Stochastic Optimization of Areas Under Precision-Recall Curves with Provable Convergence

Qi Qi*† , Youzhi Luo*‡, Zhao Xu*‡ , Shuiwang Ji‡, Tianbao Yang†

†Department of Computer Science, The University of Iowa ‡Department of Computer Science & Engineering, Texas A&M University

{qi-qi,tianbao-yang}@uiowa.edu, {yzluo,zhaoxu,sji}@tamu.edu * Equal contribution.

Background and Problem Definition

Aera Under Precision-Recall Curves (AUPRC):

$$\text{AUPRC} = \int_{-\infty}^{\infty} \Pr(Y=1|F \geq c) d\Pr(F \leq c|Y=1)$$

where c is the prediction threshold. $Y \in \{0,1\}$ is the label. For a finite set of examples $\mathbf{D} = \{(\mathbf{x}_i, y_i), i = 1,\ldots,n\}$ with the prediction score for each example $\mathbf{x}_i$ given by $h_{\mathbf{w}}(\mathbf{x}_i)$, we consider to use AP to approximate AUPRC, which is given by

$$\text{AP} = \frac{1}{n_+}\sum_{i=1}^{n} \mathbf{I}(y_i=1)\frac{\sum_{s=1}^{n}\mathbf{I}(y_s=1)\mathbf{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{s=1}^{n}\mathbf{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))}$$

where n_+ denotes the number of positive examples. It can be shown that AP is an unbiased estimator in the limit $n \to \infty$ [1].

Surrogate AP Loss

$$\min_{\mathbf{w}} P(\mathbf{w}) = \frac{1}{n_+}\sum_{\mathbf{x}_i \in \mathbf{D}_+} \frac{-\sum_{s=1}^{n}\mathbf{I}(y_s=1)\ell(\mathbf{w};\mathbf{x}_s;\mathbf{x}_i)}{\sum_{s=1}^{n}\ell(\mathbf{w};\mathbf{x}_s;\mathbf{x}_i)}. \quad (1)$$

$\ell(\mathbf{w};\mathbf{x}_s,\mathbf{x}_i)$ is a surrogate function of $\mathbf{I}(h_{\mathbf{w}}(\mathbf{x}_s) \geq h_{\mathbf{w}}(\mathbf{x}_i))$, e.g., squared hinge loss, logistic loss and sigmoid loss, etc.

Related Work

1. Traditional: Hill Climb Search [1], Dynamic Programming [2]
2. Maximizing AP Score: Approximate gradient of AP or its smooth function Sensitive to batch size: SmoothAP [3], FastAP [4]

Research Question

Can we design direct stochastic optimization algorithms both in SGD-style and Adam-style for maximizing AP with provable convergence guarantee? Yes!

Problem Formulation

We cast the problem into a finite-sum of compositional functions. By denoting

$$g(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i) = [g_1(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i), g_2(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)]^\top = [\ell(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)\mathbf{I}(y_j=1), \ell(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)]^\top$$
$$g_{\mathbf{x}_i}(\mathbf{w}) = \mathbb{E}_{\mathbf{x}_j\sim\mathbf{D}}[g(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)], \quad (2)$$

where $g_{\mathbf{x}_i}(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}^2$. Let $f(\mathbf{s}) = -\frac{s_1}{s_2} : \mathbb{R}^2 \to \mathbb{R}$. Then, we can write the surrogate AP loss $P(\mathbf{w})$ as a sum of compositional functions:

$$P(\mathbf{w}) = \frac{1}{n_+}\sum_{\mathbf{x}_i \in \mathbf{D}_+} f(g_{\mathbf{x}_i}(\mathbf{w})) = \mathbb{E}_{\mathbf{x}_i\sim\mathbf{D}_+}[f(g_{\mathbf{x}_i}(\mathbf{w}))]. \quad (3)$$

Assumptions

Assumption 1: Assume that

(a) There exists Δ_1 such that $P(\mathbf{w}_1) - \min_{\mathbf{w}} P(\mathbf{w}) \leq \Delta_1$;

