Semi-optimal on-line learning for restricted gradients

Stochastic Gradient Methods 2014

Noboru Murata

Waseda University

February 28, 2014

outline

problem setting for batch and on-line learning

statistical properties of batch learning

optimal learning rate for on-line learning

restricted gradient problem (e.g. Elo rating system)

concluding remarks

notation

- **data**: i.i.d observations from the ground truth distribution P

$$z_1, z_2, \ldots, z_t, \ldots \sim^{\text{i.i.d.}} P$$

- **learning machine**: specified by a finite dimensional parameter

$$\theta \in \Theta \subset \mathbb{R}^m$$

- **loss function**: penalty of a machine θ for a given datum z

$$l(z; \theta) \quad \text{(a smooth function with respect to } \theta\text{)}$$

for example:

$$l(z; \theta) = -\log p(z : \theta) \qquad \text{negative log loss}$$
$$l(z; \theta) = |y - f(x; \theta)|^2 \qquad \text{squared loss for } z = (x, y)$$

expected losses

- **population loss**: not accessible

$$L(\theta) = \mathbb{E}_{Z\sim P}[l(Z;\theta)]$$

$$\theta_* = \arg\min_{\theta} L(\theta) \quad \text{(optimal parameter)}$$

- **empirical loss**: accessible

$$\hat{L}_t(\theta) = \frac{1}{t}\sum_{z_i \in D_t} l(z_i;\theta), \quad D_t = \{z_i; i = 1, \ldots, t\}$$

- $\hat{L}$ is justified by *the law of large numbers*

$$\hat{L}_t(\theta) = \frac{1}{t}\sum_{z_i \in D_t} l(z_i;\theta) \xrightarrow{t\to\infty} L(\theta) = \mathbb{E}_{Z\sim P}\left[l(Z;\theta)\right]$$

batch and on-line learning

- **batch learning**: minimize the empirical loss

$$\hat{\theta}_t = \arg\min_{\theta} \hat{L}_t(\theta),$$

- **on-line learning**: update sequentially with a datum sampled at each time (or resampled from pooled data)

$$\theta_t = \theta_{t-1} - \Phi_t \nabla l(z_t; \theta_{t-1}),$$

where ∇ denotes the gradient with respect to θ, and Φ is a matrix which controls the rate of convergence.

asymptotic normality

Lemma (Godambe, 1991)

The distribution of $\hat{\theta}_t$ *converges to the normal distribution*

$$\hat{\theta}_t \sim \mathcal{N}\left(\theta_*, \frac{1}{t} V_*\right), \quad V_* = H^{-1} G H^{-1}$$

under some regularity condition, where

$$G = \mathbb{E}_{Z \sim P}\left[\nabla l(Z; \theta_*) \nabla l(Z; \theta_*)^{\mathrm{T}}\right],$$
$$H = \mathbb{E}_{Z \sim P}\left[\nabla \nabla l(Z; \theta_*)\right],$$

and θ_* *is the optimal parameter of the population loss:*

$$\theta_* = \arg\min_{\theta} L(\theta).$$

generalization error

Theorem

The expectation of the population loss is asymptotically given by

$$\mathbb{E}\left[L(\hat{\theta}_t)\right] = L(\theta_*) + \frac{1}{2t}\operatorname{Tr} GH^{-1} + o\left(\frac{1}{t}\right),$$

where the expectation is taken with respect to D_t*.*
The variance is asymptotically given by

$$\mathbb{V}\left[L(\hat{\theta}_t)\right] = \frac{1}{2t^2}\operatorname{Tr} GH^{-1}GH^{-1} + o\left(\frac{1}{t^2}\right).$$

training error

Theorem

The expectation of the empirical loss is asymptotically given by

$$\mathbb{E}\left[\hat{L}_t(\hat{\theta}_t)\right] = L(\theta_*) - \frac{1}{2t}\operatorname{Tr} GH^{-1} + o\left(\frac{1}{t}\right),$$

where the expectation is taken with respect to D_t.
The variance is asymptotically given by

$$\mathbb{V}\left[\hat{L}_t(\hat{\theta}_t)\right] = \frac{1}{t}\mathbb{V}_{Z\sim P}\left[l(Z;\theta_*)\right] + o\left(\frac{1}{t}\right).$$

generalized AIC

Corollary (Akaike, 1974)

