SELF-LEARNING MEDIA SEARCH ENGINES

Clement Leung[1]
Nikki Kuang[2]
[1] Chinese University of Hong Kong, Shenzhen &
Shenzhen Research Institute of Big Data
[2]University of California at San Diego
clementleung@cuhk.edu.cn

Clement Leung

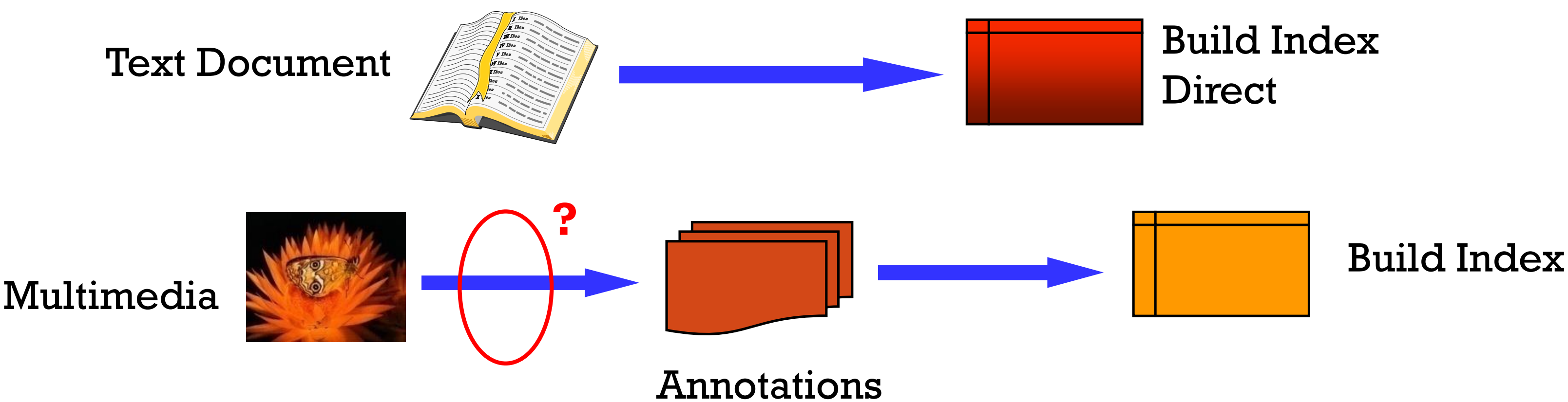
- FULL PROFESSORSHIPS at
  - University of London, UK; National University of Singapore; Chinese University of Hong Kong, Shenzhen, China; Victoria University, Australia; Hong Kong Baptist University
- Two US patents, five books and over 150 research articles
- Program Chair, Keynote Speaker, Panel Expert of major International Conferences
- Editorial Board of ten International Journals
- Listed in Who's Who in the World and Great Minds of the 21st Century
- Fellow of the British Computer Society, Fellow of the Royal Society of Arts, Chartered Engineer

Motivation & Overview

- Capture and learn the subtle nuances of human perceptions and deep knowledge for searching
- Develop an architectural paradigm enables indexes to evolve naturally while accommodating the dynamic changes of user interests
- Enable progressive improvement in search performance over time
- Use a reinforcement learning framework based on Markov Decision Process
- Prevent local optimum and use evolutionary exploration strategies which balance exploitation and exploration in reinforcement learning

The Multimedia Data Extraction and Indexing Problem

- Computer vision
  - Too slow to deliver
- Dedicated intensive manual indexing infeasible
  - Fast creation and slow indexing
  - $\text{Rate}_{\text{creation}} >> \text{Rate}_{\text{indexing}}$



Challenges of Multimedia Data Compared with Text-oriented Data

- Unlike text-based data, no automatic algorithms available for effectively extracting information from multimedia data
- Velocity mismatch: the speed of creation of multimedia data is orders of magnitude faster than the creation of text-oriented data

Example:

- "*Les Misérables*" by Victor Hugo

	One Image by smart Phone	Entire E-book
File size	2~3 MB	About 2.1 MB (530,982 Text words)
Time taken	< 1 second	1815~1832 > 10 years

Semantic and Deep Knowledge Queries

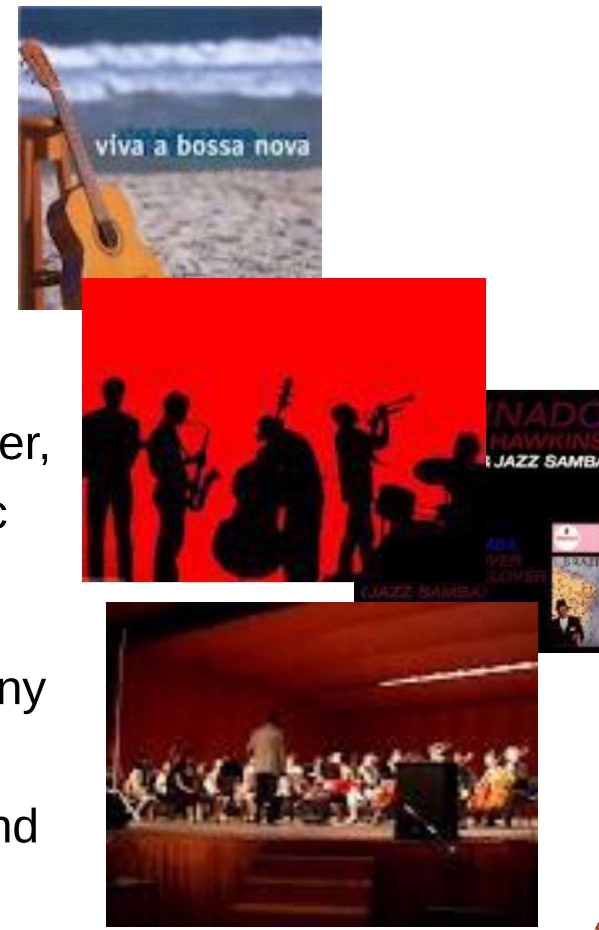
- Meaningful interpretation of the objects in the picture constitutes a significant semantic element, and requires deep knowledge based on prior familiarity with the subject matter
- Mere visual description of image objects is insufficient

Samson & Delilah

Moses & 10 Commandments

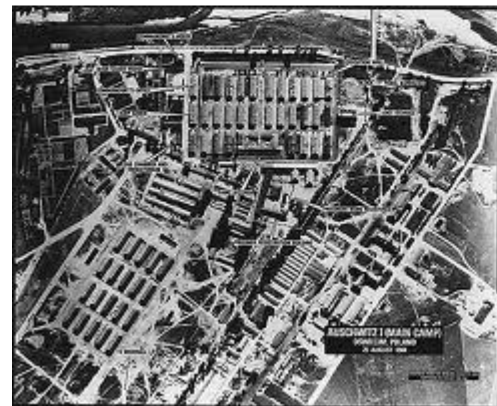
Music Genome

- Several hundred attributes (genes) are used to characterize songs, such as
  - Tempo, Key, Harmony
  - Melody, Rhythm, Syncopation
  - Piano block chord, octave guitar, oboe+flute, etc.
  - Mere descriptions of song title, performer, lyrics are insufficient to satisfy semantic queries
- Effective music search can make use of any combination of these attributes, and such subtleties and nuances may be learned and indexed collaboratively based on reinforcement learning principles



Expert Knowledge from Imagery Intelligence

- Correct identification of foreign military installations requires expert knowledge
  - Weapons of mass destruction
  - Nuclear facilities
  - Enables military commanders in to identify bombing targets, locate and track enemy forces, determine the extent and strengths of fortifications
- President Eisenhower in 1961 originally started the US National Photographic Interpretation Center (NPIC)
  - During World War II, the US Army Air Forces built a formidable capability to collect, analyze, and disseminate photographic intelligence
    - Cuban Missile Crisis
  - The functions of NPIC have since been absorbed into the National Geospatial-Intelligence Agency (NGA)



Self-Learning Search Engine

- A Self-Learning Search Engine (SLSE) is a multimedia search engine that continuously learns and evolves to adapt its answer lists to queries submitted by users

- When a user submits a query *Q*, the search engine takes a hybrid evolutionary exploration strategy to construct and present a result retrieval list of *M* objects *M_List* to the user for evaluation

Dynamic Indexing

- Dynamic indexing is an indexing technique that dynamically builds semantic indexes to associate query terms with multimedia objects
- New query terms are dynamically constructed as indexes for desired objects, and existing indexes are able to be deconstructed according to demand
- The relevance score of an index continuously changes during the process

