{1, \ldots, |\mathcal{C}|\}$ uniformly at random with replacement.

II Set element signal parameter vector $\boldsymbol{\phi}_c^{(t)}$ to common signal parameter vector $\mathbf{w}^{(t-1)}$ for all $c \in \mathcal{B}$.

III Sample coordinate index j from $\{1, \ldots, d\}$ uniformly at random with replacement.

IV Compute the updated gradient based on the sampled mini-batch with respect to the sampled coordinate

$$g_{\mathcal{B},j}^{(t)} = \nabla_j f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t)}) - \nabla_j f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t-1)}) + \frac{1}{|\mathcal{C}|} \sum_{k \in \mathcal{C}} \nabla_j f_k(\boldsymbol{\phi}_k^{(t-1)}), \tag{3.7}$$

where by defining $|\mathcal{B}|$ as the size of mini-batch $\mathcal{B}$, for all $\boldsymbol{\phi}$, $f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}) = (1/|\mathcal{B}|) \sum_{c \in \mathcal{B}} f_c(\boldsymbol{\phi}_c)$ and $\nabla_j f(\boldsymbol{\phi}) = [\nabla f(\boldsymbol{\phi})]_j = \partial f(\boldsymbol{\phi}) / \partial \phi_j$.

V Set $w_j^{(t)}$ to $\operatorname{prox}_{\eta,j}(w_j^{(t-1)} - \eta g_{\mathcal{B},j}^{(t)})$, where for all $w, u \in \mathbb{R}$,

$$\operatorname{prox}_{\eta,j}(w) = \underset{u}{\operatorname{argmin}} \frac{1}{2\eta} \|w - u\|_2^2 + r_j(u). \tag{3.8}$$

VI Set $\mathbf{w}_{\setminus j}^{(t)}$ to $\mathbf{w}_{\setminus j}^{(t-1)}$, where any subvector of $\mathbf{w}$ excluding w_j is denoted by $\mathbf{w}_{\setminus j}$.

REMARK 3.1 (FILTERING OUT NOISY SIGNALS). *The proximal operator in* (3.8) *facilitates proof of linear convergence. Without it, a subgradient method only gives a sublinear rate of convergence. There is a closed-form solution to* (3.8)*. We emphasize that this solution may clear certain signal parameter values to 0: if* $|w_j^{(t-1)} - \eta g_{\mathcal{B},j}^{(t)}| \leq \eta\lambda_1$, $w_j^{(t)} = 0$*; otherwise* $w_j^{(t)} = w_j^{(t-1)} - \eta g_{\mathcal{B},j}^{(t)} - \eta\lambda_1 (w_j^{(t-1)} - \eta g_{\mathcal{B},j}^{(t)}) / |w_j^{(t-1)} - \eta g_{\mathcal{B},j}^{(t)}|$*. Signal parameters of value 0 indicate that their corresponding noisy signals are filtered out in* (3.1)*.*

REMARK 3.2 (LIGHTER COST FOR VOLUMINOUS SIGNALS). *Given the voluminous app-related signals in the training data set (about 20 million in our experiments), updating the gradient of the signal parameter vector with respect to all coordinates consumes computational resources heavily per iteration, such as exceeding the memory budget. Our algorithm enjoys a lighter processing cost than either batch-style proximal gradient descent or any gradient update with respect to all coordinates per iteration. The update at each iteration of the algorithm is based on a mini-batch of element functions with only one coordinate. With a lighter processing cost, this algorithm converges to the global optimum at a linear rate.*

3.5.2 Computational Complexity

Multi-stage algorithms with multiple loops for each iteration requires a pass through the entire data set per iteration [20]. To avoid this high computational complexity, our algorithm is based on a single-stage update with only one loop through $t = 1, 2, \ldots$ [32]. To compare these two techniques for updating the gradient, suppose that both algorithms update the gradient with respect to the same number of element functions and coordinates. At each iteration, the inner loop of the multi-stage algorithm involves a repetitive computation of $\mathcal{O}(|\mathcal{C}|)$ time, where $|\mathcal{C}|$ is the size of the data set (number of element functions). In contrast, the single-stage algorithm requires a computation of $\mathcal{O}(1)$ time per iteration: the last term in (3.7) is a distributive function and its update takes a constant time without a need for re-computation at each iteration. For the same problem setting, the iteration complexity of the single-stage algorithm is lower than that of the multi-stage algorithm [32, 20].

In addition, it is notable that at each single-stage iteration, the update in (3.7) reduces the variance of the gradient estimator at each iteration with the stochastic average gradient. This results in a faster linear rate of convergence than a sublinear rate of the classic proximal stochastic gradient descent. We theoretically guarantee the linear rate of convergence in §3.5.3. Our empirical results in §4.3.2 reinforce that with 15 entire data passes, the objective gap value is close to 10^{-4}. Here an entire data pass is a standard measure representing the least possible iterations for passing through the entire data instances with respect to all coordinates [32, 20]. Given $|\mathcal{C}|$ compositions with d coordinates, one entire data pass of the algorithm in §3.5.1 is equivalent to $(|\mathcal{C}|d)/|\mathcal{B}|$ iterations, where $|\mathcal{B}|$ is the mini-batch size in the algorithm.

3.5.3 Optimum and Convergence

It is easy to conclude that, the global optimum $\mathbf{w}^*$ exists for the composite objective optimization problem in (3.6) because $F(\mathbf{w})$ is strongly convex and $R(\mathbf{w})$ is convex.

However, the theoretical analysis for the rate of convergence of the algorithm is nontrivial. In this subsection and Appendix A, all the expectations are taken conditional on $\mathbf{w}^{(t-1)}$ and $\boldsymbol{\phi}_c^{(t-1)}$ unless otherwise stated. For the convenience of our analysis, based on (3.7), after removal of the coordinate index we define

$$\mathbf{h}_{\mathcal{B}}^{(t)} = \nabla f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t)}) - \nabla f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t-1)}) + \frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \nabla f_k(\boldsymbol{\phi}_k^{(t-1)}), \tag{3.9}$$

$$\mathbf{h}_{c}^{(t)} = \nabla f_{c}(\boldsymbol{\phi}_{c}^{(t)}) - \nabla f_{c}(\boldsymbol{\phi}_{c}^{(t-1)}) + \frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \nabla f_k(\boldsymbol{\phi}_k^{(t-1)}), \tag{3.10}$$

where $\mathcal{B}$ is a mini-batch uniformly sampled from $\{1, \ldots, |\mathcal{C}|\}$ at random with replacement and $c \in \mathcal{C}$. Before we prove the rate of convergence, we introduce two important lemmas.

