Introduction to Optimization

Global & Bayesian Optimization

Multi-armed bandits, exploration vs. exploitation, navigation through belief space, upper confidence bound (UCB), global optimization = infinite bandits, Gaussian Processes, probability of improvement, expected improvement, UCB

Marc Toussaint
University of Stuttgart
Summer 2014
Global Optimization

- Is there an optimal way to optimize (in the Blackbox case)?
- Is there a way to find the *global* optimum instead of only local?
Outline

- Play a game

- Multi-armed bandits
  - Belief state & belief planning
  - Upper Confidence Bound (UCB)

- Optimization as infinite bandits
  - GPs as belief state

- Standard heuristics:
  - Upper Confidence Bound (GP-UCB)
  - Maximal Probability of Improvement (MPI)
  - Expected Improvement (EI)
Bandits
Bandits

- There are $n$ machines.
- Each machine $i$ returns a reward $y \sim P(y; \theta_i)$
  The machine’s parameter $\theta_i$ is unknown
Bandits

• Let $a_t \in \{1, \ldots, n\}$ be the choice of machine at time $t$
  Let $y_t \in \mathbb{R}$ be the outcome with mean $\langle y_{a_t} \rangle$

• A policy or strategy maps all the history to a new choice:

  $$\pi : [(a_1, y_1), (a_2, y_2), \ldots, (a_{t-1}, y_{t-1})] \mapsto a_t$$

• Problem: Find a policy $\pi$ that

  $\max \langle \sum_{t=1}^{T} y_t \rangle$

  or

  $\max \langle y_T \rangle$

  or other objectives like discounted infinite horizon $\max \langle \sum_{t=1}^{\infty} \gamma^t y_t \rangle$
Exploration, Exploitation

• “Two effects” of choosing a machine:
  – You collect more data about the machine $\rightarrow$ knowledge
  – You collect reward

• Exploration: Choose the next action $a_t$ to $\min \langle H(b_t) \rangle$

• Exploitation: Choose the next action $a_t$ to $\max \langle y_t \rangle$
The Belief State

- “Knowledge” can be represented in two ways:
  - as the full history
    \[ h_t = [(a_1, y_1), (a_2, y_2), ..., (a_{t-1}, y_{t-1})] \]
  - as the belief
    \[ b_t(\theta) = P(\theta|h_t) \]
    where \( \theta \) are the unknown parameters \( \theta = (\theta_1, .., \theta_n) \) of all machines

- In the bandit case:
  - The belief factorizes \( b_t(\theta) = P(\theta|h_t) = \prod_i b_t(\theta_i|h_t) \)
    e.g. for Gaussian bandits with constant noise, \( \theta_i = \mu_i \)
    \[ b_t(\mu_i|h_t) = \mathcal{N}(\mu_i|\hat{y}_i, \hat{s}_i) \]
    e.g. for binary bandits, \( \theta_i = p_i \), with prior \( \text{Beta}(p_i|\alpha, \beta) \):
    \[ b_t(p_i|h_t) = \text{Beta}(p_i|\alpha + a_{i,t}, \beta + b_{i,t}) \]
    \[ a_{i,t} = \sum_{s=1}^{t-1}[a_s=i][y_s=0], \quad b_{i,t} = \sum_{s=1}^{t-1}[a_s=i][y_s=1] \]
The Belief MDP

- The process can be modelled as

\[
\begin{align*}
a_1 & \rightarrow y_1 \rightarrow \theta \\
a_2 & \rightarrow y_2 \rightarrow \theta \\
a_3 & \rightarrow y_3 \rightarrow \theta \\
& \vdots
\end{align*}
\]

or as Belief MDP

\[
\begin{align*}
a_1 & \rightarrow y_1 \rightarrow b_0 \\
a_2 & \rightarrow y_2 \rightarrow b_1 \\
a_3 & \rightarrow y_3 \rightarrow b_2 \\
& \vdots
\end{align*}
\]

\[
P(b' | y, a, b) = \begin{cases} 
1 & \text{if } b' = b[a, y] \\
0 & \text{otherwise}
\end{cases}
, \quad P(y | a, b) = \int_{\theta_a} b(\theta_a) \ P(y | \theta_a)
\]

- The Belief MDP describes a different process: the interaction between the information available to the agent \((b_t \text{ or } h_t)\) and its actions, where the agent uses his current belief to anticipate observations, \(P(y | a, b)\).