Relevance Index Value (RIV)

- SLSE creates a learning function between the object space and the query *Q* to measure the relevance liaison:

$$\mathcal{L} : T \times O \to \mathbb{R}^+ \cup 0,$$

- where *T* is the space of query terms that represents the input query *Q*, and *O* is the object space
- The learning function takes the form

$$\mathcal{L}(\tau, o) \to \omega\mathcal{R}, \omega \in (0, 1),$$

- The output of the function is the set of non-negative real numbers r that specify the corresponding relevance with 0 indicating complete irrelevance
- At the beginning, all RIV scores are initialized by the system; later in the usage, the learning function takes the results of the reward function as input to update pertinent RIV scores iteratively

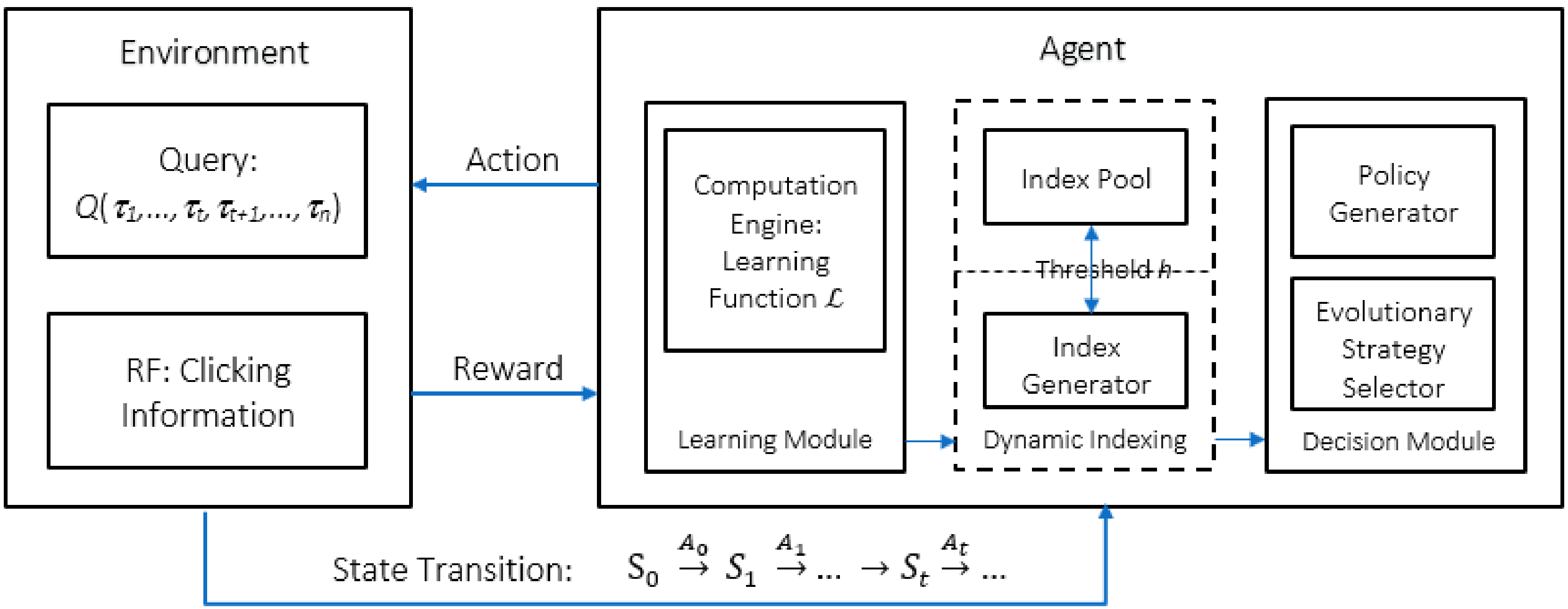
Markov Decision Process

- Represented as a Markov Decision Process with five tuple

$$\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma\}$$

- with the respective components of state space, action space, transition kernel, reward function, and discount factor
- **Action Space**: consists of a series of actions that the agent selects *M* objects to form an *M-List* and presents to the user
- **State Space**: a set of all indexes in the dynamic indexing component, including the explored and unexplored ones together with their RIVs
- For unexplored indexes, their RIV are below a pre-defined threshold $h > 0$, while for explored indexes, their RIV are at or above it

Learning Based Architecture



- After taking a particular action, RIV scores change accordingly, causing a corresponding state transition in the system
- The long-term goal of SLSE is to expose unexplored indexes for retrieval, and reward will be the net change in the total RIV scores as a result of an action

Pure Exploitation Search

- When a user submits a query *Q*, the search engine constructs a set of M_Q objects, which is a set containing *M* objects called *M-List*, and present it to the user for evaluation
- Each object will have a relevance score with respect to the query, and those objects with the highest relevance scores will be selected for inclusion in the *M-List*

  ⇒ Objects with high RIV will be shown repeatedly for user evaluation
- The RIV of these objects tends to keep increasing even though they may not be the most relevant as these are selected as relevant (clicked) by the users
- The most relevant objects may not stand any chance of having their RIV increased since they are not shown to the users for evaluation
- Local optimum problem
  - objects that have the highest RIV may not in fact be the most relevant

Departure from Pure Exploration Search

- The candidate returned objects M_Q for the *M-List* of Q should not be just consisting of those objects having the highest relevance score
- The *M-List* should aim to contain two categories of objects:
  - a K-object subset O_a that has the highest cumulative RIV scores from exploitation, and
  - a subset O_b of random objects selected for exploration, i.e.

$$M_Q = O_a \cup O_b$$

ε-Greedy Search

- In the case of pure exploitation search, we have

$$|O_a| = K = M$$

$$|O_b| = 0$$

- Since $|O_a| = M$ would risk landing in a local optimum for the query Q, we wish to strike a balance between exploitation and exploration

  ⇒ Design the *M-List* in such a way that

$$|O_a| < M$$

$$|O_b| > 0$$

ε-Greedy Search

- Here, we assume that the ordering of objects within the *M*-list is unimportant, and repeated sequential presentations of the *M*-list in response to the given query *Q* (mostly likely from different users) are denoted by the M_1, M_2, M_3, …, where M_i signifies the i^{th} *M*-list presented for the query *Q*
- For 0 < ε < 1, we let $r = \varepsilon M$, and $K=(1-\varepsilon)M$
  - i.e. we include *r* randomly chosen objects in the *M*-list, where each available object apart from the *K* objects from exploitation is chosen with equal probability
- Without jeopardizing the performance of the search system, we in general use a small value for ε, and in this study we take ε < 15%

$$|O_a| = K > 0.85M$$

$$|O_b| = r < 0.15M$$

ε-Greedy Algorithm A

- When a given random object *Z* has been included in a previous *M*-list presentation, it can be re-selected for inclusion in a subsequent *M*-list presentation in the exploration process
- Advantage: Greater fault-tolerance – the most relevant object can be shown again

Algorithm 1 EGSE-A: Search Space Exploration with Constant Probability

Require: epsilon E, length of result list M, query max counter C

Initialize terminating condition $\Delta \leftarrow False$,
Initialize query counter $\Theta \leftarrow 0$,
Initialize exploration proportion $R \leftarrow E \times M$,
Initialize exploitation proportion $K \leftarrow (1 - E) \times M$,
while Δ == **False** **do**
    Retrieve and parse new user query Q
    Determine $S_1 = \{O_i \mid$ objects with the highest relevant scores$\}_{i=1}^{k}$, where $|S_1| = K$
    Determine $S_2 = \{O_j \mid O_j \in S_1^{\complement}\}_{j=1}^{R}$, where $|S_2| = R$
    Present M-list := $S_1 \cup S_2$ to user
    Capture object click information from user
    Increment the score of clicked objects
    $\Theta \leftarrow \Theta + 1$
    if $\Theta == C$ **then**
        $\Delta \leftarrow True$

