Learning When to Stop Thinking and Do Something!

Barnabás Póczos, Yasin Abbasi-Yadkori, Csaba Szepesvári, Russell Greiner, Nathan Sturtevant
Department of Computing Science, University of Alberta

Framework

- $\{X_t\}_t$ is an IID sequence
- For X_t, we start a "thinking process"
- $Y_{tk} \in \mathcal{Y}_k$: information about X_t at stage k of thinking
- τ_{tk} : the time used in thinking at stage k
- Y_{tk} is independent of $Y_{t,1}, \ldots, Y_{t,k-2}$ given $Y_{t,k-1}$ and X_t
- q_k determines whether we terminate the thinking process at stage k
- At stage k, we continue thinking at X_t with probability $q_k(0|Y_{tk})$, or quit with probability $q_k(1|Y_{tk}) = 1 - q_k(0|Y_{tk})$
- $L_t \in \{1, \ldots, K\}$ denotes when we quit and $T_t = \sum_{k=1}^{L_t} \tau_k$
- $A_t = \mu_{L_t}(Y_{t,L_t})$ is the action taken on instance X_t
- The performance criterion:

$$\rho^q = \mathbb{E}\left[\liminf_{t\to\infty} \frac{\sum_{s=1}^t r(X_s, A_s)}{\sum_{s=1}^t T_s}\right]$$

Learning Stopping Policies

- Policy-gradient-based algorithms
- By the law of large numbers $\rho^q = \frac{\mathbb{E}[r(X_1,A_1)]}{\mathbb{E}[T_1]}$
- $\frac{\partial}{\partial\theta}\rho^q = \frac{\partial}{\partial\theta}\frac{\mathbb{E}[r_1]}{\mathbb{E}[T_1]} = \frac{\mathbb{E}[T_1]\Delta\mathbb{E}[r_1] - \mathbb{E}[r_1]\Delta\mathbb{E}[T_1]}{\mathbb{E}[T_1]^2}$

Direct Gradient Ascent

- $\mathbb{E}[T_1] \approx \frac{1}{n}\sum_{t=1}^n \sum_{k=1}^{L_t} \tau_k$, $\mathbb{E}[r(X_1, A_1)] \approx \frac{1}{n}\sum_{t=1}^n R_t$
- $\mathbb{E}[r(X_1, A_1)]$ is just the reward obtained in an episodic problem
- Use likelihood ratios (aka REINFORCE) to calculate the derivative
- Unbiased estimate:

$$\frac{\partial}{\partial\theta}\mathbb{E}[r(X_1, A_1)] \approx \frac{1}{n}\sum_{t=1}^n r(\widetilde{Y}_t)\left(\sum_{k=1}^{L_t-1}\frac{\partial}{\partial\theta}\ln q^\theta(0|Y_{tk}) + \frac{\partial}{\partial\theta}\ln q^\theta(1|Y_{tL_t})\right)$$

- Similar result for $\frac{\partial}{\partial\theta}\mathbb{E}[T_1]$
- Putting the pieces together...

$$\hat{G}_n = \frac{1}{n}\sum_{t=1}^n\left(\frac{r(\widetilde{Y}_t)}{\hat{T}} - \frac{\hat{r}T_t}{\hat{T}^2}\right)\left(\sum_{k=1}^{L_t-1}\frac{\partial}{\partial\theta}\ln q^\theta(0|Y_{tk}) + \frac{\partial}{\partial\theta}\ln q^\theta(1|Y_{tL_t})\right)$$

The Quality of the Estimated Gradient

Proposition 1. *Assume that $n \geq 2\log(1/\delta)/\tau_0^2$, where τ_0 is an almost sure lower bound on T_1. Then with probability $1-\delta$,*

$$\|G - \hat{G}_n\| \leq c_1\sqrt{\frac{\log(4/\delta)}{n}} + c_2\frac{\log(4/\delta)}{n} = c(\delta, n),$$

where c_1, c_2 are constants that depend only on the range of the rewards, thinking times and their gradients.

A Stopping Rule for Preventing Slow Convergence Near Optima

Theorem 1. *Fix $0 < \delta < 1$ and let $n = n(\delta)$ be the first (random) time when*