(b) There exist $C, M > 0$ such that $\ell(\mathbf{w};\mathbf{x}_i,\mathbf{x}_i) \geq C$ for any $\mathbf{x}_i \in \mathbf{D}_+$, $\ell(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i) \leq M$, and $\ell(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)$ is Lipscthiz continuous and smooth with respect to $\mathbf{w}$ for any $\mathbf{x}_i \in \mathbf{D}_+, \mathbf{x}_j \in \mathscr{D}$;

(c) There exists $V > 0$ such that $\mathbb{E}_{\mathbf{x}_j\sim\mathbf{D}}[\|g(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i) - g_{\mathbf{x}_i}(\mathbf{w})\|^2] \leq V$, and $\mathbb{E}_{\mathbf{x}_j\sim\mathbf{D}}[\|\nabla g(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i) - \nabla g_{\mathbf{x}_i}(\mathbf{w})\|^2] \leq V$ for any $\mathbf{x}_i$.

With a bounded score function $h_{\mathbf{w}}(\mathbf{x})$ the above assumption can be easily satisfied. Assumption 1 also implies $P(\mathbf{w})$ is smooth.

Proposed Stochastic Algorithms: SOAP

Algorithm 1 SOAP

Input: γ, α, u_0, and other parameters for SGD-stype update or Adam-stype update.
Initialize $\mathbf{w}_1 \in \mathbb{R}^d$, $\mathbf{u} \in \mathbb{R}^{|n_+|\times 2}$
for $t = 1,\ldots,T$ **do**
  Draw a batch of B_+ positive samples denoted by $\mathscr{B}_+$.
  Draw a batch of B samples denoted by $\mathscr{B}$.
  $\mathbf{u} = \text{UG}(\mathscr{B}, \mathscr{B}_+, \mathbf{u}, \mathbf{w}_t, \gamma, u_0)$
  Compute (biased) Stochastic Gradient Estimator

$$G(\mathbf{w}_t) = \frac{1}{B_+}\sum_{\mathbf{x}_i\in\mathscr{B}_+}\sum_{\mathbf{x}_j\in\mathscr{B}} \frac{(\mathbf{u}^1_{\mathbf{x}_i} - \mathbf{u}^2_{\mathbf{x}_i}\mathbf{I}(j=1))\nabla\ell(\mathbf{w};\mathbf{x}_j,\mathbf{x}_i)}{B(\mathbf{u}^2_{\mathbf{x}_i})^2} \quad (4)$$

  Update $\mathbf{w}_{t+1}$ by a SGD-style method or by a Adam-style method

$$\mathbf{w}_{t+1} = \text{UW}(\mathbf{w}_t, G(\mathbf{w}_t))$$

end for

Algorithm 2 $\text{UG}(\mathscr{B}, \mathscr{B}_+, \mathbf{u}, \mathbf{w}_t, \gamma, u_0)$

for each positive $\mathbf{x}_i \in \mathscr{B}_+$ **do**
  Compute

$$[\tilde{g}_{\mathbf{x}_i}(\mathbf{w}_t)]_1 = \frac{1}{|\mathscr{B}|}\sum_{\substack{x_j\in\mathscr{B}\\ y_j=1}} \ell(\mathbf{w}_t;\mathbf{x}_j,\mathbf{x}_i)$$

$$[\tilde{g}_{\mathbf{x}_i}(\mathbf{w}_t)]_2 = \frac{1}{|\mathscr{B}|}\sum_{\mathbf{x}_j\in\mathscr{B}} \ell(\mathbf{w}_t;\mathbf{x}_j,\mathbf{x}_i)$$