Performance of EGSE-A

- Let X be the multimedia object that is most relevant to the query Q, but its current RIV is not sufficient for inclusion in O_a
- Let $\boldsymbol{U_{r,M}}$ the random variable signifies the time to discover X for the first time; we have

$$\mathbb{P}\left[\boldsymbol{U_{r,M}} = k\right] = \alpha_{r,M}\beta_{r,M}^{k-1},$$

where

$$\alpha_{r,M} = \frac{\binom{N-M+r-1}{r-1}}{\binom{N-M+r}{r}},$$

and $\alpha_{r,M} + \beta_{r,M} = 1$

- The corresponding probability generating function is given by

$$F(z) = \frac{\alpha_{r,M} z}{1 - \beta_{r,M} z}$$

Performance of EGSE-A

- The mean and variance of $\boldsymbol{U_{r,M}}$ can be obtained by differentiation

$$\mathrm{E}\left[\boldsymbol{U_{r,M}}\right] = \frac{\binom{N-M+r}{r}}{\binom{N-M+r-1}{r-1}}$$

$$\mathrm{Var}\left[\boldsymbol{U_{r,M}}\right] = \frac{\binom{N-M+r}{r}^2}{\binom{N-M+r-1}{r-1}^2} \times \left\{ \frac{\binom{N-M+r}{r} - \binom{N-M+r-1}{r-1}}{\binom{N-M+r}{r}} \right\}$$

ε-Greedy Algorithm B

- When a given random object *Z* has been included in a previous *M*-list presentation, it is excluded in a subsequent *M*-list presentation in the exploration process
- Advantage: Greater efficiency in the speed of discovery

Algorithm 2 EGSE-B: Search Space Exploration with Variable Probability

Require: epsilon E, length of result list M, query max counter C

Initialize terminating condition $\Delta \leftarrow$ *False*,
Initialize query counter $\Theta \leftarrow 0$,
Initialize exploration proportion $R \leftarrow E \times M$,
Initialize exploitation proportion $K \leftarrow (1 - E) \times M$,
Initialize previously presented M-list for Query Q_i as $S_i \leftarrow \emptyset$, for all possible i
while Δ == **False** **do**
  Retrieve and parse new user query Q_i
  Determine $S_1 = \{O_l \mid \text{objects with the highest relevant scores}\}_{l=1}^{k}$, where $|S_1| = K$
  if $|(S_1 \cup S_i)^{\complement}| \geq R$ **then**
    Determine $S_2 = \{O_j \mid O_j \in (S_1 \cup S_i)^{\complement}\}_{j=1}^{R}$, where $|S_2| = R$
  else
    Determine $S_2 = (S_1 \cup S_i)^{\complement}$, where $|S_2| = |(S_1 \cup S_i)^{\complement}|$
  Present M-list := $S_1 \cup S_2$ for query Q_i to user
  $S_i \leftarrow S_i \cup S_1 \cup S_2$
  Capture object click information from user
  Increment the score of clicked objects
  $\Theta \leftarrow \Theta + 1$
  if $\Theta == C$ or $(S_1 \cup S_i)^{\complement} == \emptyset$ **then**
    $\Delta \leftarrow$ *True*

Performance of EGSE-B

- We denote by $f_{r,M,,k}$ the following first passage probability

$$f_{r,M,k} = \mathbb{P}\{X \in M_k : X \notin M_1, \dots, X \notin M_{k-1}\}$$

- For $k = 3$, we have

$$f_{r,M,3} = \mathbb{P}\{X \in M_3 : X \notin M_1, X \notin M_2\}$$

$$= \frac{\binom{N-M+r-1}{r}}{\binom{N-M+r}{r}} \times \frac{\binom{N-M-1}{r}}{\binom{N-M}{r}} \times \frac{\binom{N-M-r-1}{r-1}}{\binom{N-M-r}{r}}$$

Performance of EGSE-B

- The general recurrence relationship for $f_{r,M,k}$ can be seen to be

$$f_{r,M,k+1} = f_{r,M,k} \times \frac{\binom{N-K-kr}{r}}{\binom{N-K-kr-1}{r-1}}$$

$$\times \frac{\binom{N-K-kr-1}{r}}{\binom{N-K-kr}{r}} \times \frac{\binom{N-K-(k+1)r-1}{r-1}}{\binom{N-K-(k+1)r}{r}},$$

where the second factor serves to remove the successful inclusion probability in $f_{r,M,,k}$, and then replace this success probability by a failure to include *X* probability, which is the third factor. The final factor gives the successful inclusion probability at the (*k*+1)th presentation after *k* failed attempts to include *X* before

Performance of EGSE-B

- Solution of the above yields the mean and variance of $\boldsymbol{V_{r,M}}$, which is the random variable signifying the discovery of *X* for the first time

$$\mathrm{E}(\boldsymbol{V_{r,M}}) = \frac{N - M + 2r}{2r}$$

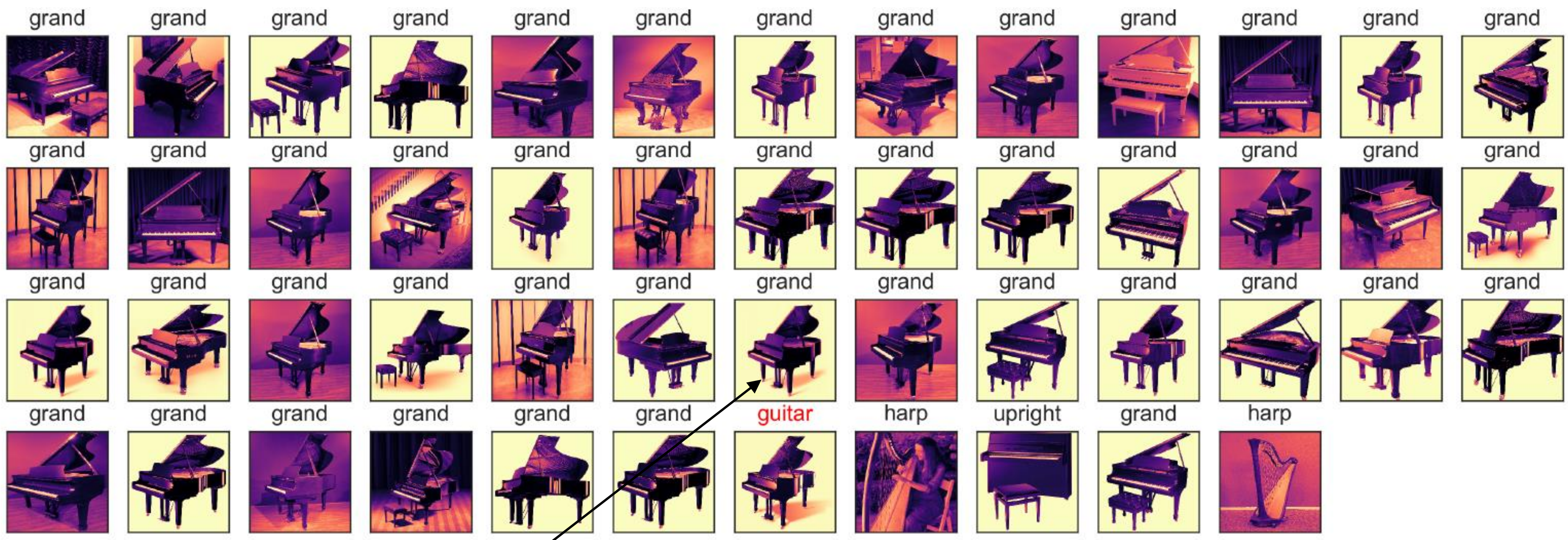
$$\mathrm{Var}(\boldsymbol{V_{r,M}}) = \frac{1}{12}\{[\frac{N - M + r}{r}]^2 - 1\}$$