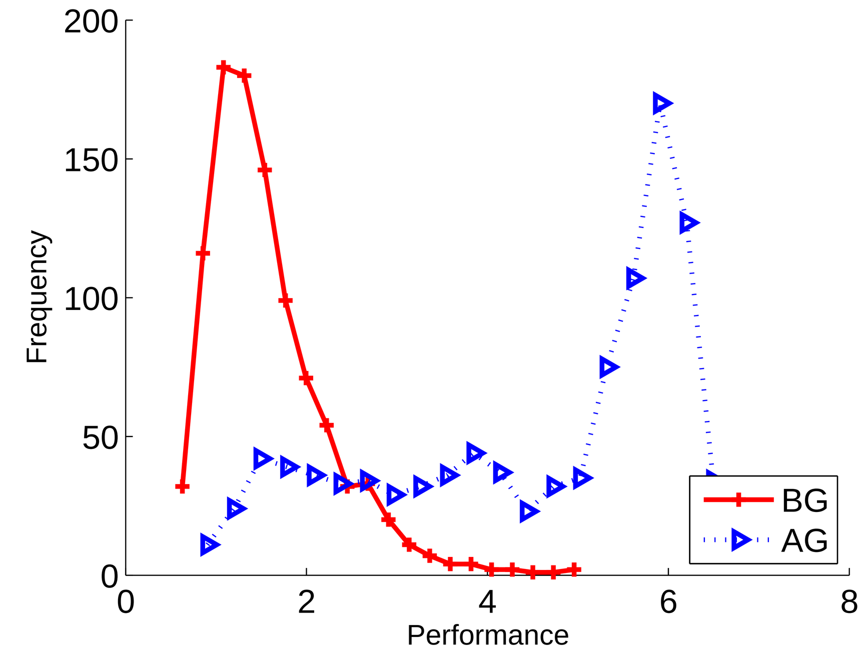
$$c(\delta, n) \leq \frac{1}{2}\max(0, \|\hat{G}_n\| - c(\delta, n)).$$

Then $\hat{G}_n^T G > 0$ with probability $\geq 1 - \delta$.

Experiments

A Toy Problem

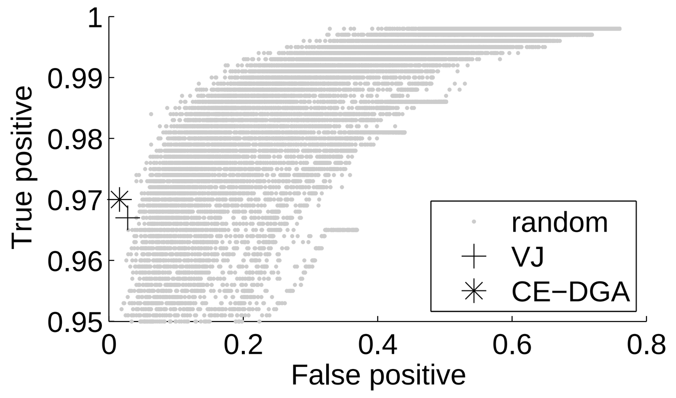
- Sort envelopes based on their zipcodes
- For envelope X_t apply subroutines $\langle A_1, \ldots, A_K\rangle$
- p_k: the probability that $y_{t,k}$ is the correct zip code ($p_0 \sim$ Beta$(1,1)$ and improves as $p_{k+1} = \min\{p_k + 0.1,\ 1\}$)
- Generate 1,000,000 random parameter vectors (policies)
- The highest, lowest and average performances: 6.85, 0.34 and 1.60
- The performance histogram of 1000 parameters before (BG) and after (AG) applying the DGA method. DGA improves the policies considerably:

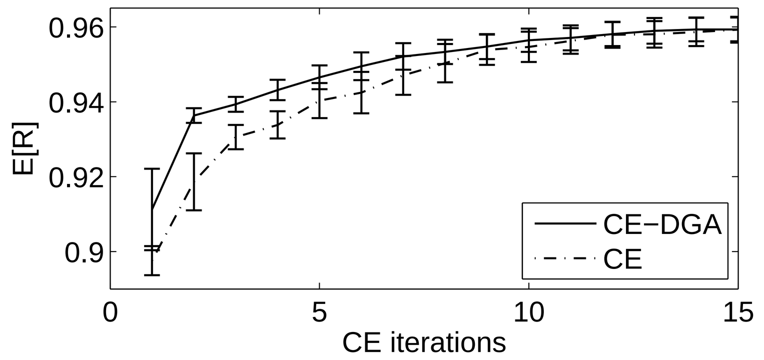


Face Detection

- Face database: 4916 pieces of facial and 7872 pieces of non-facial gray scale images of size 24×24 from the VJ database
- A 22-stage hierarchical face classifier of Lienhart et al. (2003)
- While the higher level classifiers perform better, they have higher complexity
- Containes 22 parameters $\alpha_k \in \mathbb{R}$, $k = 1, \ldots, 22$ (chosen such that $TPR = 99.5\%$)
- Not able to classify an image before reaching stage 22
- $TPR = 99.5$ seems ad-hoc
- Optimize these parameters
- Combine the gradient descent with the Cross-Entropy method (CE-DGA)
- CE-DGA achieves higher expected reward, higher TPR, and smaller FPR than the VJ parameters, while using many fewer classification stages:

	VJ	Random	CE-DGA
$\mathbb{E}[R]$	96.95	76.80	97.70
$\mathbb{E}$[stage]	13.17	1.25	6.1
TPR	96.70%	99.20%	97.00%
FPR	2.80%	45.60%	1.60%





References

R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM'03, 25th Pattern Recognition Symposium*, pages 297–304, 2003.