LEMMA 3.3. *For the algorithm in §3.5.1, with definitions in* (3.9) *and* (3.10) *we have*

$$\mathbb{E}_{\mathcal{B}}[\mathbf{h}_{\mathcal{B}}^{(t)}] = \mathbb{E}_c[\mathbf{h}_c^{(t)}] = \nabla F(\mathbf{w}^{(t-1)}).$$

The proof is in Appendix A.1. Lemma 3.3 guarantees that $\mathbf{h}_{\mathcal{B}}^{(t)}$ is an unbiased gradient estimator of F.

Recall that the algorithm in §3.5.1 samples a mini-batch of compositions uniformly at random with replacement at every iteration. To facilitate evaluation of expectation terms with respect to randomly sampled mini-batches of compositions, we introduce the following lemma.

LEMMA 3.4. *For the algorithm in §3.5.1 and for all* $\mathbf{x}$ *and* $\mathbf{y}$,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\mathbf{x}) - \nabla f_{\mathcal{B}}(\mathbf{y})\|^2\big] \\
&= \frac{|\mathcal{B}|\cdot|\mathcal{C}| - |\mathcal{C}|}{|\mathcal{B}|\cdot|\mathcal{C}| - |\mathcal{B}|}\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|^2 \\
&\quad + \frac{|\mathcal{C}| - |\mathcal{B}|}{|\mathcal{B}|\cdot|\mathcal{C}| - |\mathcal{B}|}\mathbb{E}_c\big[\|\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\|^2\big].
\end{aligned}$$

Lemma 3.4 is proved in Appendix A.2. Now we present the main theory for bounding the rate of convergence.

THEOREM 3.5. *The algorithm in §3.5.1 is able to converge to the optimal solution at a linear rate.*

We give the detailed proof in Appendix A.3. The empirical results in §4.3.2 agree with our theory that the optimization algorithm converges to the global optimum at a linear rate.

4. EVALUATION

We comprehensively evaluate the proposed mobile QAC model, AppAware, on a large real-world commercial data set.

4.1 Data Description

We describe important details of our collected mobile log data set. Due to the proprietary nature of the data, some details are omitted. The mobile log data set is sampled among 5 months in 2015 and from mobile devices with the Android operating system. All queries are submitted via the search bar of the *Yahoo Aviate* homescreen in Figure 1(c). One million compositions are randomly sampled, then tail queries and apps are filtered out: the most popular 10,000 unique queries and most installed 2,000 unique apps (excluding the *Yahoo Aviate* homescreen) remain. The final data set contains 823,421 compositions. In one composition, all keystrokes (with the timestamp of the first keystroke), the submitted query, installed apps at the first keystroke time, and recently opened apps with timestamps are collected. The maximum count of unique recently opened apps within 30 minutes before queries is 48.

The training and testing data sets are split in an ascending time order: the first and second half of a user's compositions are used for training and testing respectively. All the app-related signals and the relevance scores are standardized: the data standardization procedure is transforming data to zero mean and unit variance.

4.2 Experimental Setting

Measures for Accuracy. Mean reciprocal rank (MRR) is a standard measure to evaluate the ranking accuracy of QAC [3, 24, 19, 34, 43]. It is calculated by the average reciprocal of the submitted query's ranking in a suggestion list. Success Rate@top k (SR@k) is the average percentage of the submitted queries that can be found in the top k suggestions during testing. SR@k is also used to evaluate the QAC ranking accuracy [19, 43]. In general, a higher MRR or SR@k indicates a higher ranking accuracy of QAC [3, 24, 19, 34, 6, 43]. The statistical significance of the accuracy improvements is validated by a paired-t test ($p < 0.05$).

Methods for Comparison. The relevance scores with parameter settings in our experiments reuse the existing research as described below. None of these baseline methods uses mobile devices' exclusive signals. Thus, they are referred to as Standard QAC.

- **MPC:** Given an input prefix, Most Popular Completion (MPC) ranks suggested queries based on their historical query frequency counts. A more popular query has a higher rank. It was found competitive by various studies [3, 19, 24, 34].

Table 5: Accuracy comparison of Standard QAC and AppAware (in percentage). All the boldfaced results denote that the accuracy improvements over Standard QAC are statistically significant ($p < 0.05$) for the same relevance score.

Relevance	MRR		SR@1		SR@2		SR@3	
	Std.	AppAware	Std.	AppAware	Std.	AppAware	Std.	AppAware
MPC	35.13	**41.55 (+18.27%)**	27.36	**34.08 (+24.56%)**	37.09	**44.50 (+19.98%)**	41.69	**48.61 (+16.60%)**
Personal	39.06	**43.57 (+11.55%)**	31.32	**37.16 (+18.65%)**	40.52	**46.36 (+14.41%)**	46.21	**50.15 (+8.53%)**
Personal-S	40.48	**44.62 (+10.23%)**	32.70	**38.69 (+18.32%)**	42.53	**47.54 (+11.78%)**	47.53	**50.62 (+6.50%)**
TimeSense	39.91	**43.94 (+10.10%)**	32.79	**38.48 (+17.35%)**	42.10	**46.91 (+11.43%)**	46.83	**49.45 (+5.59%)**
TimeSense-S	40.88	**44.93 (+9.91%)**	34.01	**39.98 (+17.55%)**	43.76	**47.58 (+8.73%)**	47.66	**50.12 (+5.16%)**

*Std.: Standard QAC

- **Personal:** Personal QAC by distinguishing different users can achieve a higher accuracy [3, 6, 34]. Here the Personal relevance score is an equal-weighted linear combination of the MPC score and the standardized personal historical query frequency counts as suggested by a study [43].
- **Personal-S:** It is the Personal relevance score with an optimal combination with different weights of the MPC score and the standardized personal query frequency counts. Optimal weights achieving the highest MRR makes Personal-S more competitive.
- **TimeSense:** Time signals are useful in QAC [6, 35, 37]. TimeSense is the same as Personal except that the personal historical query frequency count is replaced by the frequency count of a query from all users within 28 days before a composition [37].
- **TimeSense-S:** It is the same as Personal-S except that the Personal score is replaced by the TimeSense score.