Experimental Evaluations

Sample Images from Dataset

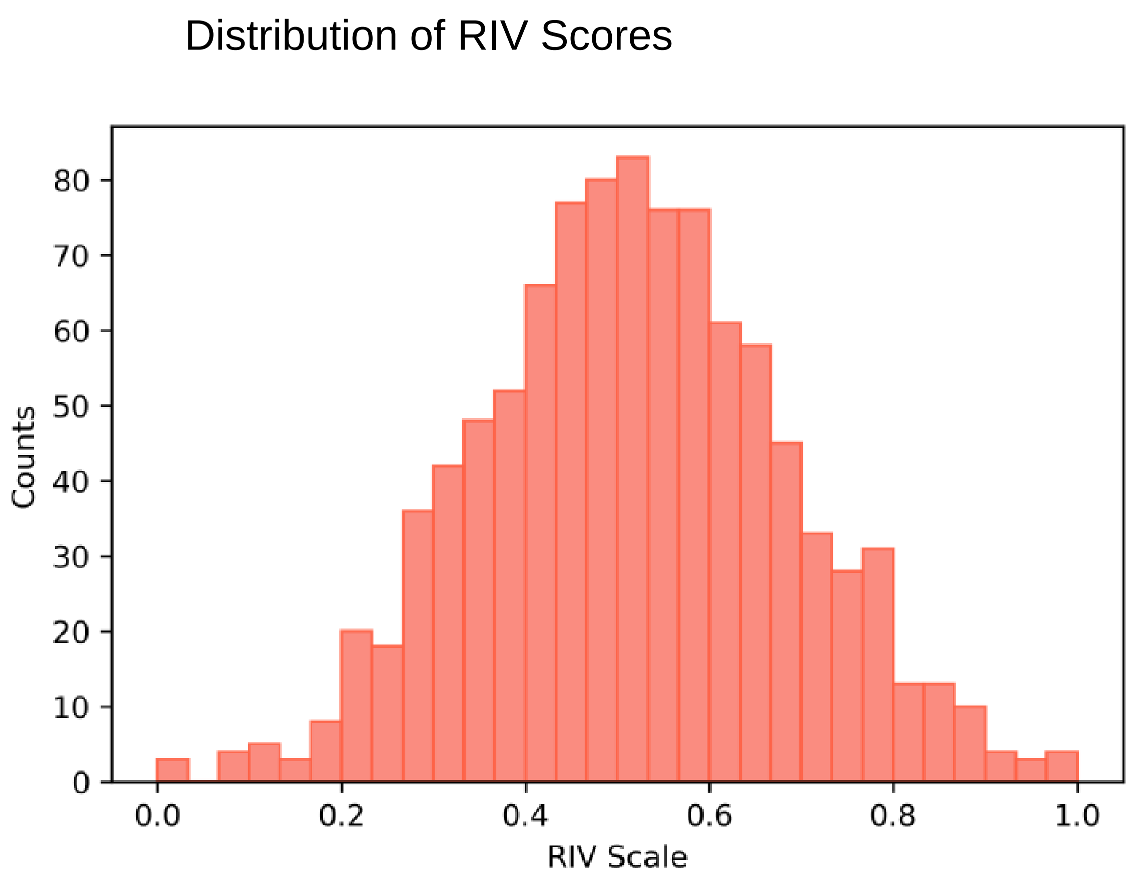
Experimental Evaluations

Final Returned *M-List* of "grand piano" using EGSE-A
Settings: N = 1,000, M = 50, $\varepsilon = 0.1$



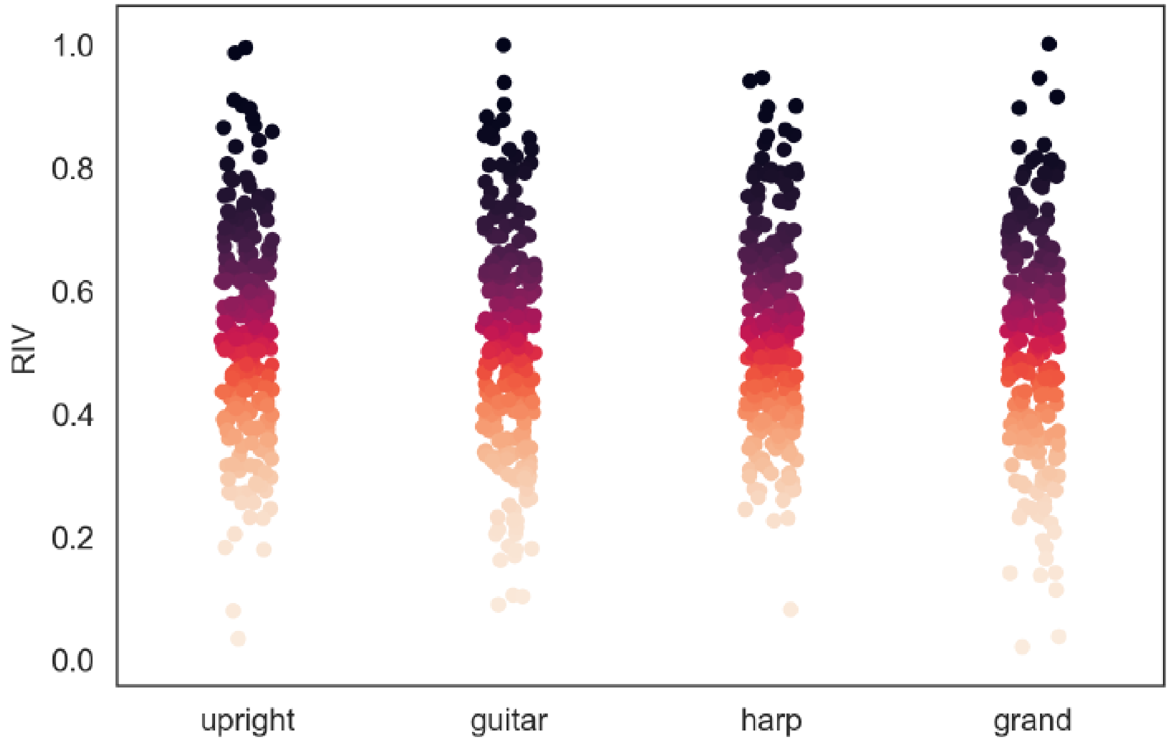
Note that the correct result is retrieved even though it is incorrectly labeled as "guitar"

Experimental Evaluations



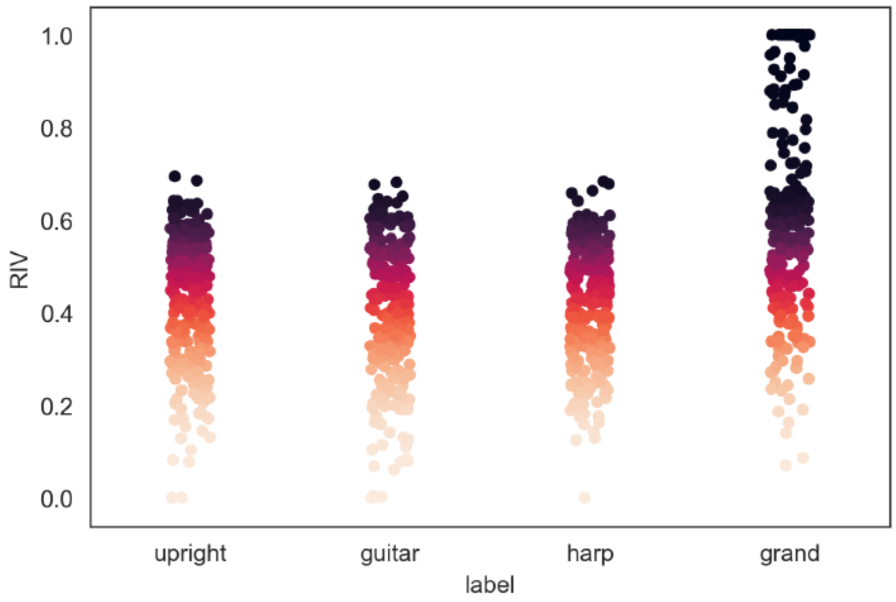
Experimental Evaluations

Distribution of Initial RIV Scores for
Each Category (EGSE-B)

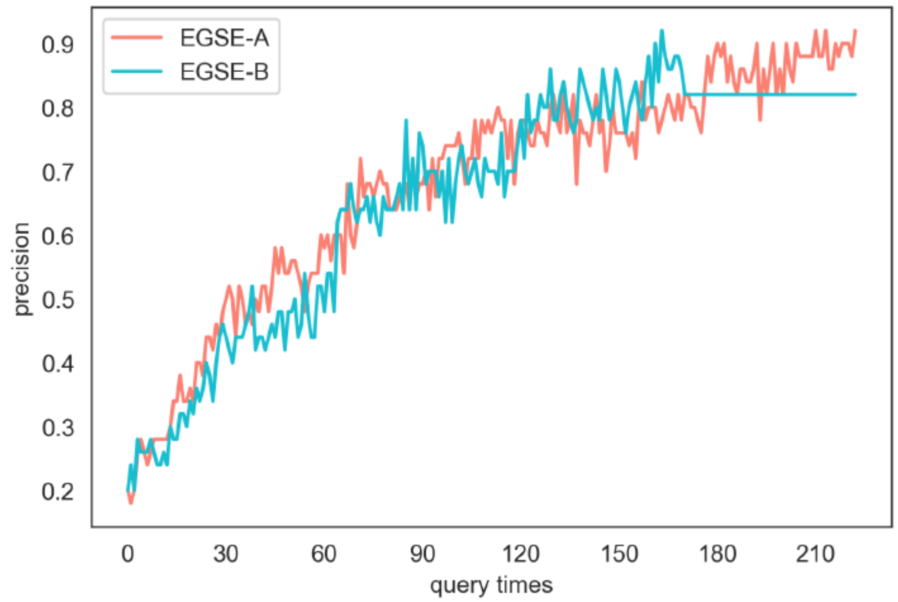


Experimental Evaluations

Distribution of RIV Scores for Each Category when Hidden Object *X* is Discovered (EGSE-B)

