Parallel Gaussian Bandits

Johannes Verwijnen
(jverwijn@cs.helsinki.fi)
16.4.2014
Bandits

- A multi-armed bandit captures the tradeoff between exploration and exploitation
- How do we maximize the total expected reward?
- Used for dynamic effort allocation

Image credit:
David Tolpin, Solomon Eyal Shimony
Ben Gurion University of the Negev,
Beer Sheva, Israel
http://www.offtopia.net/ecai-2012-vago-poster/
Different types of bandits

- Deterministic bandits
- Stochastic bandits
- Restless bandits
- Adversial bandits
- Arm-acquiring
- Switching penalty
- Correlated bandits

Image credit: Buffalo Games, Jorge Baeza
http://calhoonplay.com/?q=node/18
Independent bandits

- For results in solving the basic case of independent bandits, one should look at Gittins indices [1] and Lai and Robbins [2].
- Auer et al [3] introduced the simple and efficient UCB algorithms.
- Regret bounds are the usual way to evaluate the policies:

\[ R_T = \sum_t f(x^*) - f(x_t) \]
Dependent arm bandits

- Arms are not always independent
  - Amazon buy-next display or other internet advertising
  - Finding the sensors to activate for local measurement
- How do we model this prior knowledge?

Image credit: Unknown
http://karlalant.com/2013/01/26/individuality-interdependence-and-motivation/
Gaussian Processes

• A generalization of the Gaussian probability distribution
• A collection of independent, normally distributed random variables
• Any finite subset of this collection has a multivariate normal distribution.
• If mean is zero, defined only by covariance kernel.

Image credit: John Reid
http://sysbio.mrc-bsu.cam.ac.uk/johns/infpy/docs/build/html/gps.html
Why Gaussian Processes?

- Simple!
- Noisy sample of rewards $y_{1:t-1} = [y, \ldots, y_{t-1}]$ at points $X = \{x_1, \ldots, x_{t-1}\}$ with Gaussian noise $\epsilon_t \sim N(0, \sigma^2)$

the posterior over $f$ is a Gaussian

$$f(x) \mid y_{1:t-1} \sim N(\mu_{t-1}(x), \sigma_{t-1}^2(x))$$

where

$$\mu_{t-1}(x) = k \left[ K + \sigma^2 I \right]^{-1} y_{1:t-1}$$

$$\sigma_{t-1}^2(x) = k(x, x) - k \left[ K + \sigma^2 I \right]^{-1} k^T$$
The GP-UCB selection rule [4]

\[ x_t = \arg \max_{x \in X} \left( \mu_{t-1}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(x) \right) \]

- consists of the exploitation part (left) and the exploration part (right)
- Beta is the balancing factor and its optimum shown to depend on the kernel function, t and |X|
The algorithm itself is quite simple:

for \( t = 1,2,\ldots \) do

Choose

\[ x_t = \arg \max_{x \in \mathcal{X}} \left( \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x) \right) \]

Sample

\[ y_t = f(x_t) + \epsilon_t \]

Perform Bayesian update to obtain new mu and sigma

end for

GP-UCB algorithm
GP-UCB results

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Linear</th>
<th>RBF</th>
<th>Matérn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regret $R_T$</td>
<td>$d\sqrt{T}$</td>
<td>$\sqrt{T(\log T)^{d+1}}$</td>
<td>$\frac{\nu+d(d+1)}{T^{2\nu+d(d+1)}}$</td>
</tr>
</tbody>
</table>

Figure 2. (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b,c) Two iterations of the GP-UCB algorithm. It samples points that are either uncertain (b) or have high posterior mean (c).
Parallel

- Finding the reward may take time
- Let’s sample many decisions
- How do we choose the batch?

