Finite-Sum Coupled Compositional Stochastic Optimization

Theory and Applications

Bokun Wang and Tianbao Yang

ICML
International Conference
On Machine Learning

THE UNIVERSITY OF IOWA
ORGANIZED FEB. 25TH 1847.

Empirical Risk Minimization (ERM)

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i).$$

Hypothesis

Sample size

Loss

Feature

Label

$$\mathbf{z} = (\mathbf{x}, y)$$

$$\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$$

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$$

Finite-Sum Optimization

$$\min_{h \in \mathcal{H}} \hat{R}(h), \quad \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i).$$

Hypothesis parameterized by **w**

$$\min_{\mathbf{w} \in \Omega} F(\mathbf{w}), \quad F(\mathbf{w}) := \frac{1}{n} \sum_{\mathbf{z}_i \in \mathcal{D}} \ell(\mathbf{w}; \mathbf{z}_i)$$

Stochastic Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \hat{\nabla} F(\mathbf{w})$$

Unbiased estimator, e.g., $\nabla \ell(\mathbf{w}; \mathbf{z}_i)$

$$\mathbb{E}[\hat{\nabla} F(\mathbf{w})] = \nabla F(\mathbf{w})$$

Finite-Sum Optimization

$$\min_{h \in \mathcal{H}} \hat{R}(h), \quad \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i).$$

Hypothesis parameterized by **w**

$$\min_{\mathbf{w} \in \Omega} F(\mathbf{w}), \quad F(\mathbf{w}) := \frac{1}{n} \sum_{\mathbf{z}_i \in \mathcal{D}} \ell(\mathbf{w}; \mathbf{z}_i)$$

Stochastic Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \hat{\nabla} F(\mathbf{w})$$

Unbiased estimator, e.g., $\nabla \ell(\mathbf{w}; \mathbf{z}_i)$

Independent of n. Looks good?

Surrogate of Average Precision (AP) Maximization

$$F(\mathbf{w}) = -\frac{1}{|\mathcal{S}_+|} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{\sum_{\mathbf{x} \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{\mathbf{x} \in \mathcal{S}} \ell(h_{\mathbf{w}}(\mathbf{x}) - h_{\mathbf{w}}(\mathbf{x}_i))}$$

Positive Data

All Data

$$\mathcal{S} = \mathcal{S}_+ \cup \mathcal{S}_-$$

Surrogate of Average Precision (AP) Maximization

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}), \quad F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} \ell(\mathbf{w};\mathbf{z}_i)$$

$$F(\mathbf{w}) = -\frac{1}{|\mathcal{S}_+|}\sum_{\mathbf{x}_i\in\mathcal{S}_+} \frac{\sum_{\mathbf{x}\in\mathcal{S}_+}\ell(h_\mathbf{w}(\mathbf{x}) - h_\mathbf{w}(\mathbf{x}_i))}{\sum_{\mathbf{x}\in\mathcal{S}}\ell(h_\mathbf{w}(\mathbf{x}) - h_\mathbf{w}(\mathbf{x}_i))}$$

$\ell(\mathbf{w};\mathbf{z}_i)$

Stochastic Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla\ell(\mathbf{w};\mathbf{z}_i)$$

Unbiased estimator is still expensive !

Robust Logistic Regression

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}), \quad F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} \ell(\mathbf{w};\mathbf{z}_i)$$

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}} \left[\log\left(1+\exp\left(-y_i\mathbb{E}_{\xi|\mathbf{x}_i}\left[\xi^T\mathbf{w}\right]\right)\right)\right]$$

Perturbed data

$$\ell(\mathbf{w};\mathbf{z}_i)$$

Stochastic Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla\ell(\mathbf{w};\mathbf{z}_i)$$

Infeasible !

Finite-Sum Coupled Composition Optimization (FCCO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

How is it related to finite-sum optimization?