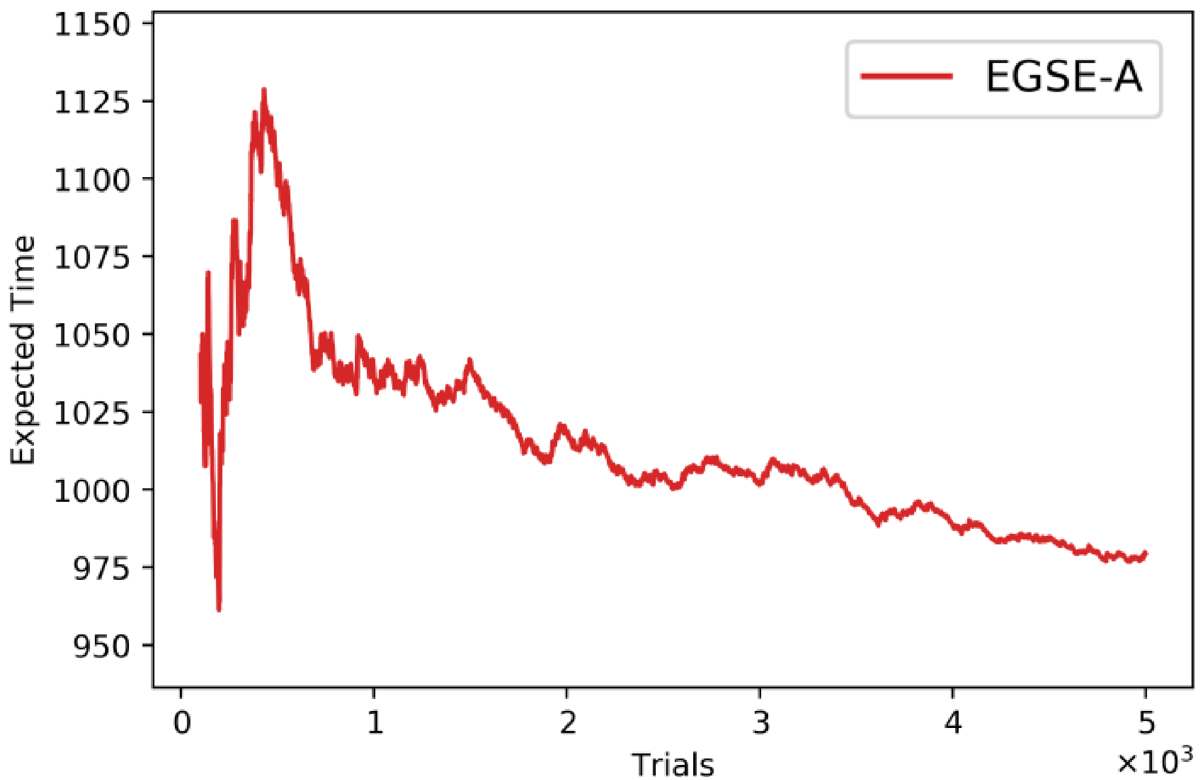


Evolution of Query Precision against Query Time



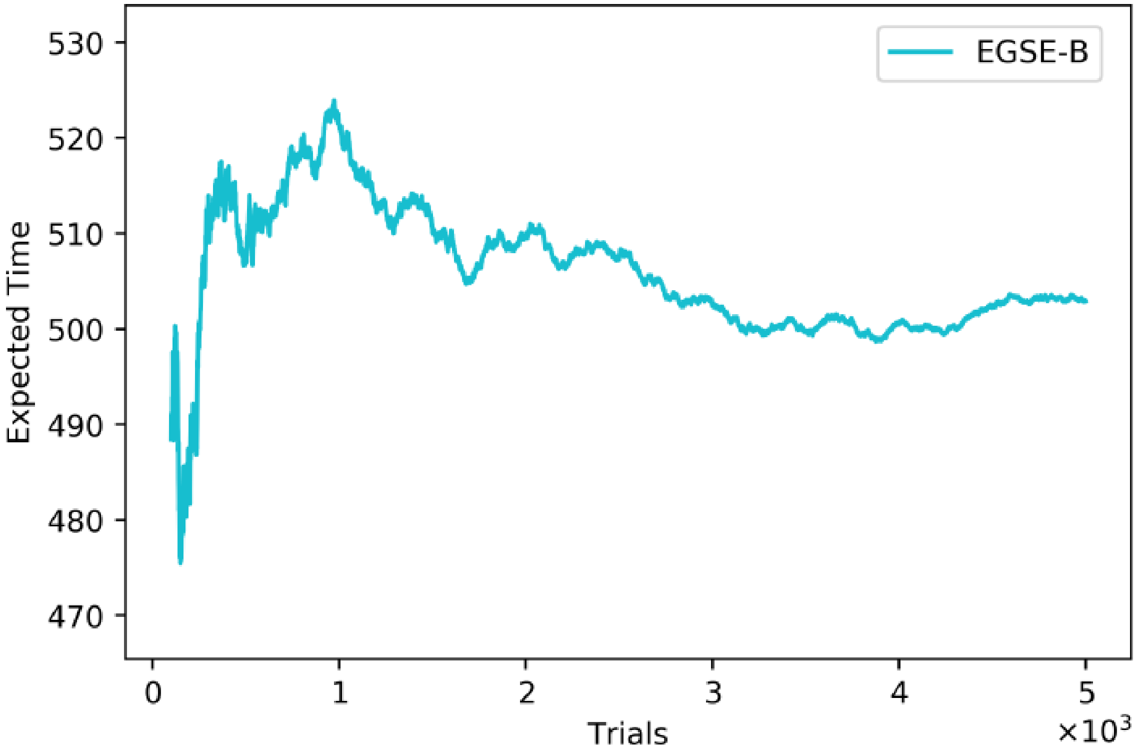
Experimental Evaluations

Expected Discovery Time of EGSE-A
Settings: N = 10,000, M = 100



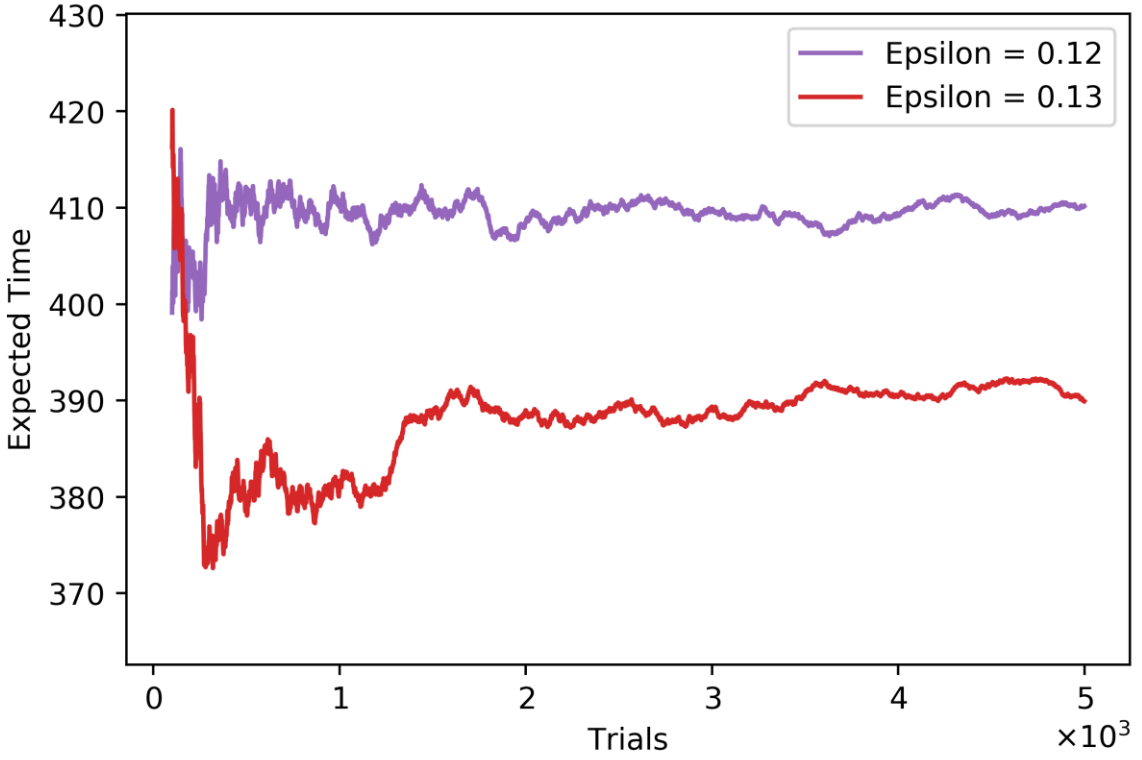
Experimental Evaluations

Expected Discovery Time of EGSE-B.
Settings: N = 10,000, M = 100, ε = 0.1.