The generalization error is estimated from the training error by correcting the bias as

$$L(\hat{\theta}_t) = \hat{L}_t(\hat{\theta}_t) + \frac{1}{t}\,\mathrm{Tr}\,GH^{-1}.$$

In the case of the maximum likelihood estimation, if the ground truth is realized by θ_**,*

$$L(\hat{\theta}_t) = \hat{L}_t(\hat{\theta}_t) + \frac{m}{t} \quad (m: \text{ dim. of } \theta),$$

because $H = G$*.*

recursive relation of consecutive estimates

Lemma (Bottou & Le Cun, 2005)

Let $\hat{\theta}_{t-1}$ and $\hat{\theta}_t$ be estimates for D_{t-1} and $D_t = D_{t-1} \cup \{z_t\}$. Then

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \frac{1}{t}\hat{H}_t^{-1}\nabla l(z_t; \hat{\theta}_{t-1}) + \mathcal{O}_p\left(\frac{1}{t^2}\right)$$

holds under some mild condition, where $\hat{H}_t$ is the empirical Hessian defined by

$$\hat{H}_t = \frac{1}{t}\sum_{z_i \in D_t} \nabla\nabla l(z_i; \hat{\theta}_{t-1}).$$

optimal on-line learning

- optimal design: Newton-Raphson $+ 1/t$-annealing

$$\Phi_t = \frac{1}{t}\hat{H}_t^{-1},$$

- on-line estimate of Hessian: (MLE case; Bottou, 1998)

$$\Phi_{t+1} = \Phi_t - \frac{\Phi_t \nabla l \nabla l^{\mathrm{T}} \Phi_t}{1 + \nabla l^{\mathrm{T}} \Phi_t \nabla l}$$

where $\nabla l = \nabla l(z_{t+1}; \theta_t)$

stochastic-BFGS (Nocedal, Wednesday talk), etc.

- rate of convergence: equivalent with batch learning (NM, 1998; NM & Amari, 1999; Bottou & Le Cun, 2005)

recursive relation of smooth functions

Lemma (Amari, 1967)

$$\mathbb{E}^{\theta_{t+1}}\left[f(\theta_{t+1})\right] = \mathbb{E}^{\theta_t}\left[f(\theta_t)\right] - \mathbb{E}^{\theta_t}\left[\nabla f(\theta_t)^{\mathrm{T}} \Phi_t \nabla L(\theta_t)\right] + \frac{1}{2}\operatorname{Tr}\mathbb{E}^{\theta_t}\left[\Phi_t G(\theta_t)\Phi_t^{\mathrm{T}} \nabla\nabla f(\theta_t)\right] + \mathcal{O}(\|\Phi_t\|^3)$$

holds for any smooth function $f(\theta)$, *where* $\mathbb{E}^{\theta}$ *denotes the expectation with respect to* θ, *and* $G(\theta)$ *is defined by*

$$G(\theta) = \mathbb{E}_{Z\sim P}\left[\nabla l(Z;\theta)\nabla l(Z;\theta)^{\mathrm{T}}\right].$$

linear operators for covariance analysis

Definition

Let A be an $m \times m$ square matrix and M be an $m \times m$ symmetric matrix. We define two linear operators as follows:

$$\Xi_A M = AM + (AM)^{\mathrm{T}},$$

$$\Omega_A M = AMA^{\mathrm{T}}.$$

recursive relations of parameter statistics

Lemma

Around the optimal parameter, the following approximated recursive relations for the expectation $\bar{\theta}_t = \mathbb{E}^{\theta_t}[\theta_t]$ *and the covariance* $V_t = \mathbb{V}^{\theta_t}[\theta_t]$ *hold:*

