Personalized Interactive Image Search from a Large Database: Bandits Meets Hashing

Aniruddha Bhargava Xin Hunt Kwang-Sung Jun Robert Nowak Rebecca Willet
University of Wisconsin - Madison

Introduction

Traditional image search systems are passive: given a query image x_0, they simply return similar images (e.g., nearest neighbors). We propose a personalized, interactive image search system where the search is guided by user feedback on the fly.

Datasets: images of shoes[1], images of galaxies, etc.
Feedback: a good starting point, a binary "yes" or "no" response
Goal: show as many images that elicit a "yes" response as possible
Comparisons with existing systems:

- Existing systems:
  - given a query, return nearest neighbors (NN). They learn better features to improve NN
  - ask curated questions (e.g., do you want shoes that are less feminine than this shoe?), needs experts and intensive batch training
- Our system:
  - uses feedback to learn what *aspect* of image is vital
  - asks a very *simple question* and does not need a curation or batch learning process.

Mathematical modeling: we have a set of feature vectors $\mathcal{X}$. At every time step, we can present one such $x \in \mathcal{X}$ and, we get back a "reward" of +1 if the user likes it or -1 if she doesn't. We model rewards $y_t \in \{+1, -1\}$ as

$$y_t = \langle x_t, \theta^* \rangle + \text{noise}$$

This is an instance of the **linear stochastic multi-armed bandit (MAB)** problem.

Challenges

Q: Can we directly apply existing linear MAB algorithms? **No.**

i. They have linear dependence on the number of images (N), which does not scale. E.g., $N=10^7$ in ImageNet. Terabytes of images in astronomy databases.
ii. No principled way to incorporate starting point x_0

Contributions

- The first scalable personalized, interactive image search system.
- A new algorithm called QOFUL that has good regret guarantees.
- A principled way to incorporate the initial image x_0.
- Real-world system evaluation with human feedback.

Image Features

Caffe ImageNet features[5,6]

- Deep learning based feature
- Model trained over the ImageNet dataset, which includes over 14 million images in various categories
- Eight layers of features in total: we use the seventh (length: 4096) and project down to 1000 dimensions

Scaling Up Bandits

What is the issue?

$\mathbf{X}_t := [\mathbf{x}_1^T; \cdots \mathbf{x}_t^T], \mathbf{y}_t := [y_1; \cdots; y_t]$ // design matrix (d x d) and reward vector (t x 1)

$\overline{\mathbf{V}}_t := \lambda \mathbf{I} + \sum_{s=1}^t \mathbf{x}_s \mathbf{x}_s^T$ // d x d

$\widehat{\theta}_t := \overline{\mathbf{V}}_t^{-1} \mathbf{X}_t \mathbf{y}_t$ // ridge regression estimator

Current state-of-the-art: **OFUL**

"exploitation" "exploration" $O(\sqrt{d \log(t)})$

$$\mathbf{x}_t^{\text{OFUL}} = \arg\max_{\mathbf{x} \in \mathcal{X}_t} \boxed{\langle \widehat{\boldsymbol{\theta}}_{t-1}, \mathbf{x} \rangle} + \sqrt{\beta_{t-1}} \cdot \boxed{||\mathbf{x}||_{\overline{\mathbf{V}}_{t-1}^{-1}}} \quad (1)$$

How to avoid N evaluations above?

- Subsample uniformly at random (naïve)
- Hashing has been successful for NN. Can we use hashing with OFUL?

Proposed Algorithm: QOFUL

QOFUL(Quadratic Optimism in the Face of Uncertainty for Linear rewards)

- **MIPS hash-amenable!**

$$\mathbf{x}_t^{\text{QOFUL}} = \max_{\mathbf{x} \in \mathcal{X}_t} \langle \widehat{\boldsymbol{\theta}}_{t-1}, \mathbf{x} \rangle + \frac{\beta_{t-1}^{1/4}}{4 c_1 m_{t-1}} \cdot ||\mathbf{x}||^2_{\overline{\mathbf{V}}_{t-1}^{-1}}$$
$$= \left\langle \begin{pmatrix} \widehat{\boldsymbol{\theta}}_{t-1} \\ \text{vec}\left(\frac{\beta_{t-1}^{1/4}}{4 c_1 m_{t-1}} \overline{\mathbf{V}}_{t-1}^{-1} \right) \end{pmatrix}, \begin{pmatrix} \mathbf{x} \\ \text{vec}(\mathbf{x}\mathbf{x}^\top) \end{pmatrix} \right\rangle$$