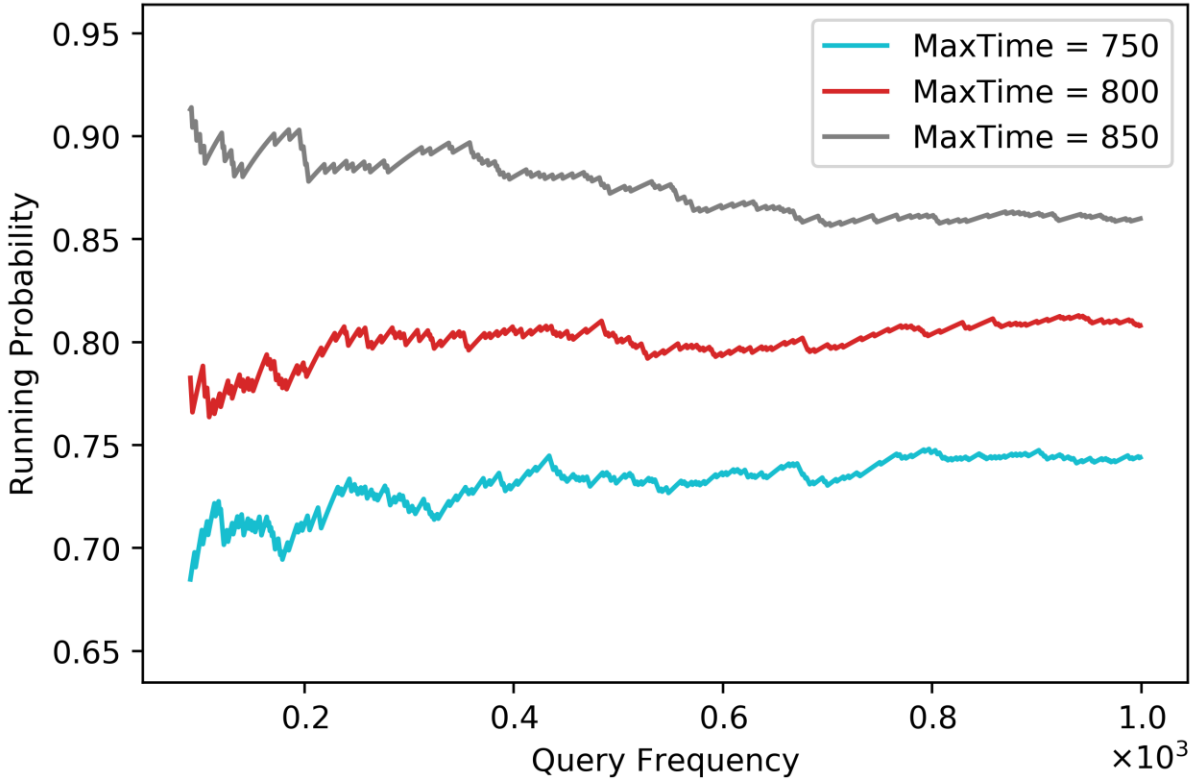
Experimental Evaluations

Expected Discovery Time of EGSE-B with
ε = 0.12, 0.13



Experimental Evaluations

Probability of discovering the most relevant object in EGSE-B with time constraints



Advantages

- Pure exploitation strategy can risk landing in a local optimum
- The ε-greedy strategy allows a balance between exploration and exploitation and avoids the local optimum problem
  - EGSE-A, which has the advantage of having greater fault-tolerant
  - EGSE-B, which has the advantage of sweeping the entire search space rapidly

Learning Convergence

- In a SLSE, it is important to be assured that the learning process will eventually lead to the correct terms being indexed
- Let $X(t)$ be the number of times that an unexplored index is being indexed (i.e. receive reinforcement) in the time interval $(0, t)$, with $\Pr[X(t, t+\varepsilon) > 1] = o(\varepsilon)$

Learning Convergence Behavior

- Let $dX(t)$ denote the number of times the unexplored index is indexed in the time interval $(t, t+dt)$, and $a(t) = E[dX(t)]/dt$
  - The value of $a(t)$ depends on actual usage, popularity and indexing frequency of the search engine
- Very often the point process is taken to be a stationary non-homogeneous process, and if further the point events are uncorrelated, we can take the probability that the unexplored index remaining unexplored in the time interval $(0, t)$ to be $\exp(-\alpha t)$

Learning Convergence Behavior

- Let S_t be the number of unexplored indexes in the system at time t, then it can be shown that

$$\mathbb{E}(S_t) = S_0 e^{-\alpha t},$$
$$V(S_t) = S_0 e^{-\alpha t}\left(1 - e^{-\alpha t}\right).$$

- Since the average of $S_t \to 0$ as $t \to \infty$, this indicates eventually the entire collection of unexplored indexes will be fully discovered

Learning Convergence Behavior

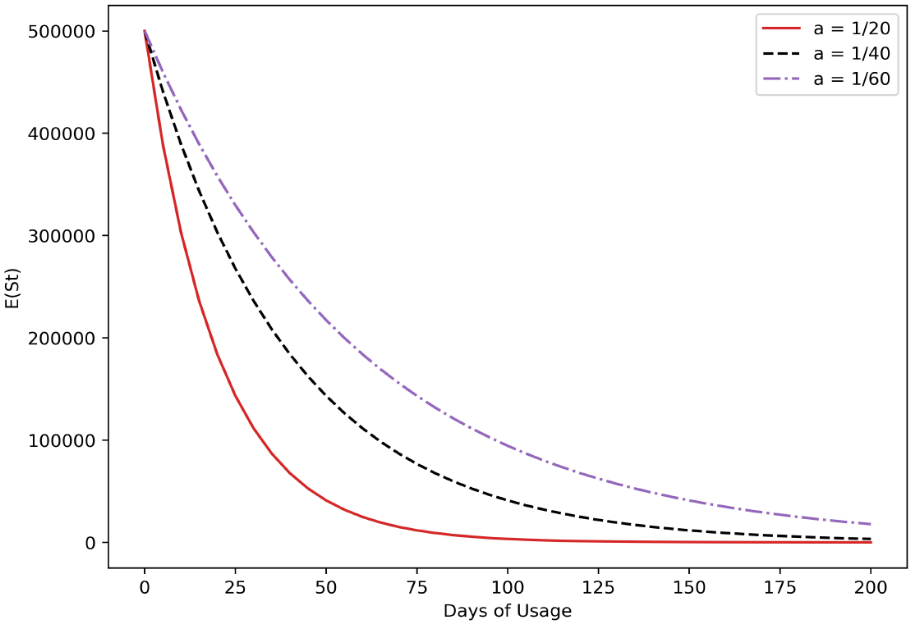
- Let T_s denote the expected time spent on indexing a proportion *p* of unexplored indexes

