Finding a personal needle in a haystack: bandit algorithms and image search

Aniruddha Bhargava  Xin Jiang  Robert Nowak  Rebecca Willet
University of Wisconsin - Madison

**Introduction**

Image search is becoming personalized. This can be because the application demands it: astronomer looking for a particular type of galaxy or, for user experience: online shopping.

**User**

![Image dataset](http://image.net)

Is this shoe close to what you are looking for?

\[ \theta^* \]

\[^{\text{D}}_{\text{user}} \]

**Datasets:** images of shoes, images of galaxies, etc. Set might be getting bigger over time

**Feedback:** a good starting point, a binary “yes” or “no” response

**Goal:** show as many images that illicit a “yes” response as possible

Mathematically we have a set of feature vectors \( \chi \). At every time step, we can present one such \( x \in \chi \) and, we get back a “reward” of +1 if the user likes it or -1 if she doesn’t. This is similar to what is looked at in linear bandits.

**Bandit model:** Reward \( r_t = \langle x_t, \theta^* \rangle + \epsilon \)

Some bandit algorithms and their merits and demerits:

- **UE, Tatsiklis:** first such algorithm, easy to run if we already have rewards a set of linearly independent feature vectors
- **OFUL:** tight bounds, smoothly blends exploration and exploitation, depends on ambient dimension, not that of the subspace
- **Smooth Explore:** similar to smooth explorer but the greediest version of it
- **SL-UCB:** designed for sparse \( \theta \), wasteful initially to estimate support
- **Online-to-confidence sets:** online, can work with any black-box algorithm with regret guarantees, cited sparse algorithm is hard to implement

**Image Features**

**Gist descriptors:**
- A 512-length feature vector that captures the “gist” of the scene
- Uses 32 Gabor filters to produce gradient maps and then smoothed over each map to reduce dimensionality

**SIFT features + spatial pyramid:**
- A popular choice in image classification
- Partition the image into increasingly fine sub-regions and compute histograms of local features (SIFT) found inside each sub-region
- Feature length depends on the number of layers in the pyramid

**Caffe ImageNet features:**
- State of the art, extremely popular
- 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 eighth (length: 1000) layers

**Synthetic data for Bandits**

Two types of data for the experiments on bandits:
- Purely synthetic: once the ambient dimension \( d \) and the number of arms \( d \) were fixed, all the data points were taken from the unit sphere in \( d \)-dimensions. \( \theta^* \) was also picked from the unit sphere.
- Semi-synthetic: the \( n \) arms were chosen from a dataset (MNIST or Zappos) and a random feature vector from the same set was chose to be \( \theta^* \).

**Experiments and Results**

**Objective:** Find boots from all the images
- Moderate number of positive samples (13k/50k)
- Relatively distinct features with less confusion with other categories
- \( l_1 \) Penalized logistic regression
- Maximum false alarm rate set to about 3%

**Results:**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>GIST Features</th>
<th>SIFT + SP</th>
<th>Caffe Layer 7</th>
<th>Caffe Layer 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>400</td>
<td>5.83%</td>
<td>5.07%</td>
<td>4.56%</td>
</tr>
<tr>
<td>Max P = ( P_0 )</td>
<td>85.5%</td>
<td>85.8%</td>
<td>90.1%</td>
<td>87.1%</td>
</tr>
<tr>
<td>P = ( P_1 )</td>
<td>2.84%</td>
<td>2.86%</td>
<td>2.80%</td>
<td>2.82%</td>
</tr>
<tr>
<td>Sample size</td>
<td>80</td>
<td>9.10%</td>
<td>7.22%</td>
<td>5.70%</td>
</tr>
<tr>
<td>Max P = ( P_0 )</td>
<td>73.7%</td>
<td>81.1%</td>
<td>85.4%</td>
<td>81.3%</td>
</tr>
<tr>
<td>P = ( P_1 )</td>
<td>3.15%</td>
<td>3.20%</td>
<td>2.64%</td>
<td>3.24%</td>
</tr>
</tbody>
</table>

**Bandit algorithm observations:**
- The algorithms work well when they are given the inner products
- The algorithms have a very hard time in high dimension even when a good starting point is given
- Empirically the algorithms show promise when we can pull the arms only once

**References**

1. Linearly Parameterized Bandits, Rusmevichientong and Tsitsiklis 2010
2. Improved Algorithms for Linear Stochastic Bandits, Abbasi-Yadkori et al 2011
3. Linear Bandits in High Dimensions and Recommendation Systems, Deshpande and Montanari 2012
4. Bandit Theory meets Compressed Sensing for high-dimensional Stochastic Linear Bandits, Caringer and Mannor 2012
5. Online-to-Confidence Set Constructions and Application to Sparse Stochastic Bandits, Abbasi-Yadkori et al 2012
7. Modeling the shape of the scene: A holistic representation of the spatial envelope, Oliva and Torralba 2001
8. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, Lazebnik et al 2006