Finite-Sum Coupled Composition Optimization (FCCO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$

Take into account the cost of S_i

Finite-Sum Optimization (FO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} \ell(\mathbf{w}; \mathbf{z}_i)$$

$$F(\mathbf{w}) := \frac{1}{n} \sum_{\mathbf{z}_i \in \mathcal{D}} f_i(g(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$

Finite-Sum Coupled Composition Optimization (FCCO)

- Bipartite ranking by p-norm Push

$$F(\mathbf{w}) = \frac{1}{|\mathcal{S}_-|} \sum_{\mathbf{z}_i \in \mathcal{S}_-} \left(\frac{1}{|\mathcal{S}_+|} \sum_{\mathbf{z}_j \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{z}_j) - h_{\mathbf{w}}(\mathbf{z}_i)) \right)^p$$

- Neighborhood Component Analysis

$$F(A) = - \sum_{\mathbf{x}_i \in \mathcal{D}} \frac{\sum_{\mathbf{x} \in \mathcal{C}_i} \exp(-\|A\mathbf{x}_i - A\mathbf{x}\|^2)}{\sum_{\mathbf{x} \in \mathcal{S}_i} \exp(-\|A\mathbf{x}_i - A\mathbf{x}\|^2)} \quad \mathcal{S}_i = \mathcal{D} \setminus \{\mathbf{x}_i\}, \quad \mathcal{C}_i = \{\mathbf{x}_j \in \mathcal{D} : y_j = y_i\}$$

$$F(\mathbf{w}) := \frac{1}{n} \sum_{\mathbf{z}_i \in \mathcal{D}} \ell(\mathbf{w}; \mathbf{z}_i)$$

Finite-Sum Optimization (FO)

- Logistic regression

$$F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle}\right)$$

- Ridge regression

$$F(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{x}_i^\top \mathbf{w} - y_i\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

Stochastic Alg. for FCCO problems

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

Stochastic Gradient (**Biased);**

Sample both $\mathcal{D}$ and $\mathcal{S}_i$

Stochastic Alg. for FO problems

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} \ell(\mathbf{w};\mathbf{z}_i)$$

Stochastic Gradient (Unbiased);

Sample $\mathcal{D}$

Finite-Sum Coupled Composition Optimization (FCCO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

Wait ! We have already seen something similar ...

Hu et al. "Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning." NeurIPS 2020.

Finite-Sum Coupled Composition Optimization (FCCO)

Conditional Stochastic Optimization (CSO)

Goal: better sample complexity & O(1) batch size !

Special Case: Outer problem has finite support

BSGD, BSpiderBoost: O(√T) batch size

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i \in \mathcal{D}} f_i(g(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$

$$F(\mathbf{w}) = \mathbb{E}_{\xi} f_{\xi}\left(\mathbb{E}_{\zeta|\xi}[g_{\zeta}(\mathbf{w}; \xi)]\right)$$

Wang et al. "Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions." Math. Program. 161(1-2):419–449, 2017.

Finite-Sum Coupled Composition Optimization (FCCO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

Coupled

Finite-Sum Composition Optimization (FCO)

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}),$$

$$F(\mathbf{w}) = \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathcal{S}))$$

Wang et al. "Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions." Math. Program. 161(1-2):419–449, 2017.

Finite-Sum Coupled Composition Optimization (FCCO)

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i \in \mathcal{D}} f_i(g(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$

Reformulate FCCO as FCO

Finite-Sum Composition Optimization (FCO)

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}_i(\mathbf{g}(\mathbf{w}; \mathcal{S}))$$

$$\mathbf{g}(\mathbf{w}; \mathcal{S}) = \left[g(\mathbf{w}; \mathbf{z}_1, \mathcal{S}_1)^\top, \ldots, g(\mathbf{w}; \mathbf{z}_n, \mathcal{S}_n)^\top\right]^\top$$

$$\mathcal{S} = \mathcal{S}_1 \cup \cdots \mathcal{S}_i \cdots \cup \mathcal{S}_n$$

$$\hat{f}_i(\cdot) = f_i(\mathbb{I}_i \cdot) \quad \mathbb{I}_i := [0_{d\times d}, \ldots, I_{d\times d}, \ldots, 0_{d\times d}]$$

The NASA Algorithm for FCO problem

Ghadimi et al. "A single timescale stochastic approximation method for nested stochastic optimization." SIAM J. Optim., 30:960–979,2020.