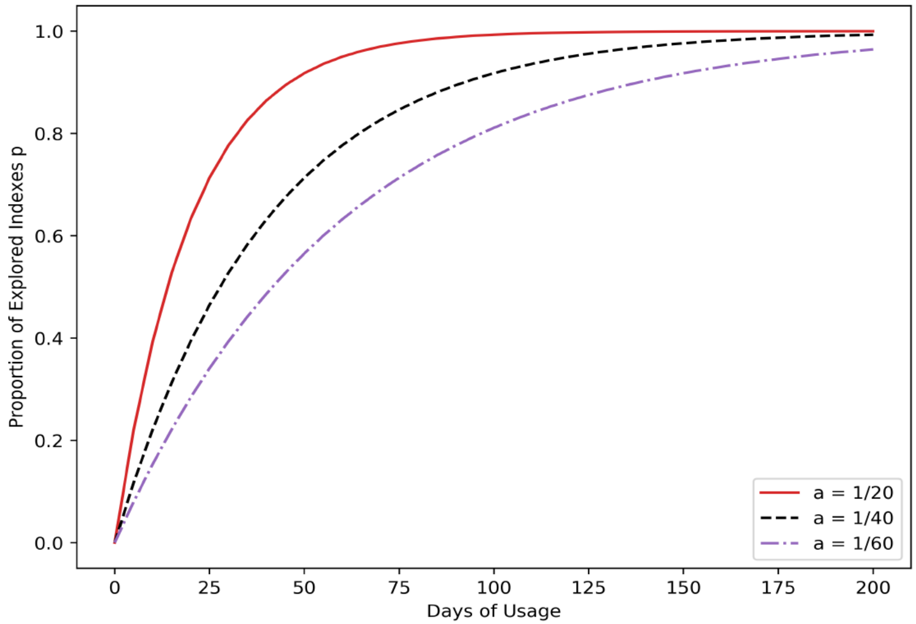
  $\Rightarrow$ the proportion of exposed explored indexes during a time interval T_s is