We study the effect of varying parameter values in §4.3. Unless otherwise stated, the time-window size for recently opened apps before query submissions is 30 minutes, the mini-batch size is 100, the pre-indexed query count is 10, the suggested query count is 5 (considering display sizes of mobile devices), and the number of entire data passes is 15. Personal-S and TimeSense-S both linearly combine a MPC score with the optimal weight θ and the other score with the weight $1 - \theta$. The optimal weights in Personal-S and TimeSense-S enable Standard QAC to achieve the highest MRR.

4.3 Experimental Results

We perform comprehensive experiments to evaluate the performance of the proposed AppAware model. We first compare methods employing different relevance scores in §4.3.1. Then throughout the remaining §4.3.2—4.3.7, we study different general properties of AppAware by fixing the relevance score to MPC; the results with the other relevance scores are similar.

4.3.1 Boosting the Accuracy of Standard QAC with App-related Signals on Mobile Devices

Table 5 presents the accuracy comparison of Standard QAC and AppAware with different relevance scores as described in §4.2. All the boldfaced results denote that the accuracy improvements over Standard QAC are statistically significant ($p < 0.05$) for the same relevance score. We highlight that, for each same relevance score, mobile devices' exclusive signals of installed apps and recently opened apps significantly and consistently boost the accuracy of these Standard QAC models that do not use exclusive signals of mobile devices. For instance, for the same MPC relevance score, signals of installed apps and recently opened apps significantly boost Standard QAC by 18.27% in MRR. Such an improvement is significant across all the different accuracy measures.

When relevance scores become more accurate, such as Personal and TimeSense in comparison with MPC, AppAware also ranks query suggestions more accurately. Given the relevance scores with different parameter settings (Personal *vs.* Personal-S and TimeSense *vs.* TimeSense-S), AppAware has slightly varying accuracy. Such variance depends on the accuracy of the relevance scores for the chosen parameter values. We conclude that, installed app and recently opened app signals are useful in boosting the accuracy of such existing Standard QAC models on mobile devices.

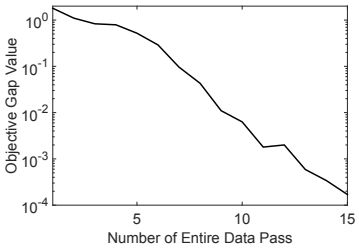


Figure 3: Convergence study.

4.3.2 Convergence Study

In §3.5.3 we theoretically prove that the rate of convergence for AppAware is linear. Our theory is reinforced by the experimental results averaged over 50 replications in Figure 3. The objective gap value is $[F(\mathbf{w}) + R(\mathbf{w})] - [F(\mathbf{w}^*) + R(\mathbf{w}^*)]$ in log scale, where $F(\mathbf{w}) + R(\mathbf{w})$ are the composite objectives and $\mathbf{w}^*$ is the global optimum in (3.6). Recall the definition of the entire data pass in §3.5.2, AppAware converges fast by using the single-stage randomized coordinate descent with mini-batches. With iterations of 15 entire data passes, the objective gap value is close to 10^{-4}.

4.3.3 Varying-Length Prefix Study

We study the performance of AppAware and Standard QAC for prefixes with varying lengths. We group prefixes into five bins according to their lengths in characters. The ranking accuracy of AppAware and Standard QAC is evaluated on prefixes from the same bin. Figure 4 illustrates the ranking accuracy comparison of AppAware and Standard QAC for prefixes of varying lengths. It is interesting to observe that accuracy improvements by app-related signals are not constant with respect to varying-length prefixes.

In general, when prefixes are shorter, the accuracy gap between AppAware and Standard QAC is larger across different accuracy measures. So, installed app and recently opened app signals take better effect in boosting accuracy of Standard QAC when handling more challenging scenarios of shorter input prefixes. This may be explained by the declining challenges for longer prefixes due to a reduction of the matched queries: Standard QAC is more accurate for such cases and it is harder to make further improvements.

4.3.4 App-Related Signal Study

AppAware makes use of two types of exclusive signals to mobile devices: installed apps and recently opened apps. To more

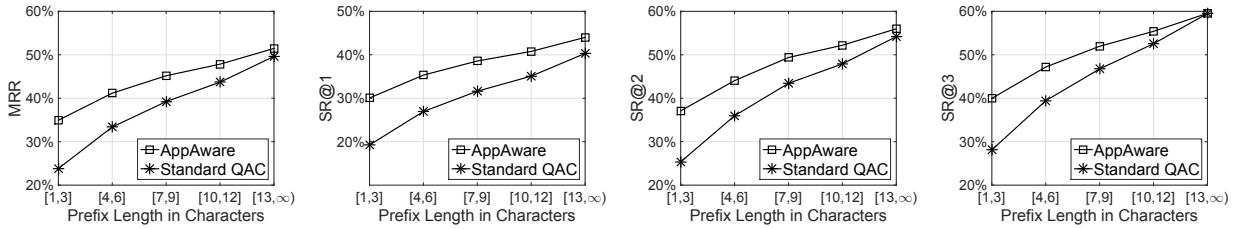


Figure 4: Accuracy comparison of AppAware and Standard QAC for prefixes with varying lengths.

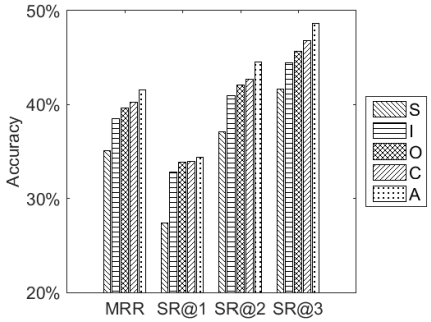


Figure 5: AppAware achieves the highest accuracy in comparison with its variants (S: Standard QAC; I: AppAware variant using installed app signals only; O: AppAware variant using recently opened app signals only; C: AppAware "case-by-case" variant using recently opened app signals only when they exist, otherwise using installed app signals only; A: AppAware).