  Compute $\mathbf{u}^1_{\mathbf{x}_i} = (1-\gamma)\mathbf{u}^1_{\mathbf{x}_i} + \gamma[\tilde{g}_{\mathbf{x}_i}(\mathbf{w}_t)]_1$
  $\mathbf{u}^2_{\mathbf{x}_i} = \max((1-\gamma)\mathbf{u}^2_{\mathbf{x}_i} + \gamma[\tilde{g}_{\mathbf{x}_i}(\mathbf{w}_t)]_2, u_0)$
end for
Return u

Algorithm 3 $\text{UW}(\mathbf{w}_t, G(\mathbf{w}_t))$

Option 1: SGD-style update (paras: α)

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha G(\mathbf{w}_t)$$

Option 2: Adam-style update (paras: $\alpha, \varepsilon, \eta_1, \eta_2$)

$$h_{t+1} = \eta_1 h_t + (1-\eta_1)G(\mathbf{w}_t)$$
$$v_{t+1} = \eta_2 \hat{v}_t + (1-\eta_2)(G(\mathbf{w}_t))^2$$
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\frac{h_{t+1}}{\sqrt{\varepsilon + \hat{v}_{t+1}}}$$

where $\hat{v}_t = v_t$ (Adam) or $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ (AMSGrad)
Return: $\mathbf{w}_{t+1}$

SGD-Style Theorem (Algorithm 1 + 2)

Suppose Assumption 1 holds, let the parameters be $\alpha = \frac{1}{n_+^{2/5}T^{3/5}}, \gamma = \frac{n_+^{2/5}}{T^{2/5}}$, $\forall\, t \in 1,\cdots,T$, and $T > n_+$. Then after running T iterations, SOAP with a SGD-style update satisfies $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla P(\mathbf{w}_t)\|^2\right] \leq O(\frac{n_+^{2/5}}{T^{2/5}})$, where O suppresses constant numbers.

Adam-Style Theorem (Algorithm 1 + 3)

Suppose Assumption 1 holds, let the parameters $\eta_1 \leq \sqrt{\eta_2} \leq 1$, $\alpha = \frac{1}{n_+^{2/5}T^{3/5}}, \gamma = \frac{n_+^{2/5}}{T^{2/5}}$, $\forall\, t \in 1,\cdots,T$, and $T > n_+$. Then after running T iterations, SOAP with an AMSGRAD update satisfies $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla P(\mathbf{w}_t)\|^2\right] \leq O(\frac{n_+^{2/5}}{T^{2/5}})$, where O suppresses constant numbers.

Literature

[1]Metzler, D., Croft, W. B. (2005, August). A markov random field model for term dependencies. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.

[2] Song, Y., Schwing, A., Urtasun, R. (2016, June). Training deep neural networks via direct loss minimization. In International Conference on Machine Learning.

[3] Brown, A., Xie, W., Kalogeiton, V., Zisserman, A. (2020, August). Smooth-ap: Smoothing the path towards large-scale image retrieval. In European Conference on Computer Vision.

[4] Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S. (2019). Deep metric learning to rank. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Ablation Study

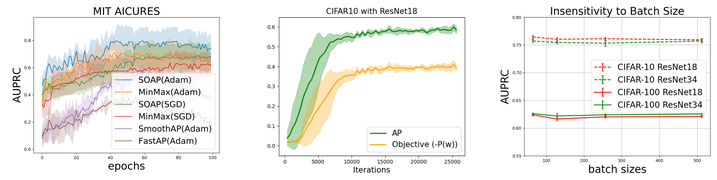

Figure 1:Left most: Comparison of convergence of different methods in terms of test AUPRC scores on MIT AICURES. Middle: Consistency between AP andSurrogate Objective -P(w)vs Iterations on CIFAR10. Right most: Insensitivity to batch size of SOAP.

Table: The test AUPRC over 3 independent runs by SOAP with different surrogate functions.