$$\bar{\theta}_{t+1} = \bar{\theta}_t - Q_t(\bar{\theta}_t - \theta_*),$$
$$V_{t+1} = V_t - \Xi_{Q_t} V_t + \Omega_{Q_t} V_* - \Omega_{Q_t}(\bar{\theta}_t - \theta_*)(\bar{\theta}_t - \theta_*)^{\mathrm{T}},$$

where

$$Q_t = \Phi_t H, \quad V_* = H^{-1} G H^{-1},$$

$$\Xi_A M = AM + (AM)^{\mathrm{T}},$$
$$\Omega_A M = AMA^{\mathrm{T}}.$$

convergence rate of $1/t$-annealing

Theorem

Let Φ be C/t, where C is a constant matrix. If $\lambda_{\min}(CH) \geq 1$, the leading terms are given by

$$\bar{\theta}_t = \theta_* + S_t(\theta_0 - \theta_*), \quad S_t = \prod_{\tau=2}^{t} \left(I - \frac{CH}{\tau} \right) = \mathcal{O}\left(\frac{1}{t^{\lambda_{\min}}} \right)$$

$$V_t = \left[\left(\Xi_{CH} - I \right)^{-1} \Omega_{CH} \right] \frac{1}{t} V_*,$$

where θ_0 is an initial parameter, and

$$V_* = H^{-1} G H^{-1}.$$

eigenvalues of operators

Lemma

Let $\lambda_i,\ i = 1, \ldots, m$ *be eigenvalues of* A. *The eigenvalues of* Ξ_A *and* Ω_A *are given by*

$$\Xi_A : \lambda_i + \lambda_j,\ i, j = 1, \ldots, m,$$
$$\Omega_A : \lambda_i \lambda_j,\ i, j = 1, \ldots, m.$$

Proof.

This follows by the relation

$$\mathrm{vec}(ABC) = (C^{\mathrm{T}} \otimes A) \,\mathrm{vec}\, B$$

for any $m \times m$ square matrices A, B, C. □

optimal design of $\Phi_t = C/t$

- larger $\lambda_{\min}$ is advantageous to faster convergence of $\bar{\theta}_t$.
- $(\Xi_{CH} - I)^{-1}\Omega_{CH}$ expands V_*/t, which is the minimum covariance attained by batch learning.
- eigenvalues of $(\Xi_{CH} - I)^{-1}\Omega_{CH}$ are given by

  $$\frac{\lambda_i \lambda_j}{\lambda_i + \lambda_j - 1},$$

  where λ_i's are eigenvalues of CH.
- if $C = H^{-1}$, all the eigenvalues of $(\Xi_I - I)^{-1}\Omega_I$ are equal to 1, i.e. $V_t = V_*/t$.
- $\Phi_t = H^{-1}/t$ is optimal.

equivalence to batch learning

- on-line learning:

$$\mathbb{E}\left[(\theta_t - \theta_*)(\theta_t - \theta_*)^{\mathrm{T}}\right] = \mathbb{V}\left[\theta_t\right] + \mathbb{E}\left[\theta_t - \theta_*\right]\mathbb{E}\left[\theta_t - \theta_*\right]^{\mathrm{T}}$$
$$= \frac{1}{t}V_* + \mathcal{O}\left(\frac{1}{t^2}\right).$$