$$F(\mathbf{w}) = \frac{1}{n}\sum_{\mathbf{z}_i \in \mathcal{D}} f_i(g(\mathbf{w}; \mathcal{S}))$$

Sample mini-batches $\mathcal{B}_1 \subset \mathcal{D}, \mathcal{B}_2 \subset \mathcal{S}$

$$u \leftarrow (1-\gamma)u + \gamma g(\mathbf{w}; \mathcal{B}_2)$$

$$\mathbf{v} \leftarrow (1-\beta)\mathbf{v} + \beta \frac{1}{|\mathcal{B}_1|}\sum_{\mathbf{z}_i \in \mathcal{B}_1} \nabla g(\mathbf{w}; \mathcal{B}_2)\nabla f_i(u)$$

$$\mathbf{w} \longleftarrow \mathbf{w} - \eta \mathbf{v}$$

Apply NASA to FCCO?

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

Reformulation

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}_i(\mathbf{g}(\mathbf{w};\mathcal{S}))$$

$$\mathbf{g}(\mathbf{w};\mathcal{S}) = \left[g(\mathbf{w};\mathbf{z}_1,\mathcal{S}_1)^\top,\ldots,g(\mathbf{w};\mathbf{z}_n,\mathcal{S}_n)^\top\right]^\top$$

$$\mathcal{S} = \mathcal{S}_1\cup\cdots\mathcal{S}_i\cdots\cup\mathcal{S}_n$$

$$\hat{f}_i(\cdot) = f_i(\mathbb{I}_i\cdot) \quad \mathbb{I}_i := [0_{d\times d},\ldots,I_{d\times d},\ldots,0_{d\times d}]$$

$$\mathbf{g}(\mathbf{w};\mathcal{B}_2) = \left[g(\mathbf{w};\mathbf{z}_1,\mathcal{B}_{2,1})^\top,\ldots,g(\mathbf{w};\mathbf{z}_n,\mathcal{B}_{2,n})^\top\right]^\top$$

$$u \leftarrow (1-\gamma)u + \gamma g(\mathbf{w};\mathcal{B}_2) \qquad u\in\mathbb{R}^n$$

Each iteration: sample and update for all n coordinates !

Not efficient when n is large.

Say n = 5, B_1 = 2

Remedy: NASA + Rand. Sparsification

$$\mathbf{g}(\mathbf{w};\mathcal{B}_2) = \left[g(\mathbf{w};\mathbf{z}_1,\mathcal{B}_{2,1})^\top, g(\mathbf{w};\mathbf{z}_2,\mathcal{B}_{2,2})^\top, g(\mathbf{w};\mathbf{z}_3,\mathcal{B}_{2,3})^\top, g(\mathbf{w};\mathbf{z}_4,\mathcal{B}_{2,4})^\top, g(\mathbf{w};\mathbf{z}_5,\mathcal{B}_{2,5})^\top\right]^\top$$

$$\mathbf{g}(\mathbf{w};\mathcal{B}_2) = \left[0, g(\mathbf{w};\mathbf{z}_2,\mathcal{B}_{2,2})^\top, 0, 0, g(\mathbf{w};\mathbf{z}_5,\mathcal{B}_{2,5})^\top\right]^\top \times \frac{n}{B_1}$$

Only compute B_1 << n coordinates.

Randomly replace others with zeros

$$u \leftarrow (1-\gamma)u + \gamma \mathbf{g}(\mathbf{w};\mathcal{B}_2) \quad u \in \mathbb{R}^n$$

1) overflow?

2) per-iteration cost of rescaling (n-B_1) coordinates by (1 - γ).

3) no speed-up w.r.t. B_2.

4) need of function value bounded.

