UNIVERSITY AT ALBANY
State University of New York

Fast Algorithms for AUC Maximization

Michael Natole, Jr. [1], Yiming Ying [1], Siwei Lyu [2]

[1]Department of Mathematics and Statistics

[2]Department of Computer Science

JMM, January 2019

Classification

- Given data $\{z_i = (x_i, y_i) \in \mathcal{Z} : i = 1 \ldots T\}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$, we wish to learn the following function

$$f(x_i) = \text{sign}(\mathbf{w}^T x_i) \tag{1}$$

  where $\mathbf{w} \in \mathbb{R}^d$ is the parameter to be learned.
- Evaluation by 0-1 loss is usually replaced by a convex surrogate loss $\phi : \mathbb{R} \to \mathbb{R}^+$ satisfying $\mathbb{I}_{[s<0]} \leq \phi(s)$.
  - Least Square Loss: $\phi(s) = (1 - s)^2$
  - Hinge Loss: $\phi(s) = (1 - s)_+$
- Empirical Risk Minimization (ERM)

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{T} \sum_{i=1}^{T} \phi(y_i \mathbf{w}^T x_i). \tag{2}$$

Stochastic Gradient Descent

Stochastic Gradient Descent

Initialize $\mathbf{w}_1$, and for any $t \geq 1$, draw sample $z_t = (x_t, y_t)$ at random, and then

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} \phi(y_t \mathbf{w}^T x_t) \tag{3}$$

- The idea of SGD dates back to [Robbins and Monro, 1951].
- The literature on SGD is extensive [Bottou and Cun, 2004, Moulines and Bach, 2011, Srebro and Tewari, 2010].
- Most of the literature focuses on the misclassification error or accuracy.

Accuracy

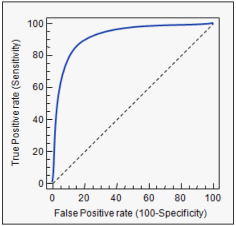
- Consider the case for a sample of 1000 instances with 990 "true" negative instances and 10 "true" positive instances. Suppose we obtain the following results:

	True +1	True -1
Predicted +1	1	11
Predicted -1	9	979

- The misclassification error (or classification accuracy) could be misleading for real world applications.
- This classifier has 98% accuracy, but told us very little.
- For this reason, we consider the use of AUC.

Probabilistic Definition of AUC

- A ROC curve is a plot of the false positive rate vs. the true positive rate.
- AUC (*area under the ROC curve*) is a widely used measure for imbalanced classification.



Definition

For a linear scoring function $f(x) = \mathbf{w}^T x$, AUC is

$$AUC(\mathbf{w}) = \Pr(\mathbf{w}^T x \geq \mathbf{w}^T x' | y = 1, y' = -1)$$
$$= 1 - \mathbb{E}[\mathbb{I}_{[\mathbf{w}^T (x - x') < 0]} | y = 1, y' = -1]$$

where (x, y), $(x', y') \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ are independent.

[Hanley and McNeil, 1982, Bradley, 1997, Fawcett, 2006]

AUC Maximization

- AUC maximization can be easily modified to a minimization problem:

$$\min_{\mathbf{w}} \mathbb{E}[\mathbb{I}_{[\mathbf{w}^T(x-x')<0]}|y=1, y'=-1] + \Omega(\mathbf{w})$$

  where $\Omega(\cdot)$ is a penalty function.
- Replacing the indicator function by the least square loss, AUC optimization can be formulated as:

$$\min_{\mathbf{w}} \mathbb{E}[(1-\mathbf{w}^T(x-x'))^2|y=1, y'=-1] + \Omega(\mathbf{w}) \quad (4)$$

- Key Challenges
  - What happens if the dataset is very large?
  - How to handle streaming data?

Summary of Existing Work

- Common approach is SGD based on local empirical error:

$$\mathcal{L}_t(\mathbf{w}) = \frac{1}{|\{j : y_j \neq y_t\}|} \sum_{j=1}^{t-1} \phi(y_t \mathbf{w}^T (x_t - x_j)) \mathbb{I}_{[y_j \neq y_t]} + \lambda \|\mathbf{w}\|^2$$

Algorithm	Loss	Penalty	Storage	Iteration	Rate
OAM	General	L^2	$\mathcal{O}(td)$	$\mathcal{O}(td)$	$\mathcal{O}(1/\sqrt{T})$
OPAUC	Least-Square	L^2	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$	$\mathcal{O}(1/\sqrt{T})$
SOLAM	Least-Square	L^2	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(1/\sqrt{T})$
New Alg.	Least-Square	General	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(1/T)$

[Zhao et al. (2012); Kar et al. (2014); Gao et al (2013); Ying et al. (2016)]

Previous Work

Theorem

AUC optimization (4) in the linear case is equivalent to the following saddle point problem:

$$\min_{\mathbf{w},a,b} \max_{\alpha\in\mathbb{R}} \left\{\mathbb{E}[F(\mathbf{w},a,b,\alpha;z)] + \Omega(\mathbf{w})\right\}, \quad (5)$$

where the expectation is with respect to $z = (x, y)$, and $F(\mathbf{w}, a, b, \alpha; z)$ is a quadratic function involving $p = Pr(y = 1)$.