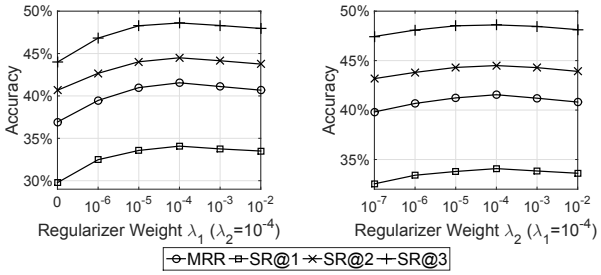


Figure 6: Regularizer weight study.

comprehensively study such signals, we compare two variants of AppAware using different subsets of such signals: installed app signals only and recently opened app signals only. In addition, we introduce another "case-by-case" variant: it uses recently opened app signals only when they exist, otherwise uses installed app signals only. The results are compared in Figure 5.

Although both types of signals are able to improve the ranking accuracy of Standard QAC alone, recently opened app signals are slightly better at predicting query intents than installed app signals on mobile devices. Since recently opened app signals do not always exist, the "case-by-case" variant is slightly more accurate than the variant using recently opened apps only. When recently opened app signals exist, the "case-by-case" variant uses such signals only; while AppAware integrates extra installed app signals. To illustrate, even though some apps are recently opened before query submissions, these queries may still be related to installed app signals only or both types of signals. Being capable of modeling all such potential scenarios, AppAware achieves the highest accuracy across different measures in comparison with its variants.

4.3.5 Regularization Study

Figure 6 plots the accuracy measures of AppAware with varying regularizer weights λ_1 (left) and λ_2 (right). We vary the value of one regularizer weight while fixing that of the other at 10^{-4}.

It is noteworthy from Figure 6 (left) that the accuracy is highest when $\lambda_1 = 10^{-4}$ but degrades sharply when $\lambda_1 = 0$. It empirically corroborates the effect of the ℓ_1 norm in filtering out noisy signals. When λ_1 gets smaller than 10^{-4}, the accuracy is lower due to a lighter penalty applied to signal parameters associated with noisy signals. However, when λ_1 is greater than 10^{-4}, a heavier penalty may suppress useful signals and result in a slightly lower accuracy.

Recall §3.4 that λ_2 must be positive to ensure the strong convexity of $F(\mathbf{w})$ in (3.6) to guarantee the linear convergence of the optimization algorithm. In Figure 6 (right), the highest accuracy is attained when $\lambda_2 = 10^{-4}$. Note that the accuracy for varying λ_1 and λ_2 is stable around the optimum 10^{-4}, such as between 10^{-5} and 10^{-3}. This eases parameter tuning.

4.3.6 Pre-Indexed Query Count Study

Figure 7 (left) illustrates the growing accuracy of AppAware with more pre-indexed queries for re-ranking. This is because fewer pre-indexed queries may exclude users' potential submissions. However, re-ranking more queries is computationally more expensive. Several studies showed that re-ranking 10 pre-indexed queries is feasible in practice [34, 43] and the outperforming of AppAware is obtained with the pre-indexed query count set to 10 in §4.3.1.

4.3.7 Opened App Recency Study

Figure 7 (right) plots the accuracy measures of AppAware when recently opened apps come from time-windows of varying sizes before query submissions. The regularizer weights are optimal for achieving the highest MRR. On one hand, when the time-window size is smaller, all the accuracy measures are consistently lower because useful recently opened app signals are fewer. On the other hand, when its size gets larger, such as larger than 30 minutes, some measures rise slightly while some other ones start to fall. To explain, for those apps that are opened less recently, they may be less relevant to the query intents at the time of query submissions.

5. RELATED WORK

QAC has received a growing attention in recent years, such as popularity-based QAC using historical frequency count signals [3], time-based QAC using time signals [35, 37], context-based QAC using user previous query signals [3], and personalized QAC using user profile signals [34]. The relevance scores evaluated in this work make use of the existing research, such as MPC [3, 19, 24, 34], Personal(-S) [3, 6, 34], and TimeSense(-S) [6, 35, 37, 29].

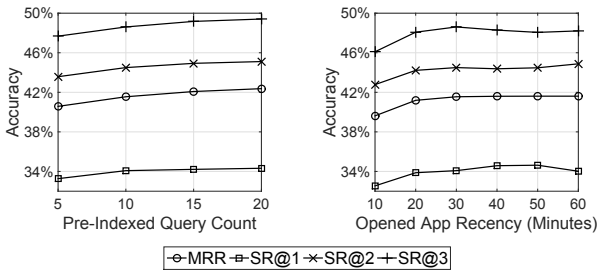


Figure 7: Pre-indexed query count (left) and opened app recency (right) studies.

More recent QAC methods also predicted the likelihood that suggested queries would be selected by users based on keystroke behaviors during query compositions [24, 43, 23], determined suggestion rankings based on query reformulation signals [19], exploited web content signals [22], or combined signals such as time and previous queries from users [6]. Specifically, Zhang *et al.* proposed adaQAC, an adaptive QAC model incorporating users' implicit negative feedback [43]. Other aspects of QAC have also been studied, such as user interactions with QAC [28, 15], space efficient indexing [16], and spelling error tolerance [7, 18, 12, 38]. However, none of the aforementioned work aimed at specifically solving the mobile QAC problem by exploiting mobile devices' exclusive signals. We take the initiative to show that mobile QAC can be more accurate by employing mobile app-related signals.

The idea of using mobile app-related signals for mobile QAC is inspired by a recent mobile app usage prediction work of Baeza-Yates *et al.* [2]. Their model used signals of relations between sequentially opened apps via the Android API. Our work answers an important open question on whether sequentially submitted queries and opened apps can boost the QAC accuracy on mobile devices.

Mobile app recommendation and usage were also studied with respect to app replacement behaviors [42], security preferences [44, 27], version descriptions [26], personalized signal discovery [25], implicit feedback [11], serendipitous apps [4], and many other aspects [9, 10, 33, 39, 40]. A joint research of both mobile queries and mobile apps sets our work apart from these studies.

6. CONCLUSION AND DISCUSSION

Users tend to rely on QAC more heavily on mobile devices than on desktops. Motivated by its importance, we studied the new mobile QAC problem to exploit mobile devices' exclusive signals. We proposed a novel AppAware model employing installed app and recently opened app signals. To overcome the challenge of such noisy and voluminous signals, AppAware optimizes composite objectives at a lighter processing cost. Our algorithm converges to the global optimum at a linear rate with a theoretical guarantee. Experiments demonstrated high efficiency and effectiveness of AppAware.