- The belief (or history \(h_t\)) is all the information the agent has available; \(P(y | a, b)\) the “best” possible anticipation of observations. If it acts optimally in the Belief MDP, it acts optimally in the original problem.

**Optimality in the Belief MDP \implies \text{ optimality in the original problem}**
Optimal policies via Belief Planning

- The Belief MDP:

  \[ P(b' | y, a, b) = \begin{cases} 1 & \text{if } b' = b[a, y] \\ 0 & \text{otherwise} \end{cases}, \quad P(y | a, b) = \int_{\theta_x} b(\theta_x) P(y | \theta_x) \]

- Belief Planning: Dynamic Programming on the value function

  \[ V_{t-1}(b_{t-1}) = \max_{\pi} \left\langle \sum_{t=1}^{T} y_t \right\rangle 
  = \max_{a_t} \int_{y_t} P(y_t | a_t, b_{t-1}) \left[ y_t + V_t(b_{t-1}[a_t, y_t]) \right] \]
Optimal policies

- The value function assigns a value (maximal achievable return) to a state of knowledge
- The optimal policy is greedy w.r.t. the value function (in the sense of the $\max_{a_t}$ above)
- Computationally heavy: $b_t$ is a probability distribution, $V_t$ a function over probability distributions

- The term $\int_{y_t} P(y_t|a_t, b_{t-1}) \left[ y_t + V_t(b_{t-1}[a_t, y_t]) \right]$ is related to the Gittins Index: it can be computed for each bandit separately.
Example exercise

- Consider 3 binary bandits for \( T = 10 \).
  - The belief is 3 Beta distributions \( \text{Beta}(p_i|\alpha + a_i, \beta + b_i) \rightarrow 6 \) integers
  - \( T = 10 \rightarrow \) each integer \( \leq 10 \)
  - \( V_t(b_t) \) is a function over \( \{0, \ldots, 10\}^6 \)

- Given a prior \( \alpha = \beta = 1 \),
  a) compute the optimal value function and policy for the final reward and the average reward problems,
  b) compare with the UCB policy.
Greedy heuristic: Upper Confidence Bound (UCB)

1: Initialization: Play each machine once
2: repeat
3: Play the machine $i$ that maximizes $\hat{y}_i + \sqrt{\frac{2 \ln n}{n_i}}$
4: until

$\hat{y}_i$ is the average reward of machine $i$ so far
$n_i$ is how often machine $i$ has been played so far
$n = \sum_i n_i$ is the number of rounds so far

UCB algorithms

- UCB algorithms determine a confidence interval such that

$$\hat{y}_i - \sigma_i < \langle y_i \rangle < \hat{y}_i + \sigma_i$$

with high probability.

UCB chooses the upper bound of this confidence interval

- Optimism in the face of uncertainty

- Strong bounds on the regret (sub-optimality) of UCB (e.g. Auer et al.)
Further reading


  Optimal Value function is submodular.
Conclusions

• The bandit problem is an archetype for
  – Sequential decision making
  – Decisions that influence knowledge as well as rewards/states
  – Exploration/exploitation

• The same aspects are inherent also in global optimization, active learning & RL

• Belief Planning in principle gives the optimal solution

• Greedy Heuristics (UCB) are computationally much more efficient and guarantee bounded regret
Global Optimization
Global Optimization

• Let $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x f(x)$$

(I neglect constraints $g(x) \leq 0$ and $h(x) = 0$ here – but could be included.)