- batch learning:

$$\mathbb{E}\left[(\hat{\theta}_t - \theta_*)(\hat{\theta}_t - \theta_*)^{\mathrm{T}}\right] = \frac{1}{t}V_* + \mathcal{O}\left(\frac{1}{t^2}\right).$$

rating systems

a method for evaluating the relative skill levels of players

- Elo rating: Arpad Elo, 1960
  used in competitor-versus-competitor games such as chess
  scores given to players are updated according to game results
- Glicko rating: Mark Glickman, 1997
  including confidence of estimated skill levels
- TrueSkill: Ralf Herbrich et al., 2007
  extension to multiplayer games
  skill levels are random variables (Bayesian framework)

model of Elo rating

- score: $\theta = (\theta^1, \theta^2, \dots)$
- event: $z_t = (a \succ b)$ (player a beats player b at time t)
- probability model:

$$\Pr(a \succ b) = P(z_t;\theta) = \frac{1}{1 + \exp(\gamma \cdot (\theta^b - \theta^a))},$$

  where γ is defined such that a player whose rating is 200 points greater than the other is expected to have a 75% chance of winning.
- loss function:

$$l(z_t;\theta) = -\log P(z_t;\theta) = \log(1 + \exp(\gamma \cdot (\theta^b - \theta^a)))$$

update rule of Elo rating

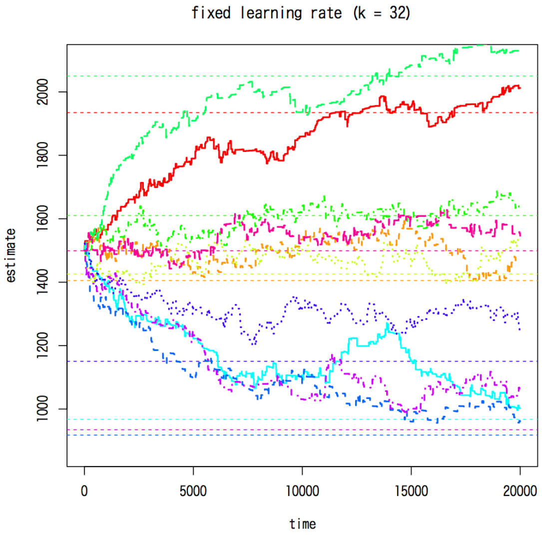
- gradient:

$$\frac{\partial}{\partial \theta^i} l(z_t;\theta) = \begin{cases} 0, & i \neq a, b \\ -\gamma \cdot (1 - P(z_t;\theta)), & i = a \text{ (winner)} \\ +\gamma \cdot (1 - P(z_t;\theta)), & i = b \text{ (looser)} \end{cases}$$

- update rule:

$$\begin{aligned} \theta_{t+1} &= \theta_t - \varepsilon \nabla l(z_t;\theta) \\ &= \theta_t + (0, \dots, \underbrace{\varepsilon\gamma(1-P)}_{a}, \dots, \underbrace{-\varepsilon\gamma(1-P)}_{b}, \dots, 0)^T \end{aligned}$$

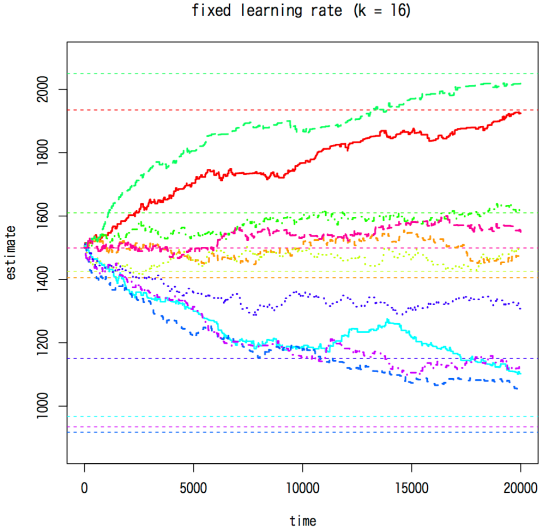
where $k = \varepsilon\gamma =$ 32 for novices, 16 for professionals.