Our study has provided a number of new insights that we hope will have general applicability to recommendation and search strategies on mobile devices (*e.g.,* mobile shopping and mobile search), to future models of mobile QAC, and to efficient optimization.

Acknowledgements. Research was sponsored in part by NSF grants 09-64392, 12-23967, 13-30491, IIS-1017362, IIS-1320617, IIS-1354329, and HDTRA1-10-1-0120, U.S. Army Research Lab under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

APPENDIX

A. THEORETICAL ANALYSIS

We provide the proof for all the lemmas and theorems (see Section 3) in this appendix.

A.1 Proof of Lemma 3.3

PROOF. We start by analyzing the first two terms in (3.9). For all $\mathbf{w}$ we have $\mathbb{E}_{\mathcal{B}}\left[\nabla f_{\mathcal{B}}(\mathbf{w})\right] = \mathbb{E}_{\mathcal{B}}\big[(1/|\mathcal{B}|)\sum_{c\in\mathcal{B}}\nabla f_c(\mathbf{w})\big]$.

By switching the order of selection in formulating mini-batches, we take expectation with respect to mini-batches and obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{B}}\left[\nabla f_{\mathcal{B}}(\mathbf{w})\right] &= \frac{1}{|\mathcal{B}|\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{i=1}^{\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{c\in\mathcal{B}_i}\nabla f_c(\mathbf{w})\\ &= \frac{1}{|\mathcal{B}|\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{c\in\mathcal{C}}\binom{|\mathcal{C}|-1}{|\mathcal{B}|-1}\nabla f_c(\mathbf{w})\\ &= \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\nabla f_c(\mathbf{w}).\end{aligned}$$

For all $\mathbf{w}$, it holds that $\mathbb{E}_{\mathcal{B}}\left[\nabla f_{\mathcal{B}}(\mathbf{w})\right] = \mathbb{E}_c\left[\nabla f_c(\mathbf{w})\right] = \nabla F(\mathbf{w})$. By the definition of $\mathbf{h}_{\mathcal{B}}^{(t)}$ and $\mathbf{h}_c^{(t)}$ in (3.9) and (3.10),

$$\begin{aligned}&\mathbb{E}_{\mathcal{B}}[\mathbf{h}_{\mathcal{B}}^{(t)}] = \mathbb{E}_c[\mathbf{h}_c^{(t)}]\\ &= \mathbb{E}_c\left[\nabla f_c(\boldsymbol{\phi}_c^{(t)}) - \nabla f_c(\boldsymbol{\phi}_c^{(t-1)})\right] + \frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}}\nabla f_k(\boldsymbol{\phi}_k^{(t-1)})\\ &= \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\nabla f_c(\mathbf{w}^{(t-1)}) - \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\nabla f_c(\boldsymbol{\phi}_c^{(t-1)})\\ &\quad + \frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}}\nabla f_k(\boldsymbol{\phi}_k^{(t-1)})\\ &= \nabla F(\mathbf{w}^{(t-1)}).\end{aligned}$$

□

A.2 Proof of Lemma 3.4

PROOF. Following the mini-batch definition in the algorithm in §3.5.1 and for all $\mathbf{x}$ and $\mathbf{y}$, we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{B}}\left[\|\nabla f_{\mathcal{B}}(\mathbf{x}) - \nabla f_{\mathcal{B}}(\mathbf{y})\|^2\right]\\ &= \frac{1}{|\mathcal{B}|^2}\mathbb{E}_{\mathcal{B}}\Big[\Big\|\sum_{c\in\mathcal{B}}\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\Big\|^2\Big]\\ &= \frac{1}{|\mathcal{B}|^2}\mathbb{E}_{\mathcal{B}}\Big[\sum_{c\neq c'\in\mathcal{B}}\langle\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle\Big] \qquad \text{(A.1)}\\ &\quad + \frac{|\mathcal{B}|}{|\mathcal{B}|^2}\mathbb{E}_c\left[\|\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\|^2\right].\end{aligned}$$

By switching the order of selection in formulating mini-batches, we take expectation with respect to mini-batches and obtain

$$\begin{aligned}&\frac{1}{|\mathcal{B}|^2}\mathbb{E}_{\mathcal{B}}\Big[\sum_{c\neq c'\in\mathcal{B}}\langle\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle\Big]\\ &= \frac{1}{|\mathcal{B}|^2\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{i=1}^{\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{c\neq c'\in\mathcal{B}_i}\langle\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle\\ &= \frac{1}{|\mathcal{B}|^2\binom{|\mathcal{C}|}{|\mathcal{B}|}}\sum_{c\neq c'\in\mathcal{C}}\binom{|\mathcal{C}|-2}{|\mathcal{B}|-2}\langle\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle\\ &= \frac{|\mathcal{B}|-1}{|\mathcal{B}|\cdot|\mathcal{C}|(|\mathcal{C}|-1)}\sum_{c\neq c'\in\mathcal{C}}\langle\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle. \qquad \text{(A.2)}\end{aligned}$$

Note that the right-hand size of (A.2) does not depend on expectation with respect to randomly sampled mini-batches.

Now we go on to replace term (A.1) with the right-hand side of the results in (A.2). Then we further obtain

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\mathbf{x}) - \nabla f_{\mathcal{B}}(\mathbf{y})\|^2\big] \\
&= \frac{|\mathcal{B}|-1}{|\mathcal{B}|\cdot|\mathcal{C}|(|\mathcal{C}|-1)} \sum_{c\neq c'\in\mathcal{C}} \langle \nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle \\
&\quad + \frac{1}{|\mathcal{B}|}\mathbb{E}_c\big[\|\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\|^2\big] \\
&= \frac{|\mathcal{B}|-1}{|\mathcal{B}|\cdot|\mathcal{C}|(|\mathcal{C}|-1)} \sum_{c,c'\in\mathcal{C}} \langle \nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y}), \nabla f_{c'}(\mathbf{x}) - \nabla f_{c'}(\mathbf{y})\rangle \\
&\quad - \Big(\frac{|\mathcal{B}|-1}{|\mathcal{B}|(|\mathcal{C}|-1)} - \frac{1}{|\mathcal{B}|}\Big)\mathbb{E}_c\big[\|\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\|^2\big] \\
&= \frac{|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{C}|}{|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|}\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|^2 \\
&\quad + \frac{|\mathcal{C}|-|\mathcal{B}|}{|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|}\mathbb{E}_c\big[\|\nabla f_c(\mathbf{x}) - \nabla f_c(\mathbf{y})\|^2\big],
\end{aligned}$$

where the last equality is obtained by the relation $\big[(|\mathcal{B}|-1)/[|\mathcal{B}|\cdot|\mathcal{C}|(|\mathcal{C}|-1)]\big]\|\sum_{c\in\mathcal{C}}\nabla f_c(\mathbf{x})-\nabla f_c(\mathbf{y})\|^2 = \big[(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{C}|)/(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|)\big]\|\nabla F(\mathbf{x})-\nabla F(\mathbf{y})\|^2$. □

