Stochastic optimization library: SGDLibrary

Hiroyuki Kasai (The University of Electro-Communications, Japan)

1. Stochastic optimizations for finite-sum problems

Regularized finite-sum minimization problem:

$$\min_{w \in \mathbb{R}^d} f(w) \; := \; \frac{1}{n}\sum_{i=1}^{n} f_i(w) \; = \; \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(w, x_i, y_i)}_{\text{loss function}} + \lambda \underbrace{R(w)}_{\text{regularizer}}, \qquad w : \text{parameter}, \;\; n : \# \text{ of samples}.$$

e.g. ℓ_2-norm regularized linear regression (ridge) problem, ℓ_1-norm regularized logistic regression (LR) problem.

- Full gradient descent (a.k.a. steepest descent)
  - $w_{k+1} \leftarrow w_k - \eta \sum_{i=1}^{n} \nabla f_i(w_k)$ for all samples, and its gradient calculation cost is expensive when n is extremely large.
- Stochastic gradient descent (SGD)
  - $w_{k+1} \leftarrow w_k - \eta \nabla f_i(w_k)$ for the i-th sample uniformly at random, and its calculation cost is independent of n.
  - Assume an unbiased estimator of the full gradient as $\mathbb{E}_i[\nabla f_i(w^k)] = \nabla f(w^k)$.
  - Need diminishing step-size algorithm to guarantee convergence, which causes a severe slow convergence rate.
- Three techniques for acceleration and improvement.
  - Variance reduction (VR) techniques to exploit a full gradient estimation to reduce variance of noisy stochastic gradient.
  - Second-order (SO) algorithms to solve potential problem of first-order algorithms for ill-conditioned problems.
  - Sub-sampled Hessian algorithms to achieve second-order optimality condition.

2. Why is SGDLibrary needed?

- Need an evaluation framework to test and compare algorithms at hand for fair and comprehensive experiments, because
  - Performances of stochastic optimization algorithms are strongly influenced not only by the distribution of data but also by the step-size algorithm, and
  - Evaluators encounter results that are completely deviated from data reported in papers in every experiment.
- Need to allow researchers and implementers to easily extend or add solvers and problems for further evaluations.
- Need to accelerate researchers to devise new algorithms for further improvements.

3. What is SGDLibrary?

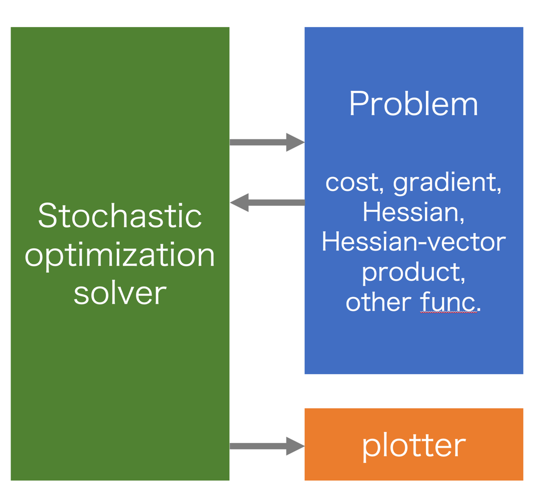
- A readable, flexible and extensible software library of a collection of stochastic optimizations and its test environment.
- Operable and executable on MATLAB as well as GNU Octave.
- Provide researchers and implementers a collection of
  - State-of-the-art stochastic optimization algorithms to solve minimization problems,
  - A variety of large-scale machine learning problems, such as linear/non-linear regression problems and classification problems, and
  - Plotting and drawing tools of performances, such as cost, optimality gap, classification accuracies, and convergence animation.

4. Software architecture and supported algorithms

Module-based architecture separating problem descriptor and optimization solver.

- Select one problem descriptor of interest and no less than one stochastic optimization solvers for use.
- Execute the selected optimization solver by calling corresponding functions via the problem descriptor such as cost calculation function (i) and stochastic derivative calculation function (ii).

Software architecture



Supported class functions of problem descriptor

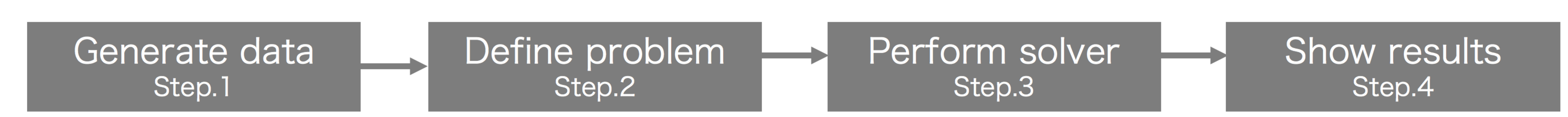
No.	Class functions (methods).	Mandatory
(i)	calculate (full) cost function $f(w)$.	✓
(ii)	calculate mini-batch stochastic derivative $v = 1/\lvert\mathcal{S}\rvert \nabla f_{i \in \mathcal{S}}(w)$ for samples set $\mathcal{S}$.	✓
(iii)	calculate stochastic Hessian.	✓
(iv)	calculate stochastic Hessian-vector product for v.	✓
(v)	problem-specific functions. (e.g., classification accuracy calculation in LR problem.)	