$$p = \frac{S_0 - S_0 e^{-\alpha T_s}}{S_0}.$$

Learning Convergence Behavior

$S_0 = 500{,}000$

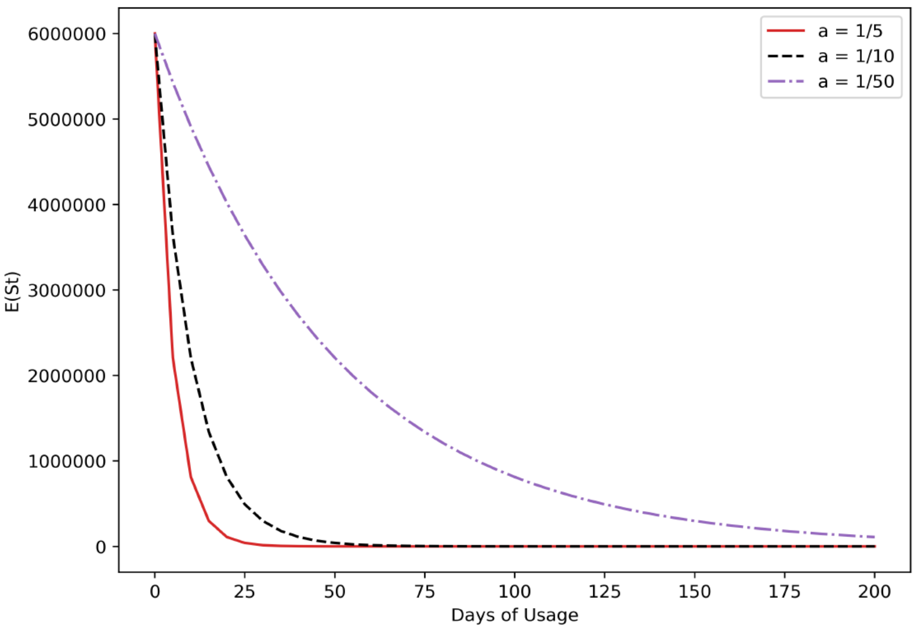


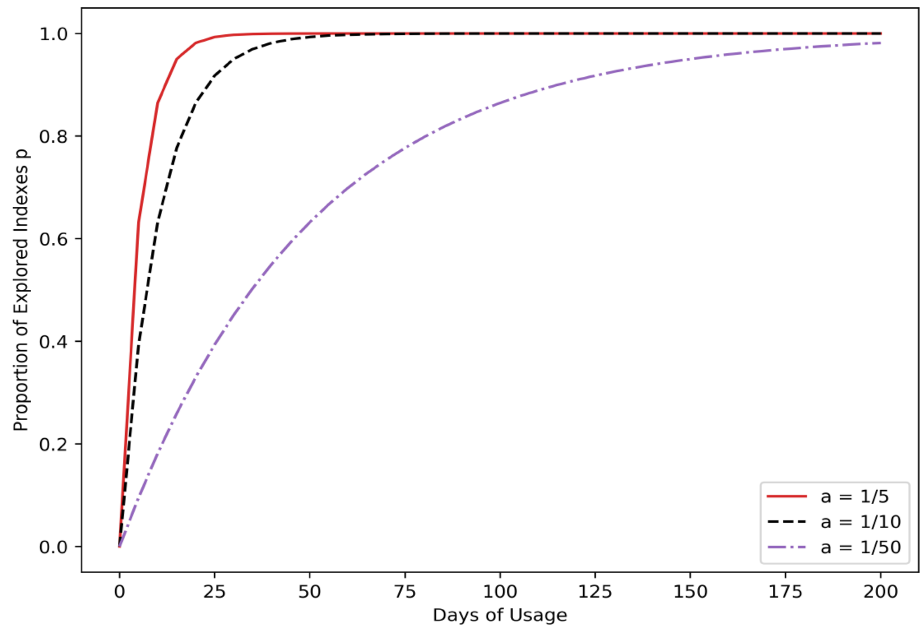


$\mathrm{E}(S_t)$ p

Learning Convergence Behavior

$S_0 = 6{,}000{,}000$



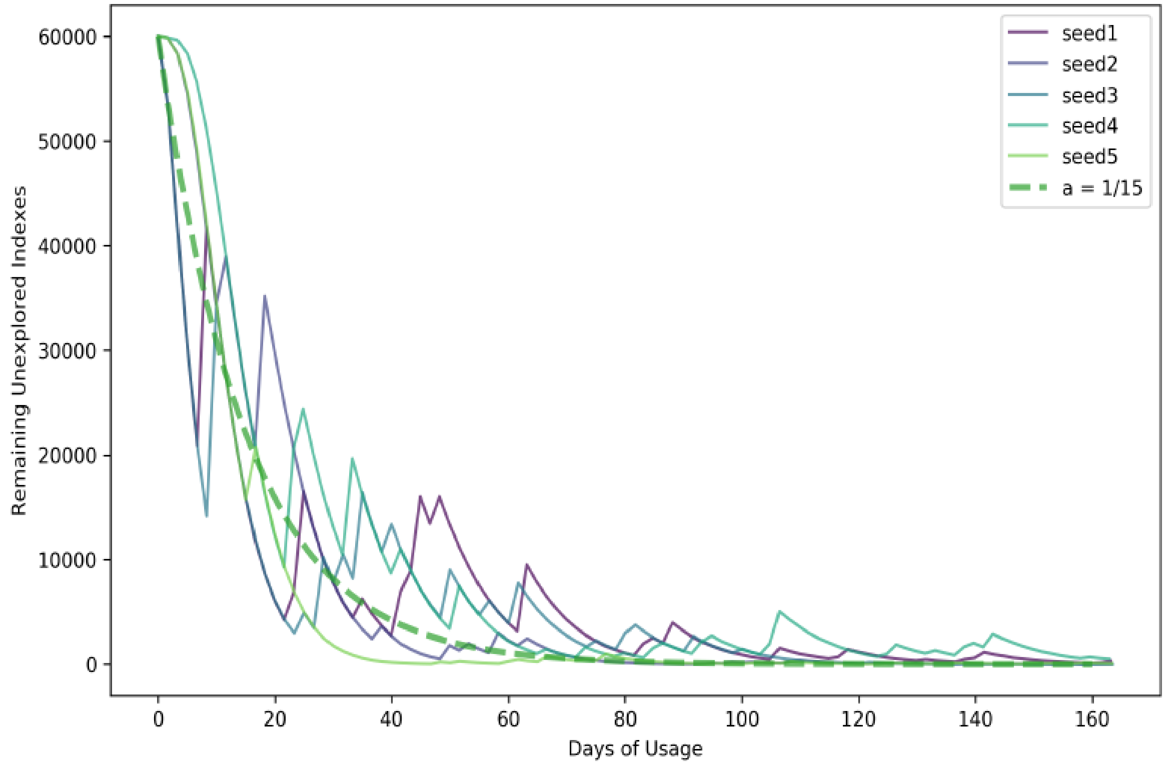


$\mathrm{E}(S_t)$

p

Experimental Evaluations

Changes of remaining unexplored indexes
λ = Poisson feedback rate

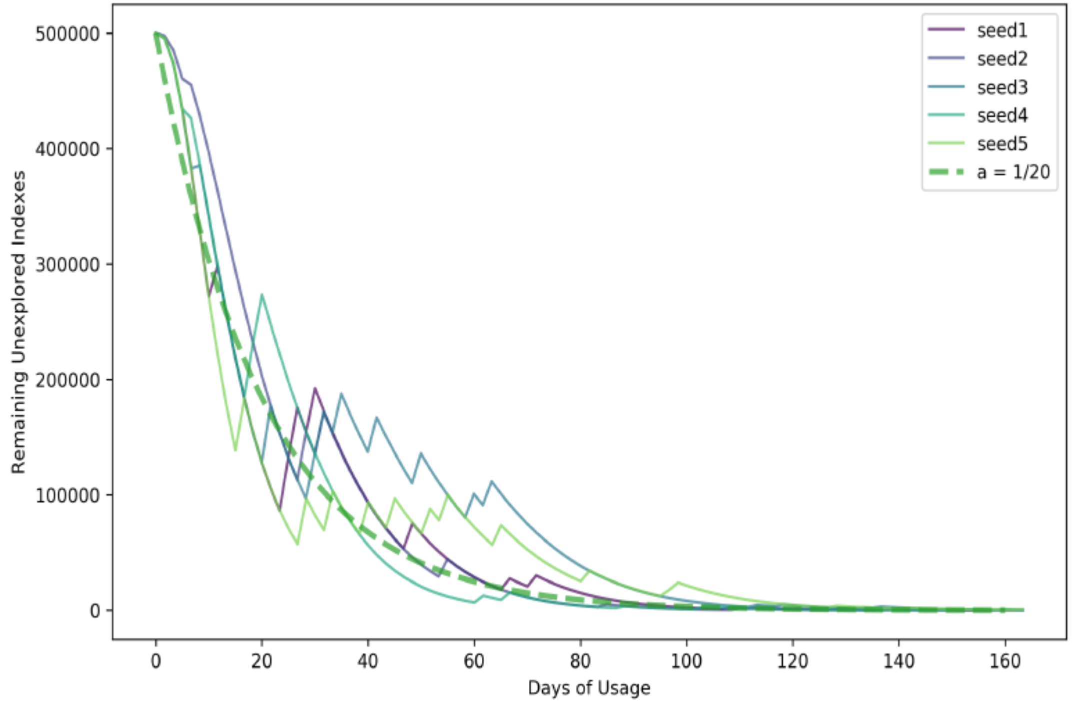


(a) S_0=60,000, λ=8,000, α=1/15

Green dotted line corresponds to simulation. Green line corresponds to theoretical results

Experimental Evaluations

Changes of remaining unexplored indexes
λ = Poisson feedback rate

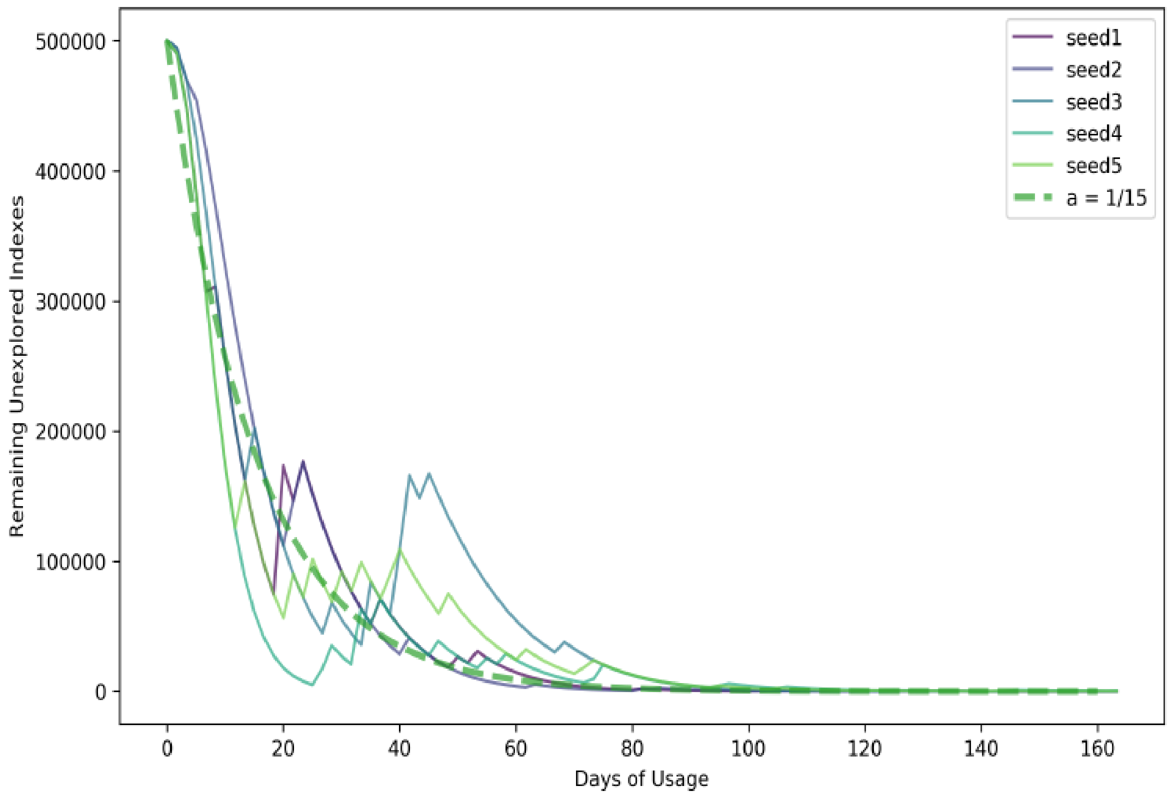


(b) S_0=500,000, λ=50,000, α=1/20

Green dotted line corresponds to simulation. Green line corresponds to theoretical results

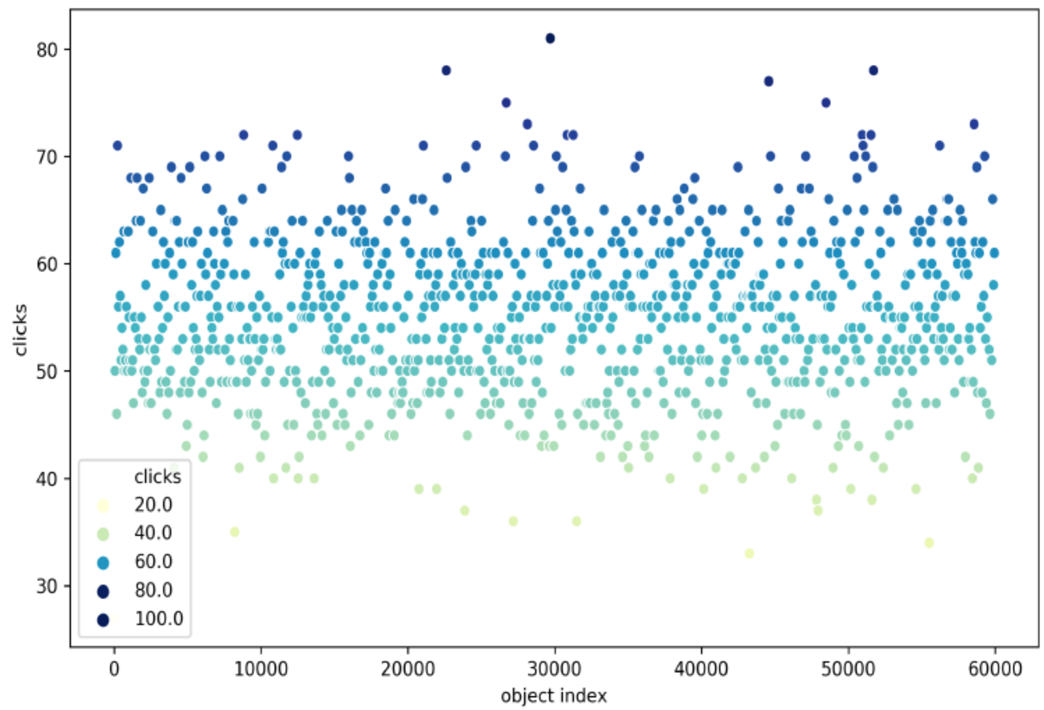
Experimental Evaluations

(c) S_0=500,000, λ=67,000, α=1/15



Experimental Evaluations

(d) Overview of no. of clicks when converged



Conclusion

- By applying reinforcement learning to a Self-Learning Search Engine within a Markov decision process framework, the subtle nuances of human perceptions and deep knowledge are gradually captured and learned
- Semantic indexes are built dynamically to interconnect search terms with the most relevant media entities and achieves steady improvements in search performance over time
- Learning convergence will be eventually achieved in the course of normal usage, indicating that the Self-Learning Search Engine architecture based on reinforcement learning is able to confer distinct advantages.


Thank You