(NEW) The SOX Algorithm

$$F(\mathbf{w}) := \frac{1}{n} \sum_{\mathbf{z}_i \in \mathcal{D}} f_i(g(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$

Sample mini-batches $\mathcal{B}_1^t \subset \mathcal{D}, \mathcal{B}_{i,2}^t \subset \mathcal{S}_i$

$$u_i^t = \begin{cases} (1-\gamma)u_i^{t-1} + \gamma g(\mathbf{w}^t; \mathbf{z}_i, \mathcal{B}_{i,2}^t), & \mathbf{z}_i \in \mathcal{B}_1^t \\ u_i^{t-1}, & \mathbf{z}_i \notin \mathcal{B}_1^t \end{cases}$$

Only update and sample for a subset of coordinates !

$$\mathbf{v}^t = (1-\beta)\mathbf{v}^{t-1} + \beta \frac{1}{B_1} \sum_{\mathbf{z}_i \in \mathcal{B}_1^t} \nabla g(\mathbf{w}^t; \mathbf{z}_i, \mathcal{B}_{i,2}^t) \nabla f_i(u_i^{t-1})$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \mathbf{v}^t$$

Per-iteration computation cost: $O(B_1)$

Finite-Sum Coupled Composition Optimization (FCCO)

(NEW) The SOX Algorithm

$$F(\mathbf{w}) := \frac{1}{n}\sum_{\mathbf{z}_i\in\mathcal{D}} f_i(g(\mathbf{w};\mathbf{z}_i,\mathcal{S}_i))$$

$$u_i^t = \begin{cases}(1-\gamma)u_i^{t-1} + \gamma g\left(\mathbf{w}^t;\mathbf{z}_i,\mathcal{B}_{i,2}^t\right), & \mathbf{z}_i \in \mathcal{B}_1^t\\ u_i^{t-1}, & \mathbf{z}_i \notin \mathcal{B}_1^t\end{cases}$$

$\Updownarrow$

$$u_i^t = \begin{cases}u_i^{t-1} - \gamma\left(u_i^{t-1} - g\left(\mathbf{w}^t;\mathbf{z}_i,\mathcal{B}_{i,2}^t\right)\right), & \mathbf{z}_i \in \mathcal{B}_1^t\\ u_i^t, & \mathbf{z}_i \notin \mathcal{B}_1^t\end{cases}$$

Stochastic block coordinate descent

$$\min_{\mathbf{u}=[u_1,\ldots,u_n]^\top} \frac{1}{2}\sum_{\mathbf{z}_i\in\mathcal{D}} \left\|u_i - g\left(\mathbf{w}^t;\mathbf{z}_i,\mathcal{S}_i\right)\right\|^2$$

(NEW) The SOX Algorithm

Sample mini-batches $\mathcal{B}_1^t \subset \mathcal{D}, \mathcal{B}_{i,2}^t \subset \mathcal{S}_i$

$$u_i^t = \begin{cases} (1-\gamma)u_i^{t-1} + \gamma g\left(\mathbf{w}^t; \mathbf{z}_i, \mathcal{B}_{i,2}^t\right), & \mathbf{z}_i \in \mathcal{B}_1^t \\ u_i^{t-1}, & \mathbf{z}_i \notin \mathcal{B}_1^t \end{cases}$$

$$\mathbf{v}^t = (1-\beta)\mathbf{v}^{t-1} + \beta \frac{1}{B_1} \sum_{\mathbf{z}_i \in \mathcal{B}_1^t} \nabla g(\mathbf{w}^t; \mathbf{z}_i, \mathcal{B}_{i,2}^t) \nabla f_i(u_i^{t-1})$$

u_i^t is more intuitive ?

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \mathbf{v}^t$$

Convergence Rates

Nonconvex Convex Strongly Convex "Twice batch size, half #iterations"

Method	NC	C	SC (PL)	Outer Batch Size $\lvert\mathcal{B}_1\rvert$	Inner Batch Size $\lvert\mathcal{B}_{i,2}\rvert$	Parallel Speed-up
BSGD (Hu et al., 2020)	$O(\epsilon^{-4})$	$O(\epsilon^{-2})$	$O(\mu^{-1}\epsilon^{-1})^{\dagger}$	1	$O(\epsilon^{-2})$ (NC) $O(\epsilon^{-1})$ (C/SC)	N/A
SOAP (Qi et al., 2021)	$O(n\epsilon^{-5})$	-	-	1	1	N/A
MOAP (Wang et al., 2021)	$O\left(\frac{n\epsilon^{-4}}{B_1}\right)$	-	-	B_1	1	Partial
SOX/SOX-boost (this work)	$O\left(\frac{n\epsilon^{-4}}{B_1B_2}\right)$	$O\left(\frac{n\epsilon^{-3}}{B_1B_2}\right)$	$O\left(\frac{n\mu^{-2}\epsilon^{-1}}{B_1B_2}\right)$	B_1	B_2	Yes
SOX ($\beta=1$) (this work)	-	$O\left(\frac{n\epsilon^{-2}}{B_1}\right)^{*}$	-	B_1	B_2	Partial