Supported stochastic optimization algorithms

Category	Algorithms
SGD method (inc. Adaptive learning rate)	Vanilla SGD, SGD-CM (classical momentum), SGD-CM-NAG (Nesterov's accelerated gradient), AdaGrad, RMSProp, AdaDelta , Adam, AdaMax
Variance reduction (VR) methods	SVRG, SAG, SAGA, SARAH, SARAH-Plus
Second-order (SO) methods	SQN, oBFGS-Inf, oLBFGS-Lim, Reg-oBFGS-Inf, Damp-oBFGS-Inf, IQN
SO with VR methods	SVRG-SQN, SVRG-LBFGS, SS-SVRG
Sub-sampled Hessian methods	SCR (Sub-sampled cubic regularization), Sub-sampled TR (trust region)
Other methods	BB-SGD, SVRG-BB

5. Tour of SGDLibrary: softmax classification problem

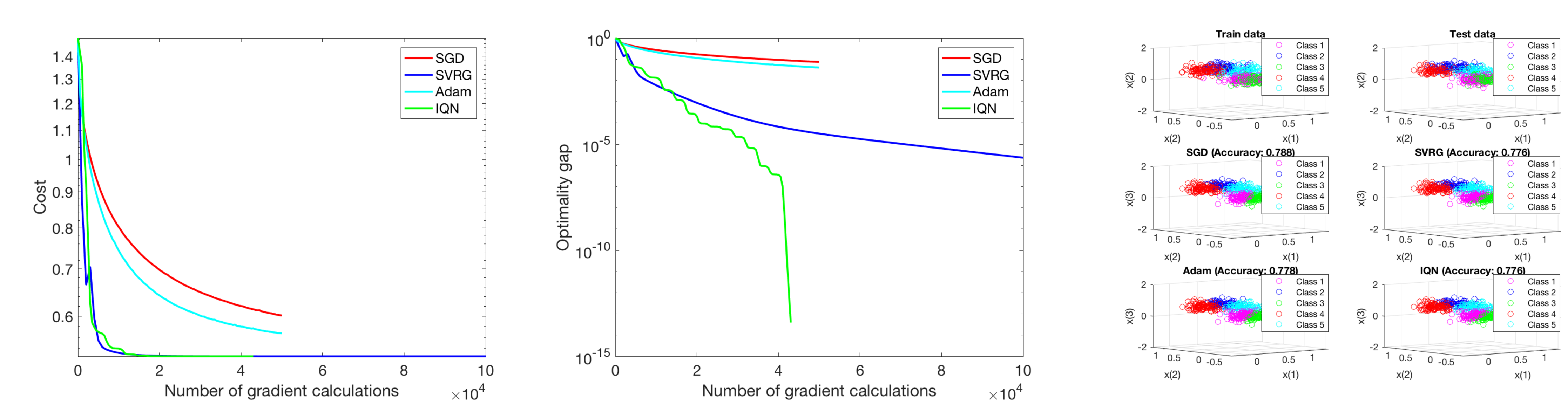
- **Only 4 steps** for simple use !



- Example: ℓ_2-norm regularized softmax classification problem.

```
% generate 100 samples of dimension 3, class 5 and std 0.15 for softmax regression
data = multiclass_data_generator(100, 3, 5, 0.15);
% define softmax regression problem
problem = softmax_regression(data.x_train, data.y_train, data.x_test, data.y_test, 5, 0.001);
% execute solvers
options.w_init = data.w_init;                    % set initial value
options.step_init = 0.0001;                      % set initial stepsize
options.verbose = 1;                             % set verbose mode
[w_sgd, info_sgd] = sgd(problem, options);       % perform SGD solver
[w_svrg, info_svrg] = svrg(problem, options);    % perform SVRG solver
% display cost vs. number of gradient evaluations
display_graph('grad_calc_count', 'cost', {'SGD','SVRG'}, {w_sgd,w_svrg}, {info_sgd,info_svrg});
```

Demonstration sample code.



(a) Cost function value (b) Optimality gap (c) Classification result

Results of cost, optimality gap, and classification for SGD, SVRG, Adam, and IQN.

MATLAB/Octave source code

`github.com/hiroyuki-kasai/SGDLibrary`.

Full paper

JMLR, vol.18, no.215, 2018 (arXiv:1710.10951).