Data	CIFAR10		CIFAR100	
Networks	ResNet18	ResNet34	ResNet18	ResNet34
Squared Hinge	0.7629 (±0.0014)	0.7012 (±0.0056)	0.6251 (±0.0053)	0.6001 (±0.0060)
Logistic	0.7542 (±0.0024)	0.6968 (±0.0121)	0.6378 (±0.0031)	0.5923 (±0.0101)
Sigmoid	0.7652 (±0.0035)	0.6983 (±0.0084)	0.6271 (±0.0043)	0.5832 (±0.0054)

Comparison with State-Of-The-Art (SOTA) methods

Soap code for reproducing results. `https://github.com/Optimization-AI`

- Cross entropy loss (CE)
- Three surrogate AP: MinMax, SmoothAP, FastAP
- SOTA imbalanced DL methods: LDAM, CB-CE, Focal, and AUC-M

Tasks: Imbalanced binary image classification.

Datasets: Binary Imbalanced CIFAR10, CIFAR100 (Manually Constructed), Melanoma **Evaluation Metric**: AUPRC **Model**: ResNet18, ResNet34 **Optimzer**: SGD-Style SOAP

Table: The test AUPRC on the image datasets with two ResNet models. We report the average AUPRC and standard deviation (within brackets) over 5 runs.

Datasets	CIFAR-10		CIFAR-100	
Networks	ResNet18	ResNet34	ResNet18	ResNet34
CE	0.7155 (± 0.0058)	0.6844(± 0.0031)	0.5946 (± 0.0031)	0.5792 (± 0.0028)
CB-CE	0.7325 (± 0.0039)	0.6936(±0.0021)	0.6165 (± 0.0096)	0.5632(± 0.0129)
Focal	0.7183(± 0.0082)	0.6943(± 0.0007)	0.6107(± 0.0093)	0.5585(± 0.0285)
LDAM	0.7346 (± 0.0125)	0.6745(± 0.0043)	0.6153 (± 0.0100)	0.5662(± 0.0212)
AUC-M	0.7399(± 0.0013)	0.6825(± 0.0089)	0.6103 (± 0.0075)	0.5306(± 0.0230)
SmoothAP	0.7365 (± 0.0088)	0.6909 (± 0.0049)	0.6071(± 0.0143)	0.5208 (± 0.0505)
FastAP	0.7028 (± 0.0341)	0.6798 (± 0.0032)	0.5618(± 0.0351)	0.5151(± 0.0450)
MinMax	0.7228 (± 0.0118)	0.6806(± 0.0027)	0.6071(± 0.0064)	0.5518(± 0.0030)
SOAP	**0.7629**(± 0.0014)	**0.7012**(± 0.0056)	**0.6251** (± 0.0053)	**0.6001**(± 0.0060)

Tasks: Graph Neural Network Prediction

Datasets: HIV, MUV, MIT-AICURES **Evaluation Metric**: AUPRC
Model: GINE, MPNN, ML-MPNN **Optimzer**: Adam-Style SOAP

Table: The test AUPRC values on the HIV datasets with three graph neural network models over 3 independent runs.

Dataset	Method	GINE	MPNN	ML-MPNN
HIV	CE	0.2774 (± 0.0101)	0.3197 (± 0.0050)	0.2988 (± 0.0076)
	CB-CE	0.3082 (± 0.0101)	0.3056 (± 0.0018)	0.3291 (± 0.0189)
	Focal	0.3179 (± 0.0068)	0.3136 (± 0.0197)	0.3279 (± 0.0173)
	LDAM	0.2904 (± 0.0008)	0.2994 (± 0.0128)	0.3044 (± 0.0116)
	AUC-M	0.2998 (± 0.0010)	0.2786 (± 0.0456)	0.3305 (± 0.0165)
	SmothAP	0.2686 (± 0.0007)	0.3276 (± 0.0063)	0.3235 (± 0.0092)
	FastAP	0.0169 (± 0.0031)	0.0826 (± 0.0112)	0.0202 (± 0.0002)
	MinMax	0.2874 (± 0.0073)	0.3119 (± 0.0075)	0.3098 (± 0.0167)
	SOAP	**0.3385** (± **0.0024**)	**0.3401** (± **0.0045**)	**0.3547** (± **0.0077**)