- To solve this problem, upon receiving data z_t we can perform gradient descent on the primal variables $\mathbf{v} = (\mathbf{w}, a, b)$ and gradient ascent on the dual variable α:

$$\mathbf{v}_{t+1} = \mathbf{v}_t - \gamma_t \partial_{\mathbf{v}} F(\mathbf{v}_t, \alpha_t, z_t), \ \alpha_{t+1} = \alpha_t + \gamma_t \partial_\alpha F(\mathbf{v}_t, \alpha_t, z_t)$$

[Ying et al., 2016, Nemirovski et al., 2009]

Stochastic Proximal AUC Maximization

- **Key Observation**: For fixed $\mathbf{w}$, it is easy to see that the optima for a, b, and α are respectively achieved at

$$a(\mathbf{w}) = \mathbf{w}^\top \mathbb{E}[x|y=1], \quad b(\mathbf{w}) = \mathbf{w}^\top \mathbb{E}[x|y=-1], \tag{6}$$

$$\alpha(\mathbf{w}) = \mathbf{w}^\top (\mathbb{E}[x|y'=-1] - \mathbb{E}[x|y=1]). \tag{7}$$

SPAM [Natole et al., 2018]

Initialize $\mathbf{w}_1 \in \mathbb{R}^d$.
Receive sample $z_t = (x_t, y_t)$
Compute $a(\mathbf{w}_t)$, $b(\mathbf{w}_t)$, and $\alpha(\mathbf{w}_t)$ according to (6) and (7).
$\mathbf{w}_{t+1} = \mathrm{prox}_{\eta_t \Omega}(\mathbf{w}_t - \eta_t \partial_1 F(\mathbf{w}_t, a(w_t), b(\mathbf{w}_t), \alpha(\mathbf{w}_t); z_t))$

- SPAM follows "proximal splitting" [Duchi and Singer, 2009, Rosasco et al., 2014] to hand non-smooth penalty term using the proximal step is given by $\mathrm{prox}_{\eta_t \Omega}(u) = \arg\min \left\{ \frac{1}{2} \| u - \mathbf{w} \|_2^2 + \eta_t \Omega(\mathbf{w}) \right\}$

Convergence Analysis: Assumptions

- (A1) Assume data $\{z_t = (x_t, y_t)\}$ is i.i.d.
- (A2) Assume that $\Omega(\cdot)$ is β-strongly convex.
- (A3) There exists an $M > 0$ such that $\|\mathbf{x}\| \leq M$ for any $x \in \mathcal{X}$.

Theorem

Under the assumptions of (A1), (A2), and (A3), and choosing step sizes $\{\eta_t = [\widetilde{C}_{\beta,M}(t+1)]^{-1} : t \in \mathbb{N}\}$, the algorithm SPAM achieves the following:

$$\mathbb{E}[\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2] = \mathcal{O}\left(\frac{\log T}{T}\right)$$

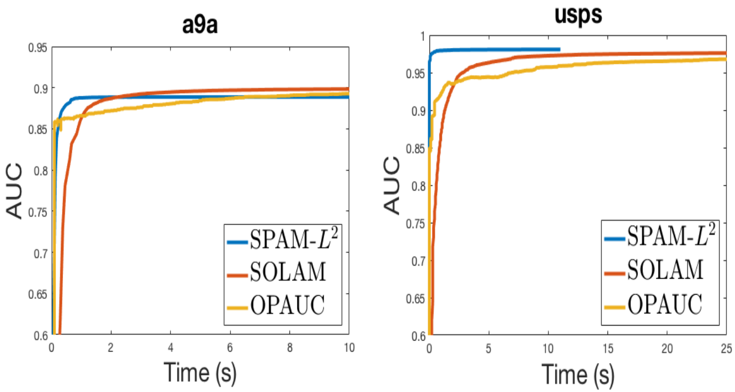
- The rate $\mathcal{O}(1/T)$ matches the optimal rate of SGD for accuracy.