A.3 Proof of Theorem 3.5

PROOF. We refer to $\mathbf{h}_{\mathcal{B}}^{(t)}$ and $\mathbf{h}_c^{(t)}$ defined in (3.9) and (3.10). By the orthogonality property for non-overlapped coordinates, the non-expansiveness of the proximal operator [31], and that $\mathbf{w}^*$ is the global optimum in (3.6), we have

$$\begin{aligned}
&\mathbb{E}_j\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big] \\
&= \frac{(d-1)}{d}\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 \\
&\quad + \frac{1}{d}\big\|\mathrm{prox}_\eta(\mathbf{w}^{(t-1)} - \eta\mathbf{h}_{\mathcal{B}}^{(t)}) - \mathrm{prox}_\eta(\mathbf{w}^* - \eta\nabla F(\mathbf{w}^*))\big\|_2^2 \\
&\leq \frac{1}{d}\Big[(d-1)\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 + \|\mathbf{w}^{(t-1)} - \eta\mathbf{h}_{\mathcal{B}}^{(t)} - \mathbf{w}^* + \eta\nabla F(\mathbf{w}^*)\|_2^2\Big].
\end{aligned}$$

After applying the results of Lemma 3.3, with a further simplification of terms, we can get

$$\begin{aligned}
&\mathbb{E}_{\mathcal{B},j}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big] = \mathbb{E}_{\mathcal{B}}\Big[\mathbb{E}_j\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big]\Big] \\
&\leq \frac{1}{d}\mathbb{E}_{\mathcal{B}}\big[(d-1)\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 \\
&\qquad + \|\mathbf{w}^{(t-1)} - \eta\mathbf{h}_{\mathcal{B}}^{(t)} - \mathbf{w}^* + \eta\nabla F(\mathbf{w}^*)\|_2^2\big] \\
&= \frac{1}{d}\Big[(d-1)\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 + \|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 \\
&\qquad - 2\eta\langle\nabla F(\mathbf{w}^{(t-1)}) - \nabla F(\mathbf{w}^*), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle \\
&\qquad + \eta^2\mathbb{E}_{\mathcal{B}}\big[\|\mathbf{h}_{\mathcal{B}}^{(t)} - \nabla F(\mathbf{w}^*)\|_2^2\big]\Big].
\end{aligned}$$

Now we use the property that $\mathbb{E}\big[\|\mathbf{x}\|_2^2\big] = \mathbb{E}\big[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|_2^2\big] + \big\|\mathbb{E}[\mathbf{x}]\big\|^2$ for all $\mathbf{x}$ and the property that $\|\mathbf{x}+\mathbf{y}\|_2^2 \leq (1+\zeta)\|\mathbf{x}\|_2^2 + (1+\zeta^{-1})\|\mathbf{y}\|_2^2$ for all $\mathbf{x}, \mathbf{y}$, and $\zeta > 0$. It holds that $\mathbb{E}_{\mathcal{B}}\big[\|\mathbf{h}_{\mathcal{B}}^{(t)} - \nabla F(\mathbf{w}^*)\|_2^2\big] \leq (1+\zeta)\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\mathbf{w}^{(t-1)}) - \nabla f_{\mathcal{B}}(\mathbf{w}^*)\|_2^2\big] - \zeta\|\nabla F(\mathbf{w}^{(t-1)}) - \nabla F(\mathbf{w}^*)\|_2^2 + (1+\zeta^{-1})\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t-1)}) - \nabla f_{\mathcal{B}}(\mathbf{w}^*)\|_2^2\big]$. Therefore, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{B},j}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big] \leq \frac{1}{d}\Big[&d\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 \\
&+ 2\eta\langle\nabla F(\mathbf{w}^*), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle - 2\eta\langle\nabla F(\mathbf{w}^{(t-1)}), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle \\
&+ \eta^2(1+\zeta)\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\mathbf{w}^{(t-1)}) - \nabla f_{\mathcal{B}}(\mathbf{w}^*)\|_2^2\big] \\
&+ \eta^2(1+\zeta^{-1})\mathbb{E}_{\mathcal{B}}\big[\|\nabla f_{\mathcal{B}}(\boldsymbol{\phi}_{\mathcal{B}}^{(t-1)}) - \nabla f_{\mathcal{B}}(\mathbf{w}^*)\|_2^2\big] \\
&- \eta^2\zeta\|\nabla F(\mathbf{w}^{(t-1)}) - \nabla F(\mathbf{w}^*)\|_2^2\Big]. \qquad \text{(A.3)}
\end{aligned}$$

Lemma 3.4 is used to replace the two expectation terms with respect to mini-batches on the right-hand side of (A.3). By the property of any function f that is convex and has a Lipschitz continuous gradient with constant L: $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2/(2L)$ for all $\mathbf{x}$ and $\mathbf{y}$ [30], we can further simplify (A.3) and multiply it by a positive constant κ:

$$\begin{aligned}
\kappa\mathbb{E}_{\mathcal{B},j}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big] \leq{}& \frac{\kappa(d-\eta\mu)}{d}\|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2 \\
&+ \Big(\frac{\kappa\eta^2(1+\zeta)(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{C}|)}{d(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|)} - \frac{\kappa\eta^2\zeta}{d}\Big)\|\nabla F(\mathbf{w}^{(t-1)}) \\
&- \nabla F(\mathbf{w}^*)\|_2^2 + \frac{2\kappa L\eta^2(1+\zeta^{-1})}{d}\Big[\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} f_c(\boldsymbol{\phi}_c^{(t-1)}) - F(\mathbf{w}^*) \\
&- \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\langle\nabla f_c(\mathbf{w}^*), \boldsymbol{\phi}_c^{(t-1)} - \mathbf{w}^*\rangle\Big] + \Big(\frac{\kappa\eta^2(1+\zeta)(|\mathcal{C}|-|\mathcal{B}|)}{d(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|)} \\
&- \frac{\kappa\eta}{dL}\Big)\mathbb{E}_c\big[\|\nabla f_c(\mathbf{w}^{(t-1)}) - \nabla f_c(\mathbf{w}^*)\|_2^2\big] \\
&- \frac{2\kappa(L-\mu)\eta}{dL}\big[F(\mathbf{w}^{(t-1)}) - F(\mathbf{w}^*) - \langle\nabla F(\mathbf{w}^*), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle\big]. \qquad \text{(A.4)}
\end{aligned}$$