• Blackbox optimization: find optimum by sampling values $y_t = f(x_t)$

No access to $\nabla f$ or $\nabla^2 f$

Observations may be noisy $y \sim \mathcal{N}(y \mid f(x_t), \sigma)$
Global Optimization = infinite bandits

- In global optimization $f(x)$ defines a reward for every $x \in \mathbb{R}^n$
  - Instead of a finite number of actions $a_t$ we now have $x_t$

- Optimal Optimization could be defined as: find $\pi : h_t \mapsto x_t$ that

$$\min \left\langle \sum_{t=1}^{T} f(x_t) \right\rangle$$

or

$$\min \left\langle f(x_T) \right\rangle$$
Gaussian Processes as belief

• The unknown “world property” is the function $\theta = f$
• Given a Gaussian Process prior $GP(f | \mu, C)$ over $f$ and a history

\[ D_t = [(x_1, y_1), (x_2, y_2), ..., (x_{t-1}, y_{t-1})] \]

the belief is

\[ b_t(f) = P(f | D_t) = GP(f | D_t, \mu, C) \]

Mean($f(x)$) = $\hat{f}(x) = \kappa(x)(K + \sigma^2 I)^{-1}y$

Var($f(x)$) = $\hat{\sigma}(x) = k(x, x) - \kappa(x)(K + \sigma^2 I_n)^{-1}\kappa(x)$

• Side notes:
  – Don’t forget that $\text{Var}(y^* | x^*, D) = \sigma^2 + \text{Var}(f(x^*) | D)$
  – We can also handle discrete-valued functions $f$ using GP classification
Optimal optimization via belief planning

- As for bandits it holds

\[ V_{t-1}(b_{t-1}) = \max_{\pi} \left\langle \sum_{t=t}^{T} y_t \right\rangle \]

\[ = \max_{x_t} \int_{y_t} P(y_t|x_t, b_{t-1}) \left[ y_t + V_t(b_{t-1}[x_t, y_t]) \right] \]

\( V_{t-1}(b_{t-1}) \) is a function over the GP-belief!

If we could compute \( V_{t-1}(b_{t-1}) \) we “optimally optimize”

- I don’t know of a minimalistic case where this might be feasible
Conclusions

• Optimization as a problem of
  – Computation of the belief
  – Belief planning

• Crucial in all of this: the prior $P(f)$
  – GP prior: smoothness; but also limited: only local correlations!
    No “discovery” of non-local/structural correlations through the space
  – The latter would require different priors, e.g. over different function classes
Heuristics
1-step heuristics based on GPs

- Maximize Probability of Improvement (MPI)
  \[ x_t = \arg\max_x \int_{-\infty}^{y^*} N(y|\hat{f}(x), \hat{\sigma}(x)) \]

- Maximize Expected Improvement (EI)
  \[ x_t = \arg\max_x \int_{-\infty}^{y^*} N(y|\hat{f}(x), \hat{\sigma}(x)) (y^* - y) \]

- Maximize UCB
  \[ x_t = \arg\max_x \hat{f}(x) + \beta_t \hat{\sigma}(x) \]

(Often, \( \beta_t = 1 \) is chosen. UCB theory allows for better choices. See Srinivas et al. citation below.)
Each step requires solving an optimization problem

- Note: each argmax on the previous slide is an optimization problem
- As $\hat{f}, \hat{\sigma}$ are given analytically, we have gradients and Hessians. BUT: multi-modal problem.
- In practice:
  - Many restarts of gradient/2nd-order optimization runs
  - Restarts from a grid; from many random points

- We put a lot of effort into carefully selecting just the next query point
Fig. 2. (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b) and (c) Two iterations of the GP-UCB algorithm. The dark curve indicates the current posterior mean, while the gray bands represent the upper and lower confidence bounds which contain the function with high probability. The “+” mark indicates points that have been sampled before, while the “o” mark shows the point chosen by the GP-UCB algorithm to sample next. It samples points that are either (b) uncertain or have (c) high posterior mean.
Fig. 6. Mean average regret: GP-UCB and various heuristics on (a) synthetic and (b, c) sensor network data.

Fig. 7. Mean minimum regret: GP-UCB and various heuristics on (a) synthetic, and (b, c) sensor network data.
Further reading

- Classically, such methods are known as *Kriging*


Bayesian Global Optimization

- Global Optimization with gradient information
  \[\rightarrow\] Gaussian Processes with derivative observations