EE5327 Optimization

Paper Implementation(theory)

Sachin Goyal (EE18MTECH11015),
Subhra Shankha Bhattacherjee (EE19MTECH01008)

March 7, 2019

Paper Title:

Hyperparameter Optimization with approximate gradient

Authors: Fabian Pedregosa

Conference: ICML, 2016

Model parameter vs Hyper-parameter

Model Parameter:is a parameter that is internal to the model and their value is estimated from the data.Their values define the skill of the model to predict output. Eg: Weights and biases.

Hyperparameters: is a parameter that is external to the model and whose value is set before the learning process begins.Their value determine how fast the model converges or performance of the model on unseen data. Eg: Learning rate, regularization parameter etc.

Optimization Problem

Author has considered to optimize $l2$-regularization hyperparameter in logistic regression model.

This is a **bi-level optimization** problem

$$\min_{\lambda \in D} \{f(\lambda) = g(X(\lambda), \lambda)\}$$

s.t.

$$X(\lambda) \in \min_{x \in R^p} h(x, \lambda)$$

where,
λ is $l2$-regularization parameter
$h(x, \lambda)$ is logistic cost function
$f(\lambda)$ is validation loss
x is model parameter

Convexity proof of *l*2-regularization

Author has used lasso method which add the *l*2-norm penalty of model parameters (weights) to the cost function.

Norms satisfy the following two properties:

$$\begin{aligned} \|kx\| &= |\ \|k\|\ |\ \|x\| \quad \text{(homogeneity)} \\ \|x+y\| &\leq \|x\| + \|y\| \quad \text{(triangle inequality)} \end{aligned}$$

So,

$$\begin{aligned} &\|\theta x + (1-\theta y)\| \leq \|\theta x\| + \|(1-\theta y\| \\ &\leq \theta\|x\| + (1-\theta)\|y\| \qquad \forall\theta \in \{0,1\} \end{aligned}$$

Hence *l*2-norm is convex.

Convexity proof of logistic loss function

Output of the logistic regression:

$$\hat{y} = \sigma(W^T X + b)$$

Where

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Loss is given by:

$$L(\hat{y}, y) = -(y log \hat{y} + (1-y) log(1-\hat{y}))$$

Using 2^{nd} derivative test,

$$\frac{\partial^2 L}{\partial \hat{y}^2} = \frac{y}{\hat{y}^2} + \frac{(1-y)}{(1-\hat{y}^2} >= 0$$

Exact gradients

$X(\lambda)$ is characterized by the implicit equation:

$$\bigtriangledown_1 h(X(\lambda), \lambda) = 0$$

Deriving the implicit equation w.r.t λ gives:

$$\bigtriangledown^2_{1,2} h + \bigtriangledown^2_1 (DX) = 0$$

Exact Gradient of f is given by:

$$\begin{aligned} \bigtriangledown f &= \bigtriangledown_2 g + (DX)^T \bigtriangledown_1 g \\ &= \bigtriangledown_2 g - (\bigtriangledown^2_{1,2} h)^T (\bigtriangledown^2_1 h)^{-1} \bigtriangledown_1 g \end{aligned}$$

Algorithm

Algorithm (HOAG). *At iteration $k = 1, 2, \ldots$ perform the following:*

(i) Solve the inner optimization problem up to tolerance ε_k. That is, find x_k such that

$$\left\| X(\lambda_k) - x_k \right\| \leq \varepsilon_k \quad .$$

(ii) Solve the linear system $\nabla_1^2 h(x_k, \lambda_k) q_k = \nabla_1 g(x_k, \lambda_k)$ for q_k up to tolerance ε_k. That is, find q_k such that

$$\left\| \nabla_1^2 h(x_k, \lambda_k) q_k - \nabla_1 g(x_k, \lambda_k) \right\| \leq \varepsilon_k \quad .$$

(iii) Compute approximate gradient p_k as

$$p_k = \nabla_2 g(x_k, \lambda_k) - \nabla_{1,2}^2 h(x_k, \lambda_k)^T q_k \quad ,$$

(iv) Update hyperparameters:

$$\lambda_{k+1} = \left(\lambda_k - \frac{1}{L} p_k \right) \quad .$$

Results

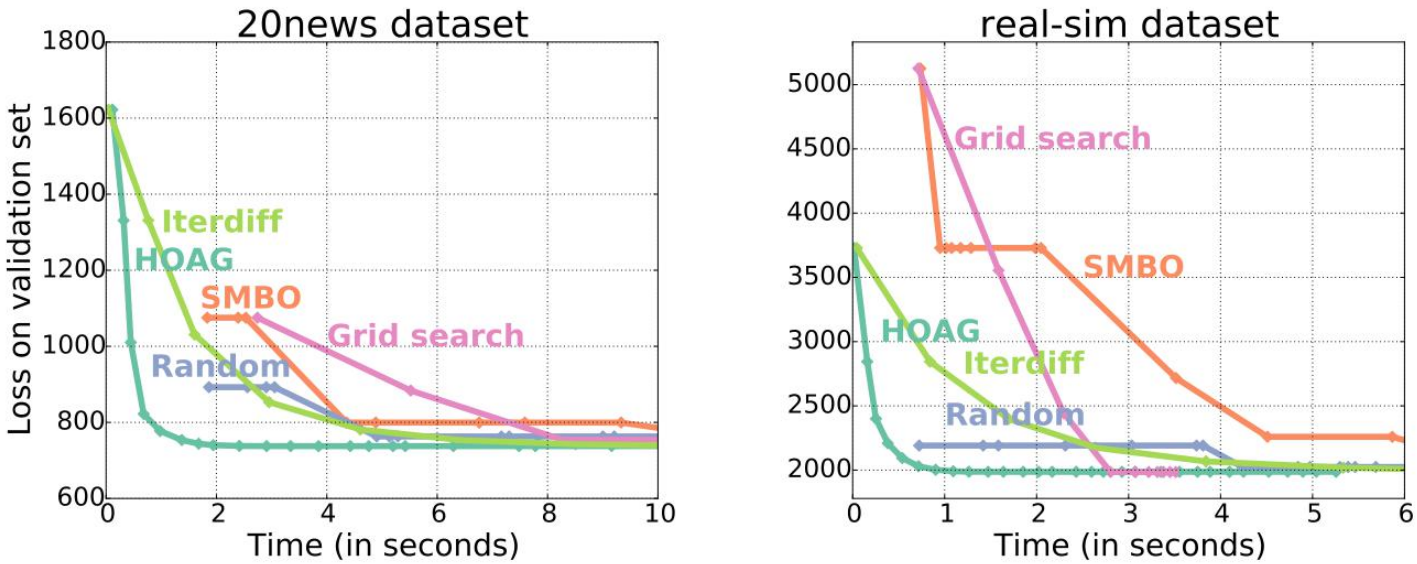

Figure: Convergence comparison for different methods