fixed rate

$\Phi_t = \varepsilon I$

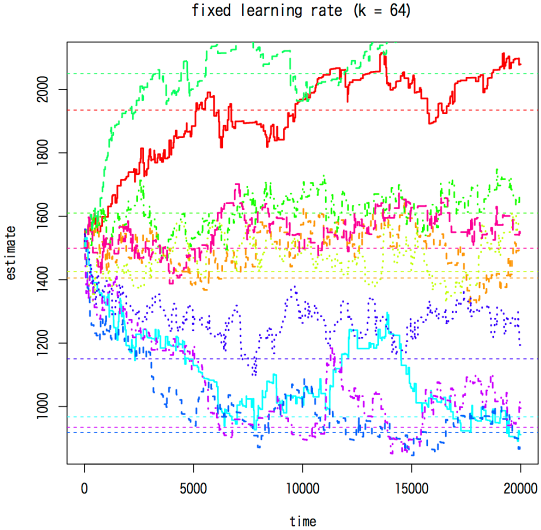
- 10 players out of 100
- 20000 games (400 [game/pl.])
- $k = 32, 16, 64$
- $\theta_0^i = 1500$



fixed rate

$\Phi_t = \varepsilon I$

- 10 players out of 100
- 20000 games (400 [game/pl.])
- $k = 32, 16, 64$
- $\theta_0^i = 1500$



fixed rate

$\Phi_t = \varepsilon I$

- 10 players out of 100
- 20000 games (400 [game/pl.])
- $k = 32, 16, 64$
- $\theta_0^i = 1500$

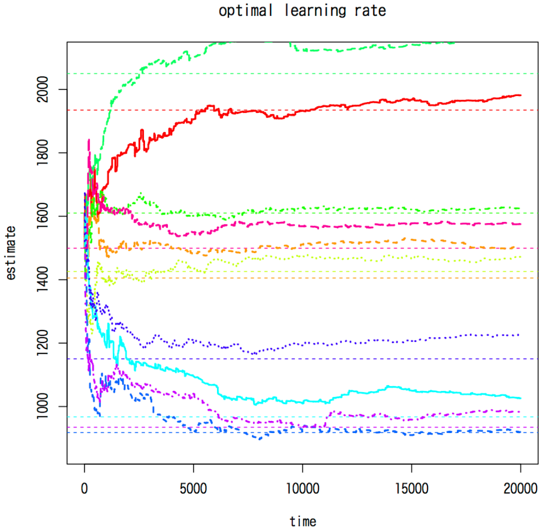
optimal update rule

- update rule: (Φ: matrix)

$$\begin{aligned}\theta_{t+1} &= \theta_t - \Phi_t \nabla l(z_t; \theta_t), \\ \Phi_{t+1} &= \Phi_t - \frac{\Phi_t \nabla l_t \nabla l_t^{\mathrm{T}} \Phi_t}{1 + \nabla l_t^{\mathrm{T}} \Phi_t \nabla l_t}, \\ \nabla l_t &= \nabla l(z_{t+1}; \theta_t) \\ &= (0, \ldots, \underbrace{\gamma(1-P)}_{a}, \ldots, \underbrace{-\gamma(1-P)}_{b}, \ldots, 0)^T\end{aligned}$$

- initial value:

$\Phi_0 = kI \quad I$ is the identity matrix



optimal rate

- 10 players out of 100
- 20000 games (400 [game/pl.])
- sensitive to initial kI

problem of semi-optimal update

- original update rule: $\Delta\theta = -\varepsilon \nabla l(z_t; \theta)$
  - only related players are updated: $\Delta\theta^i = 0,\ i \neq a, b$.
  - sum of θ is kept constant: $\mathbf{1}^{\mathrm{T}} \Delta\theta = 0$.
- optimal update rule: $\Delta\theta = -\Phi_t \nabla l(z_t; \theta)$
  - all the players are updated, because $\Phi_t = \hat{H}_t^{-1}/t$ is a dense matrix.
  - sum of θ is not necessarily kept constant.
- our problem: design Φ_t to fit the original restriction.