Evaluation on Test Data

Data	SPAM-L^2	SPAM-NET	SOLAM	OPAUC	OAM_{seq}	OAM_{gra}	B-LS-SVM
diabetes	.8272±.0277	.8085±.0431	.8128±.0304	.8309±.0350	.8264±.0367	.8262±.0338	.8325±.0329
fourclass	.8210±.0203	.8211±.0205	.8213±.0209	.8310±.0251	.8306±.0247	.8295±.0251	.8309±.0309
german	.7942±.0388	.7937±.0386	.7778±.0373	.7978±.0347	.7747±.0411	.7723±.0358	.7994±.0343
splice	.9263±.0091	.9267±.0090	.9246±.0087	.9232±.0099	.8594±.0194	.8864±.0166	.9245±.0092
usps	.9868±.0032	.9855±.0029	.9822±.0036	.9620±.0040	.9310±.0159	.9348±.0122	.9634±.0045
a9a	.8998±.0046	.8980±.0047	.8966±.0043	.9002±.0047	.8420±.0174	.8571±.0173	.8982±.0028
mnist	.9254±.0025	.9132±.0026	.9118±.0029	.9242±.0021	.8615±.0087	.8643±.0112	.9336±.0025
acoustic	.8120±.0030	.8109±.0028	.8099±.0036	.8192±.0032	.7113±.0590	.7711±.0217	.8210±.0033
ijcnn1	.9174±.0024	.9155±.0024	.9129±.0030	.9269±.0021	.9209±.0079	.9100±.0092	.9320±.0037
covtype	.9504±.0011	.9508±.0011	.9503±.0012	.8244±.0014	.7361±.0317	.7403±.0289	.8222±.0014
sector	.8768±.0126	.9077±.0104	.8767±.0129	.9292±.0081	.9163±.0087	.9043±.0100	-
news20	.8708±.0069	.8704± .0070	.8712±.0073	.8871±.0083	.8543±.0099	.8346±.0094	-

- SPAM-L^2 uses $\Omega(\mathbf{w}) = \frac{\beta}{2}\|\mathbf{w}\|_2^2$ and SPAM-NET uses $\Omega(\mathbf{w}) = \frac{\beta}{2}\|\mathbf{w}\|_2^2 + \beta_1\|\mathbf{w}\|_1$
- Online Learning: OPAUC [Gao et al., 2013] ; OAMseq and OAMgra [Zhao et al., 2011]
- Batch Learning: B-SVM-OR and B-LS-SVM [Joachims, 2006]

Running Time Comparison



References I

Bottou, L. and Cun, Y. L. (2004).
Large scale online learning.
In *Advances in neural information processing systems*.

Bradley, A. P. (1997).
The use of the area under the roc curve in the evaluation of machine learning algorithms.
Pattern recognition, 30(7):1145–1159.

Duchi, J. and Singer, Y. (2009).
Efficient online and batch learning using forward backward splitting.
Journal of Machine Learning Research, 10(Dec):2899–2934.

Fawcett, T. (2006).
An introduction to roc analysis.
Pattern recognition letters, 27(8):861–874.

Gao, W., Jin, R., Zhu, S., and Zhou, Z.-H. (2013).
One-pass auc optimization.
In *International Conference on Machine Learning*, pages 906–914.

Hanley, J. A. and McNeil, B. J. (1982).
The meaning and use of the area under a receiver operating characteristic (roc) curve.
Radiology, 143(1):29–36.

References II

Joachims, T. (2006).
Training linear svms in linear time.
In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226, New York, NY, USA. ACM.

Moulines, E. and Bach, F. R. (2011).
Non-asymptotic analysis of stochastic approximation algorithms for machine learning.
In *Advances in Neural Information Processing Systems*, pages 451–459.

Natole, Jr., M., Ying, Y., and Lyu, S. (2018).
Stochastic proximal algorithms for AUC maximization.
In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3710–3719, Stockholmsmässan, Stockholm Sweden. PMLR.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009).
Robust stochastic approximation approach to stochastic programming.
SIAM Journal on optimization, 19(4):1574–1609.

Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
The annals of mathematical statistics, pages 400–407.

References III

Rosasco, L., Villa, S., and Vũ, B. C. (2014).
Convergence of stochastic proximal gradient algorithm.
arXiv preprint arXiv:1403.5074.

Srebro, N. and Tewari, A. (2010).
Stochastic optimization for machine learning.
ICML Tutorial.

Ying, Y., Wen, L., and Lyu, S. (2016).
Stochastic online auc maximization.
In *Advances in Neural Information Processing Systems*.

Zhao, P., Jin, R., Yang, T., and Hoi, S. C. (2011).
Online auc maximization.
In *Proceedings of the 28th international conference on machine learning (ICML-11)*.