By the property of any strongly convex function f with the convexity parameter μ that $f(\mathbf{y}) \leq f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2/(2\mu)$ for all $\mathbf{x}$ and $\mathbf{y}$ [30], we have $-\|\nabla F(\mathbf{w}^{(t-1)}) - \nabla F(\mathbf{w}^*)\|_2^2 \leq -2\mu\big[F(\mathbf{w}^{(t-1)}) - F(\mathbf{w}^*) - \langle\nabla F(\mathbf{w}^*), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle\big]$.

With defining $Y_{\mathcal{B}}^{(t)} = (1/|\mathcal{C}|)\cdot\big[\sum_{c\in\mathcal{B}} f_c(\boldsymbol{\phi}_c^{(t)}) + \sum_{c\notin\mathcal{B}\wedge c\in\mathcal{C}} f_c(\boldsymbol{\phi}_c^{(t)})\big] - F(\mathbf{w}^*) - (1/|\mathcal{C}|)\cdot\big[\sum_{c\in\mathcal{B}}\langle\nabla f_c(\mathbf{w}^*), \boldsymbol{\phi}_c^{(t)} - \mathbf{w}^*\rangle + \sum_{c\notin\mathcal{B}\wedge c\in\mathcal{C}}\langle\nabla f_c(\mathbf{w}^*), \boldsymbol{\phi}_c^{(t)} - \mathbf{w}^*\rangle\big] + \kappa\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2$, by (A.4) and for all $\alpha > 0$, we obtain $\mathbb{E}_{\mathcal{B},j}[Y_{\mathcal{B}}^{(t)}] - \alpha Y_{\mathcal{B}}^{(t-1)} \leq \sum_{k=1}^4 \rho_k\tau_k$, where the four constants are $\rho_1 = (\kappa/d)\cdot\big[\eta^2(1+\zeta)(|\mathcal{C}|-|\mathcal{B}|)/(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|) - \eta/L\big]$, $\rho_2 = |\mathcal{B}|/|\mathcal{C}| + \big[2\kappa\eta^2\mu(1+\zeta)(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{C}|)\big]/\big[d(|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|)\big] - 2\kappa\eta^2\mu\zeta/d - 2\kappa(L-\mu)\eta/(dL)$, $\rho_3 = \kappa(1-\eta\mu/d-\alpha)$, and $\rho_4 = 2\kappa L\eta^2(1+\zeta^{-1})/d - \alpha + (|\mathcal{C}|-|\mathcal{B}|)/|\mathcal{C}|$; and their associated terms are $\tau_1 = \mathbb{E}_c\big[\|\nabla f_c(\mathbf{w}^{(t-1)}) - \nabla f_c(\mathbf{w}^*)\|_2^2\big]$, $\tau_2 = F(\mathbf{w}^{(t-1)}) - F(\mathbf{w}^*) - \langle\nabla F(\mathbf{w}^*), \mathbf{w}^{(t-1)} - \mathbf{w}^*\rangle$, $\tau_3 = \|\mathbf{w}^{(t-1)} - \mathbf{w}^*\|_2^2$, and $\tau_4 = (1/|\mathcal{C}|)\sum_{c\in\mathcal{C}} f_c(\boldsymbol{\phi}_c^{(t-1)}) - F(\mathbf{w}^*) - (1/|\mathcal{C}|)\sum_{c\in\mathcal{C}}\langle\nabla f_c(\mathbf{w}^*), \boldsymbol{\phi}_c^{(t-1)} - \mathbf{w}^*\rangle$.

It is obvious that $\tau_1 \geq 0$ and $\tau_3 \geq 0$. By the convexity property of F, $\tau_2 \geq 0$ and $\tau_4 \geq 0$. For the step size, we choose

$$\eta = \frac{|\mathcal{B}|\cdot|\mathcal{C}|-|\mathcal{B}|}{2(L+|\mathcal{C}|\mu)(|\mathcal{C}|-|\mathcal{B}|)}.$$

To ensure $0 < \eta\mu < 1$, we choose a mini-batch size satisfying

$$1 \leq |\mathcal{B}| < \frac{2|\mathcal{C}|(|\mathcal{C}|\mu+L)}{2(|\mathcal{C}|\mu+L)+(|\mathcal{C}|\mu-\mu)}.$$

By setting $\rho_1 = 0$ with $\zeta = (L+2|\mathcal{C}|\mu)/L > 0$, $\rho_2 = 0$ with $\kappa = (|\mathcal{B}|d)/\big[2|\mathcal{C}|\eta(1-\eta\mu)\big] > 0$, and $\rho_3 = 0$ with $\alpha = 1-(\eta\mu)/d$, we have $\rho_4 \leq 0$. Thus, $\mathbb{E}_{\mathcal{B},j}[Y_{\mathcal{B}}^{(t)}] - \alpha Y_{\mathcal{B}}^{(t-1)} \leq 0$, where the expectation is conditional on information from the previous iteration $t-1$. Taking expectation with this previous iteration gives $\mathbb{E}_{\mathcal{B},j}[Y_{\mathcal{B}}^{(t)}] \leq \alpha\mathbb{E}_{\mathcal{B},j}[Y_{\mathcal{B}}^{(t-1)}]$. By chaining over t, $\mathbb{E}_{\mathcal{B},j}[Y_{\mathcal{B}}^{(t)}] \leq \alpha^t Y_{\mathcal{B}}^{(0)}$. Since $\kappa\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2 \leq Y_{\mathcal{B}}^{(t)}$ (note that the sum of the first three terms in $Y_{\mathcal{B}}^{(t)}$ is non-negative by the convexity property of F), given the parameter settings above, for the composite objectives in (3.6) and the optimization algorithm in §3.5.1, we have $\mathbb{E}_{\mathcal{B},j}\big[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|_2^2\big] \leq \alpha^t(C_1 + C_2/\kappa)$, where C_1 and C_2 are constants determined by $\mathbf{w}^{(0)}$. Note that $0 < \alpha < 1$. The algorithm in §3.5.1 has a linear rate of convergence. □

7. REFERENCES

[1] R. Baeza-Yates, G. Dupret, and J. Velasco. A study of mobile search queries in japan. In *Proceedings of the International World Wide Web Conference (WWW)*, 2007.

