Bayesian Optimization With Censored Response Data

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. University of British Columbia, CS department, Vancouver, Canada.

Frank

Holger

Kevin

Cost-varying function minimization

Want to minimize a blackbox function f:Θ→R

Observation: *cost* of evaluating f(θ) often depends on θ

- E.g.: clinical studies (θ specifies a drug)
- E.g.: learning deep networks (θ includes #layers, #neurons/layer, etc)

Def. A *cost-varying* function minimization problem is a tuple (f,c) of blackbox functions f:Θ→R, c:Θ→R. The *budget* for minimizing f is a limit on the cumulative cost of function evaluations.

Cost-monotonicity

Often, we can cut off function evaluations early and get a lower bound on the function value

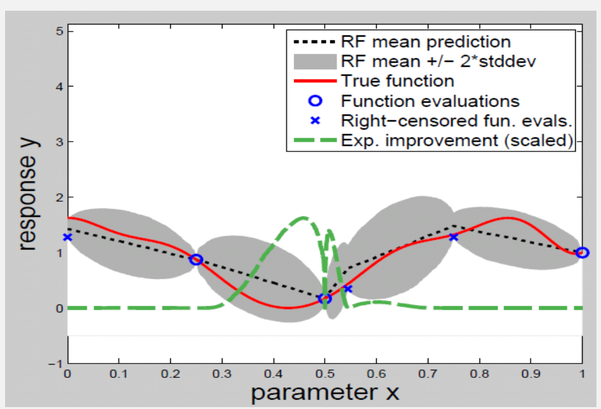
- E.g.: clinical studies (e.g. side effects too bad)

Def. A cost-varying function minimization problem (f,c) is *cost-monotonic* iff

$$\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \boldsymbol{\Theta}. \, (f(\boldsymbol{\theta}_1) < f(\boldsymbol{\theta}_2) \Leftrightarrow c(\boldsymbol{\theta}_1) < c(\boldsymbol{\theta}_2)) \, .$$

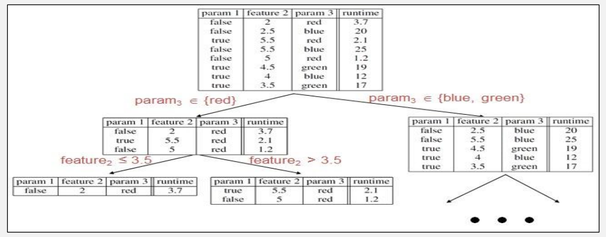
Bayesian Optimization under censoring

- Adaptively censor costly function evaluations
- Integrate censored data points in the model



Base Regression Model:

Due to our high-dimensional, predominantly discrete inputs, we use *random forests* [Breiman 2001], ensembles of regression trees like this:



Empirical mean μ_θ and variance σ_θ^2 across trees' predictions define predictive distribution $N(\mu_\theta,\sigma_\theta^2)$

Regression Models Under Censoring

Our data: $\{(\theta_i,y_i,c_i)\}_{i=1,\dots n}$, where c_i is a *censoring indicator*: $y_i = f(\theta_i)$ if $c_i=0$ and $y_i \leq f(\theta_i)$ if $c_i=1$

Truncated distribution $N(\mu_\theta,\sigma_\theta^2)_{\geq\kappa}$ is defined by pdf:

$$p(x) = \begin{cases} 0 & x < \kappa \\ \frac{1}{\sigma_{\boldsymbol{\theta}}}\varphi(\frac{x-\mu_{\boldsymbol{\theta}}}{\sigma_{\boldsymbol{\theta}}})/(1-\Phi(\frac{\mu_{\boldsymbol{\theta}}-\kappa}{\sigma_{\boldsymbol{\theta}}})) & x \geq \kappa \end{cases}$$

Direct adaptation of previous EM algorithm [Schmee & Hahn, 1979]: Fit initial random forest, then iterate:

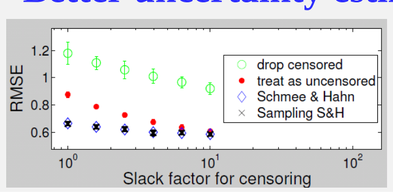
E. For each tree T and each i s.t. $c_i = 1$:
$\hat{y}_i^{(T)} \leftarrow$ mean of $\mathcal{N}(\mu_{\boldsymbol{\theta}_i}, \sigma^2_{\boldsymbol{\theta}_i})_{\geq y_i}$.