Image credit:
Argonne National Laboratory
omputer.jpg
Batch selection

- We only have feedback on the previous decisions.
- We need to find a batch of size $B$,
- GP-BUCB uses GP-UCB on limited feedback information [5]
  - also introduces trick for variance calculation
- GP-UCB-PE is a more explorative version [6]
  - introduces a relevant region for explicit exploration
The algorithm itself is quite simple:

for t = 1, 2, … do

Choose

Compute sigma

if t = fb(t+1) then

Obtain

Perform Bayesian update to obtain new mu

end if

end for

\[
x_t = \arg \max_{x \in \mathcal{X}} \left( \mu_{t-1}(x) + \beta_t^{1/2} \sigma_{t-1}(x) \right)
\]

\[
y_{t'} = f(x_{t'}) + \epsilon_{t'} \quad t' \in \{fb[t], \ldots, t\}
\]
GP-BUCB hallucination effect

Figure 1. (a): The confidence intervals $C_{fb[t]}^{seq}(x)$ (dark), computed from previous noisy observations (crosses), are centered around the posterior mean (solid black) and contain $f(x)$ (white dashed) w.h.p. To avoid overconfidence, GP-BUCB chooses $C_{fb[t]}^{batch}(x)$ (light gray) such that even in the worst case, $C_{fb[t]}^{batch}(x)$ will contain $C_{fb[t]}^{seq}(x)$. (b): Due to the observations that GP-BUCB “hallucinates” (stars), the outer posterior confidence intervals $C_{fb[t]}^{batch}(x)$ shrink from their values at the start of the batch (black dashed), but still contain $C_{fb[t]}^{seq}(x)$, as desired. (c): Upon selection of the last decision of the batch, the feedback for all decisions is obtained, and new confidence intervals $C_{fb[t']}^{seq}(x)$ and corresponding $C_{fb[t']}^{batch}(x)$ are computed.
GP-BUCB results

- Regret bounds independent of batch size (see paper for details)
• Here we find a confidence region (from GP-UCB) that f is included with high probability:

\[ \hat{f}_{t-1}^+(x) = (\mu_{t-1}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(x)) \]

\[ \hat{f}_{t-1}^-(x) = (\mu_{t-1}(x) - \beta_t^{\frac{1}{2}} \sigma_{t-1}(x)) \]

• Then we define a relevant region, which contains the argmax with high probability with lower bound:

\[ y_t^* = \hat{f}_t^-(x_t^*), \text{ where } x_t^* = \arg \max_{x \in \mathcal{X}} \hat{f}_t^-(x) \]

giving

\[ \mathcal{R}_{t-1}^+ = \{ x \in \mathcal{X} | \mu_{t-1}(x) + 2\sqrt{\beta_t \sigma_{t-1}(x)} \geq y_{t-1}^* \} \]
Fig. 2. Two queries of GP-UCB-PE on the previous example. The lower confidence bound on the maximum is represented by the horizontal dotted green line at $y_i^\ast$. The relevant region $\mathcal{R}$ is shown in light green (without edges). The first query $x^0$ is the maximizer of $\widehat{f}^+$. We show in dashed line the upper and lower bounds with the update of $\widehat{\sigma}$ after having selected $x^0$. The second query $x^1$ is the one maximizing the uncertainty inside $\mathcal{R}^+$, an extension of $\mathcal{R}$ which is not illustrated here.
GP-UCB-PE algorithm

- For $t = 0 \ldots T$ do
  
  Compute $\mu$ and $\sigma$ using Bayesian inference
  Choose first location according to GP-UCB

  for $k = 1, \ldots, B-1$ do
    Compute $\sigma$ using Bayesian inference
    Choose argmax of $\sigma$ from the relevant region
  end for

  Query rewards

end for
Results

Fig. 4. Experiments on several real and synthetics tasks. The curves show the decay of the mean of the simple regret $r^K_t$ with respect to the iteration $t$, over 64 experiments. We show with the translucent area the confidence intervals.
Empirical results

- JAVA and Matlab implementation
- using standard data until now
- Also thinking on including Monte Carlo-based SM-UCB and SM-MEI
Discussion
References


References