- Best regret bound among hash-amenable MAB algorithms

Algorithms	Regret	Hash-amenable	Time
OFUL [1]	$\tilde{O}(d\sqrt{T})$	✗	Nd^2
Rarely-switching OFUL [1]	$\tilde{O}(d\sqrt{T})$	✗	$d^2 + Nd + Nd^2(\log T)/T$
LTS [2]	$\tilde{O}(d^{3/2}\sqrt{T})$	✓	$d^2 + Nd$
QOFUL (ours)	$\tilde{O}(d^{5/4}\sqrt{T})$	✓	Nd^2

Bias Towards the Initial Image x_0

Naïve biased search (NBS)

1. Retrieve the first image by $\max_x \langle x, x_0 \rangle$ and no further biasing.
2. Treat x_0 as an arm with reward 1. Then, the first image is retrieved with (1).

Proposed (theoretically motivated):

- Let $\theta_0 = \frac{\mathbf{x}_0}{||\mathbf{x}_0||_2^2}$. Then, use $\widetilde{\boldsymbol{\theta}}_t := \boldsymbol{\theta}_0 + \widehat{\boldsymbol{\theta}}_t$ in place of $\widehat{\boldsymbol{\theta}}_t$.

Much stronger biasing than NBS1 and NBS2!

References

[1] *Fine-Grained Visual Comparisons with Local Learning, Yu and Grauman 2014*
[2] *Improved Algorithms for Linear Stochastic Bandits,* Abbasi-Yadikori et al., 2011
[3] *Thompon Sampling for Contextual Bandits with Linear Payoffs, Agarwal and Goyal ICML 2013*
[4] *NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning,* Jamieson et al., 2015
[5] *Caffe: Convolutional Architecture for Fast Feature Embedding, Jia et al., 2014*
[6] *ImageNet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, Dent et al., 2009*

Evaluation

All methods use our proposed biased search (unless indicated otherwise)

- -Light versions: subsample with rate p
- -Hash versions: use hashing to obtain p fraction. OFUL-Lazy-Hash is partially hashed.
- -NBS1, NBS2: naïve biased search shown above.