Originally proposed for AP maximization

* extra assumption: monotonicity

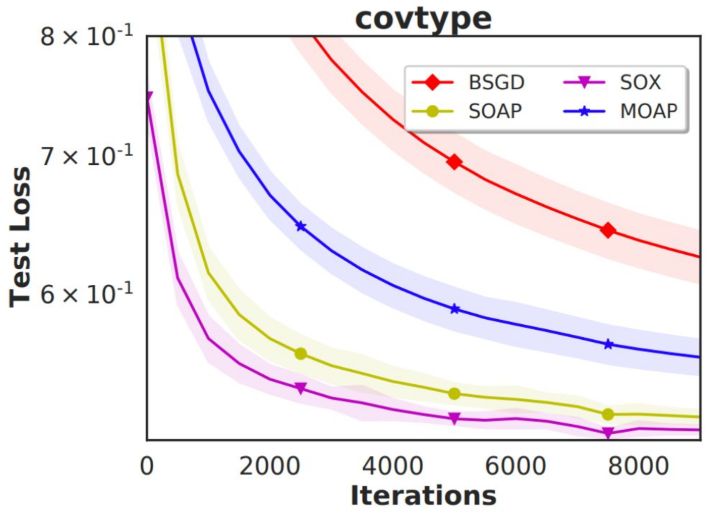
Bipartite Ranking by p-norm Push

$$F(\mathbf{w}) = \frac{1}{|\mathcal{S}_-|}\sum_{\mathbf{z}_i\in\mathcal{S}_-}\left(\frac{1}{|\mathcal{S}_+|}\sum_{\mathbf{z}_j\in\mathcal{S}_+}\ell(h_\mathbf{w}(\mathbf{z}_j)-h_\mathbf{w}(\mathbf{z}_i))\right)^p$$

A boosting-style deterministic algorithm

Algorithms	BS-PnP	SOX
Test Loss (↓)	0.778	**0.516 ± 0.003**
Time (s) (↓)	6043.90	4.62 ± 0.10

Algorithms	BS-PnP	SOX
Test Loss (↓)	0.268	**0.128 ± 0.002**
Time (s) (↓)	648.06	4.15 ± 0.06