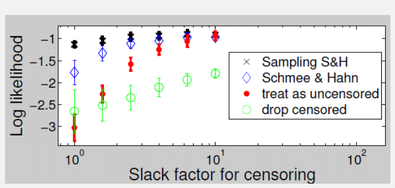
M. Re-fit the random forest using
$(\boldsymbol{\theta}_i, \hat{y}_i^{(T)}, c_i)_{i=1}^n$ as the basis for tree T.

To preserve our uncertainty, change E step to:

E. For each tree T and each i s.t. $c_i = 1$:
$\hat{y}_i^{(T)} \leftarrow$ sample from $\mathcal{N}(\mu_{\boldsymbol{\theta}_i}, \sigma^2_{\boldsymbol{\theta}_i})_{\geq y_i}$.

Better uncertainty estimates than normal EM





Algorithm configuration (AC)

Def. Given a parameterized algorithm A, a distribution D of problem instances $\pi \in \mathcal{I}$, and a performance metric $m(\boldsymbol{\theta}, \pi)$, let $f(\boldsymbol{\theta}) = \mathbb{E}_{\pi\sim D}[m(\boldsymbol{\theta}, \pi)]$. The *algorithm configuration (AC)* problem is then to find a parameter setting $\boldsymbol{\theta}$ of A that solves $\arg\min_\theta f(\boldsymbol{\theta})$.

A challenge for Bayesian optimization

- High dimensions (e.g. 76 for optimizing CPLEX)
- Mixed discrete/continuous parameters
- Tens of thousands of data points
- Very large non-Gaussian noise
- Time budget (learning & EI opt. counts as part of it!)
- Marginal optimization over heterogeneous instances
- Massive parallelism possible
- Cost-varying problem (sometimes drastic variance)

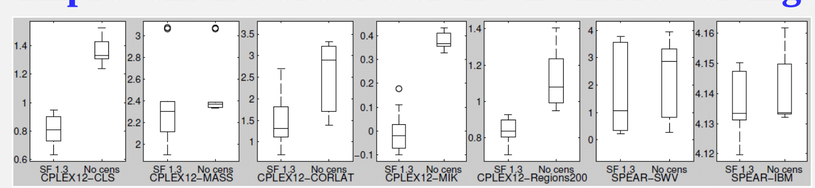
Sequential Model-based Algorithm Configuration (SMAC) [HHL-B, LION'11]

- State-of-the-art AC procedure
- Handles issues above (using random forests & heuristics)
- Room for improvement in future work:
  + Uncertainty estimates can be overconfident
  + Heuristics: number of runs per setting, which instances (& which censoring time)

Here: exploit cost monotonicity in SMAC

- Censor runs just above best observed cost: SF * $f(\theta_{inc})$

Improvements of state of the art in AC solving



Scenario	Unit	Median of mean runtimes on test set					
		SF 1	SF 1.1	SF 1.3	SF 1.5	SF 2	No censoring
CPLEX12-CLS	$[\cdot 10^0 s]$	**5.27**	**6.21**	**6.47**	8.3	6.66	21.4
CPLEX12-MASS	$[\cdot 10^2 s]$	**6.39**	**1.94**	**2.02**	**1.94**	**1.97**	**2.33**
CPLEX12-CORLAT	$[\cdot 10^0 s]$	**17.6**	**9.52**	**20.5**	**15.4**	**16.9**	826
CPLEX12-MIK	$[\cdot 10^{-1} s]$	**8.88**	**9.3**	**9.54**	**9.45**	**9.86**	23.9
CPLEX12-Regions200	$[\cdot 10^0 s]$	**6.93**	**6.65**	**6.85**	**7.21**	**8.07**	12
SPEAR-SWV	$[\cdot 10^0 s]$	**67.2**	**521**	**8.15**	**7.78**	**290**	**1030**
SPEAR-IBM	$[\cdot 10^4 s]$	**1.36**	**1.36**	**1.36**	**1.36**	**1.36**	**1.36**