Algorithm	Time order	Wall-clock time	Red boots	Asics	Pre-walker	Over-the-knee	Boat	Toddler
NN	Nd/T	2	17.1±4.3	11.5±2.7	3.7±1.3	3.5±0.9	17.5±4.5	4.1±1.3
OFUL-NBS1	$d^2 + Nd$	522	25.3±5.9	15.3±5.7	4.9±2.8	1.6±1.1	34.9±3.7	6.0±1.7
OFUL-NBS2		522	37.7±1.3	29.4±2.4	14.0±2.3	4.1±1.2	34.5±3.8	8.4±2.2
OFUL		522	**39.5±1.2**	**31.9±3.2**	14.9±2.9	6.8±1.5	38.0±4.3	11.0±2.8
OFUL-Light (p=0.01)	pNd^2	56	25.5±1.8	14.2±2.1	5.6±1.3	3.5±0.9	24.7±3.8	6.8±1.8
OFUL-Light (p=0.02)		98	30.0±2.6	17.3±2.6	7.3±1.6	3.6±1.1	32.0±2.9	7.7±1.7
OFUL-Light (p=0.05)		222	34.4±2.6	23.1±3.5	10.2±2.2	5.3±1.4	34.8±3.6	8.5±1.9
QOFUL-NBS1	$d^2 + Nd$	522	25.3±5.9	15.3±5.7	4.9±2.8	1.6±1.1	34.9±3.7	6.0±1.7
QOFUL-NBS2		522	37.6±1.4	29.2±2.2	14.3±2.3	4.5±1.1	33.6±3.9	8.7±2.0
QOFUL		522	39.1±1.2	31.3±3.7	**15.1±2.8**	**7.7±1.4**	**38.7±3.7**	**11.3±2.8**
QOFUL-Light (p=0.01)	pNd^2	54	25.3±2.7	12.9±2.4	6.1±1.6	4.0±1.1	27.5±4.2	5.7±1.7
QOFUL-Light (p=0.02)		96	31.6±2.4	18.1±2.7	7.8±1.8	3.7±1.0	31.3±4.3	8.1±2.1
QOFUL-Light (p=0.05)		220	36.1±1.9	23.4±3.1	9.8±2.5	5.6±1.0	34.0±4.7	9.9±2.3
QOFUL-Hash (p=0.01)	$(pN + \ell k)d^2$	184	35.7±2.5	18.9±3.7	5.6±1.8	4.4±0.9	30.5±4.0	8.6±2.2
QOFUL-Hash (p=0.02)		232	35.1±3.3	22.9±3.4	9.2±2.3	4.9±1.2	34.4±4.1	9.3±2.2
QOFUL-Hash (p=0.05)		348	37.6±2.6	29.5±2.9	11.9±2.8	5.3±1.3	35.6±3.9	9.1±2.4
LTS-NBS1	$d^2 + Nd$	281	23.6±6.8	15.1±6.3	3.2±2.3	1.7±1.3	31.2±6.5	5.7±2.4
LTS-NBS2		281	36.9±3.1	24.2±6.0	11.1±3.0	5.1±1.7	35.7±4.9	8.9±2.5
LTS		281	**39.5±1.2**	**31.9±3.2**	14.9±2.9	6.8±1.5	38.0±4.3	11.0±2.8
LTS-Light (p=0.01)	$d^2 + pNd$	40	23.7±2.3	13.8±2.2	5.9±1.5	3.1±0.9	28.5±3.0	6.4±1.6
LTS-Light (p=0.02)		52	30.9±2.0	18.7±2.3	7.9±2.0	4.1±0.9	33.1±3.1	9.5±2.1
LTS-Light (p=0.05)		86	36.3±2.3	24.7±2.7	9.9±2.5	5.4±1.3	33.1±4.5	8.1±2.0
LTS-Hash (p=0.01)	$d^2 + (pN + \ell k)d$	54	37.4±2.1	25.9±3.4	8.3±2.5	5.1±1.3	36.1±3.9	9.6±2.2
LTS-Hash (p=0.02)		74	38.4±1.8	28.5±3.2	9.4±2.4	5.1±1.3	36.2±4.4	9.2±2.1
LTS-Hash (p=0.05)		136	38.8±1.8	29.0±3.6	13.2±2.8	6.4±1.5	37.5±3.8	9.9±2.5
OFUL-Lazy-Light (p=0.01)	$pN(d + d^2(\log T)/T)$	36	23.2±2.2	13.9±2.3	5.3±1.3	3.4±0.8	24.6±3.8	5.7±1.5
OFUL-Lazy-Light (p=0.02)		54	29.9±2.2	17.8±2.6	6.7±1.9	3.8±0.9	29.7±3.3	7.1±1.6
OFUL-Lazy-Light (p=0.05)		114	34.3±2.2	21.8±4.0	9.9±2.2	4.8±1.3	30.9±4.7	7.7±1.8
OFUL-Lazy-Hash (p=0.01)		44	32.9±2.5	20.6±2.4	9.6±2.0	3.8±0.9	30.5±4.3	7.1±2.0
OFUL-Lazy-Hash (p=0.02)		62	36.0±1.5	22.1±3.0	9.5±2.5	5.1±1.1	34.2±3.3	9.5±2.1
OFUL-Lazy-Hash (p=0.05)		130	36.3±2.2	27.3±3.1	11.9±2.7	5.1±1.3	32.3±4.8	8.5±2.1

# of relevant images retrieved out of 50 iterations.

- Data: Zappos50k. N=50k, d=1000
- All interactive algorithms outperform NN
- Hashing versions perform better than naïve random sampling (light) versions
- Proposed biased search outperforms both NBS1 and NBS2
- Ignore time, then OFUL, LTS, and QOFUL work equally well.
- Under time constraints, LTS-Hash works the best.

Real user experiments using NEXT

- Platform for active learning
- Handles interaction with users and algorithms can be plugged in
- Deployable on Amazon EC2