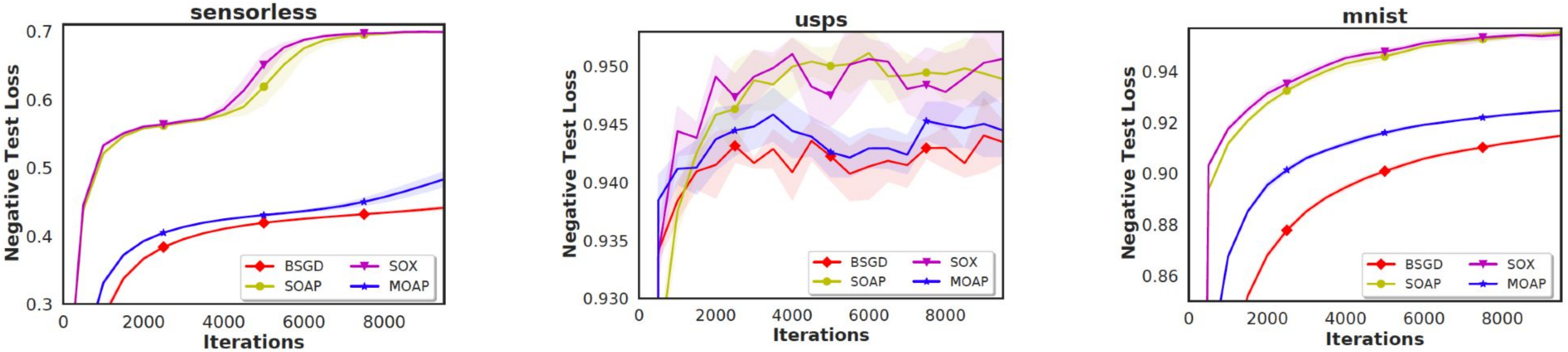


Neighborhood Component Analysis

$$F(A) = -\sum_{\mathbf{x}_i \in \mathcal{D}} \frac{\sum_{\mathbf{x} \in \mathcal{C}_i} \exp(-\|A\mathbf{x}_i - A\mathbf{x}\|^2)}{\sum_{\mathbf{x} \in \mathcal{S}_i} \exp(-\|A\mathbf{x}_i - A\mathbf{x}\|^2)}$$

$$\mathcal{C}_i = \{\mathbf{x}_j \in \mathcal{D} : y_j = y_i\}$$

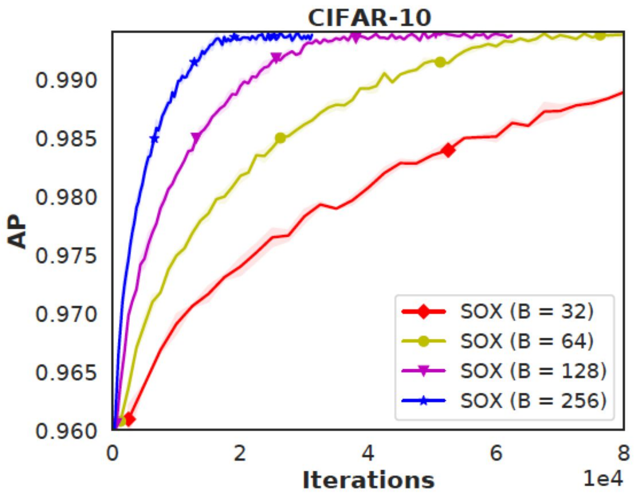
$$\mathcal{S}_i = \mathcal{D} \setminus \{\mathbf{x}_i\}$$

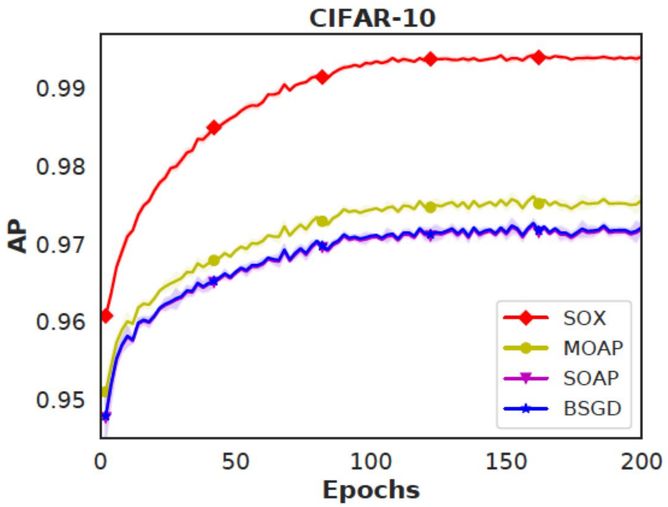


More applications of SOX: partial AUC [Zhu et. al. 2022], NDCG [Qiu et.al. 2022], contrastive learning [Yuan et.al. 2022], listwise ranking, survival analysis, etc.

AP Maximization

$$F(\mathbf{w}) = -\frac{1}{|\mathcal{S}_+|}\sum_{\mathbf{x}_i\in\mathcal{S}_+}\frac{\sum_{\mathbf{x}\in\mathcal{S}_+}\ell(h_{\mathbf{w}}(\mathbf{x})-h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{\mathbf{x}\in\mathcal{S}}\ell(h_{\mathbf{w}}(\mathbf{x})-h_{\mathbf{w}}(\mathbf{x}_i))}$$





Thank you !