[2] R. Baeza-Yates, D. Jiang, F. Silvestri, and B. Harrison. Predicting the next app that you are going to use. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2015.

[3] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2011.

[4] U. Bhandari, K. Sugiyama, A. Datta, and R. Jindal. Serendipitous recommendation for mobile apps using item-item similarity graph. In *Information Retrieval Technology*. 2013.

[5] C. M. Bishop. *Pattern recognition and machine learning*, volume 1. Springer-Verlag New York, 2006.

[6] F. Cai, S. Liang, and M. de Rijke. Time-sensitive personalized query auto-completion. In *Proceedings of the ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, 2014.

[7] S. Chaudhuri and R. Kaushik. Extending autocompletion to tolerate errors. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2009.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*, volume 2. MIT press, 2001.

[9] E. Costa-Montenegro, A. B. Barragáns-Martínez, and M. Rey-López. Which app? A recommender system of applications in markets: Implementation of the service for monitoring users' interaction. *Expert systems with applications*, 39(10), 2012.

[10] Y. Cui and K. Liang. A probabilistic top-n algorithm for mobile applications recommendation. In *IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT)*, 2013.

[11] C. Davidsson and S. Moritz. Utilizing implicit feedback and context to recommend mobile applications from first use. In *Proceedings of the Workshop on Context-awareness in Retrieval and Recommendation*, 2011.

[12] H. Duan and B.-J. P. Hsu. Online spelling correction for query completion. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2011.

[13] S. Fu, B. Pi, M. Desmarais, Y. Zhou, W. Wang, and S. Han. Query recommendation and its usefulness evaluation on mobile search engine. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2009.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. 2009.

[15] K. Hofmann, B. Mitra, F. Radlinski, and M. Shokouhi. An eye-tracking study of user interactions with query auto completion. In *Proceedings of the ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, 2014.

[16] B.-J. P. Hsu and G. Ottaviano. Space-efficient data structures for top-k completion. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2013.

[17] R. Islam, R. Islam, and T. Mazumder. Mobile application and its global impact. *International Journal of Engineering & Technology (IJEST)*, 10(6), 2010.

[18] S. Ji, G. Li, C. Li, and J. Feng. Efficient interactive fuzzy keyword search. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2009.

[19] J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien, and P.-J. Cheng. Learning user reformulation behavior for query auto-completion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.

[20] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[21] M. Kamvar, M. Kellar, R. Patel, and Y. Xu. Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices. In *Proceedings of the international conference on World Wide Web (WWW)*, 2009.

[22] W. Kong, R. Li, J. Luo, A. Zhang, Y. Chang, and J. Allan. Predicting search intent based on pre-search context. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2015.

[23] L. Li, H. Deng, A. Dong, Y. Chang, H. Zha, and R. Baeza-Yates. Analyzing user's sequential behavior in query auto-completion via markov processes. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2015.

[24] Y. Li, A. Dong, H. Wang, H. Deng, Y. Chang, and C. Zhai. A two-dimensional click model for query auto-completion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.

[25] Z.-X. Liao, S.-C. Li, W.-C. Peng, P. S. Yu, and T.-C. Liu. On the feature discovery for app usage prediction in smartphones. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013.

[26] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua. New and improved: Modeling versions to improve app recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.

[27] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2015.

[28] B. Mitra, M. Shokouhi, F. Radlinski, and K. Hofmann. On user interactions with query auto-completion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2014.

[29] T. Miyanishi and T. Sakai. Time-aware structured query suggestion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2013.

[30] Y. Nesterov. *Introductory lectures on convex optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2004.

[31] Y. Nesterov. *Gradient methods for minimizing composite objective function*. Technical report, Center for Operations Research and Econometrics, 2007.

[32] M. Schmidt, R. Babanezhad, M. O. Ahemd, A. Defazio, A. Clifton, and A. Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[33] W. Shi and A. Yin. Interoperability-enriched app recommendation. In *IEEE International Conference on Data Mining Workshop (ICDMW)*, 2014.

[34] M. Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2013.

[35] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2012.

[36] Y. Song, H. Ma, H. Wang, and K. Wang. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *Proceedings of the international conference on World Wide Web (WWW)*, 2013.

[37] S. Whiting and J. M. Jose. Recent and robust query auto-completion. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2014.

[38] C. Xiao, J. Qin, W. Wang, Y. Ishikawa, K. Tsuda, and K. Sadakane. Efficient error-tolerant query autocompletion. *Proceedings of the Very Large Data Base Endowment (VLDB)*, 6(6), 2013.

[39] C. Yang, T. Wang, G. Yin, H. Wang, M. Wu, and M. Xiao. Personalized mobile application discovery. In *Proceedings of the International Workshop on Crowd-based Software Development Methods and Technologies*, 2014.

[40] S. Yang, H. Yu, W. Deng, and X. Lai. Mobile application recommendations based on complex information. In *Current Approaches in Applied Artificial Intelligence*. 2015.

[41] S.-H. Yang, B. Long, A. J. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2011.

[42] P. Yin, P. Luo, W.-C. Lee, and M. Wang. App recommendation: a contest between satisfaction and temptation. In *Proceedings of the ACM international conference on Web search and data mining (WSDM)*, 2013.

[43] A. Zhang, A. Goyal, W. Kong, H. Deng, A. Dong, Y. Chang, C. A. Gunter, and J. Han. adaqac: Adaptive query auto-completion via implicit negative feedback. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2015.

[44] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Mobile app recommendations with security and privacy awareness. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 2014.