description of restrictions

- 1 vs 1 case: (players a and b)

$$\Delta\theta = \alpha \boldsymbol{a}, \quad \boldsymbol{a}^{\mathrm{T}} = \begin{array}{c} \begin{matrix} a & b & c & \end{matrix} \\ \begin{pmatrix} 1 & -1 & 0 & \cdots \end{pmatrix} \end{array},$$

or

$$B^{\mathrm{T}} \Delta\theta = 0, \quad B^{\mathrm{T}} = \begin{array}{c} \begin{matrix} a & b & c & d & \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & & & \ddots \end{pmatrix} \end{array}.$$

description of restrictions

- 2 vs 2 case: (players a+b and c+d)

$$\Delta\theta = A\alpha, \quad A^{\mathrm{T}} = \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & \end{array} \\ \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & -1 & 0 & \cdots \\ 0 & 1 & -1 & 0 & 0 & \cdots \end{pmatrix} \end{array},$$

or

$$B^{\mathrm{T}}\Delta\theta = 0, \quad B^{\mathrm{T}} = \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & & & & & \ddots \end{pmatrix} \end{array}.$$

problem formulation

Problem A

Find an "optimal" gradient $\Delta\theta = \Phi\nabla l(z;\theta)$ subject to

$$\Delta\theta \in \operatorname{Im} A, \quad (\Delta\theta = A\alpha,\ \alpha \in \mathbb{R}^k)$$

for a matrix $A \in \mathbb{R}^{m\times k}$.

Problem B

Find an "optimal" gradient $\Delta\theta = \Phi\nabla l(z;\theta)$ subject to

$$\Delta\theta \in \operatorname{Ker} B^{\mathrm{T}}, \quad (B^{\mathrm{T}}\Delta\theta = 0)$$

for a matrix $B \in \mathbb{R}^{m\times(m-k)}$,

cf. $f(\theta) = \text{const.} \Rightarrow \nabla f(\theta)^{\mathrm{T}}\Delta\theta = 0$

assumptions

- optimality is defined in terms of

  $$\text{minimize } \|H^{-1}\nabla l - \Delta\theta\|_M,$$

  where $\|x\|_M^2 = \langle x, x\rangle_M$ and $\langle x, y\rangle_M = \langle Mx, y\rangle$.
- M is chosen as H, because
  - quadratic approximation of population loss:

    $$\|\theta - \theta_*\|_H^2 = (\theta - \theta_*)^{\mathrm{T}} H(\theta - \theta_*) = L(\theta) - L(\theta_*)$$

  - Mahalanobis distance in maximum likelihood case:

    $$\mathbb{V}[\hat{\theta}_t] = \frac{1}{t} H^{-1} G H^{-1} = \frac{1}{t} H^{-1}$$

solutions

- decompose Φ_t into scalar and matrix parts as

$$\Phi_t = \varepsilon_t C, \quad \text{(e.g., } \varepsilon_t = 1/t\text{)}$$

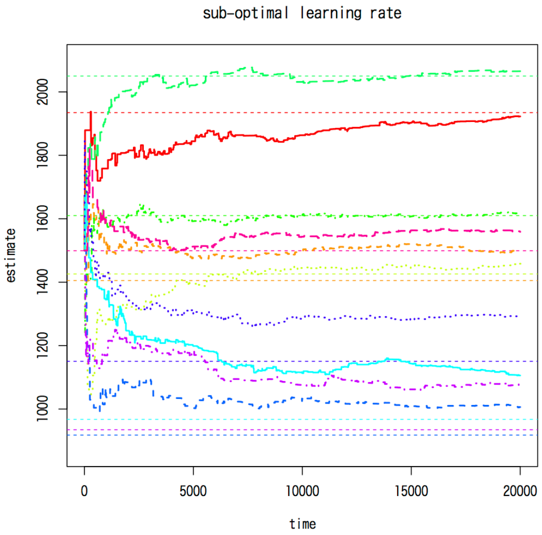
- solutions for the problems are:

Problem A

$$C_A = A(A^{\mathrm{T}} H A)^{-1} A^{\mathrm{T}}$$

Problem B

$$C_B = H^{-1} - H^{-1} B (B^{\mathrm{T}} H^{-1} B)^{-1} B^{\mathrm{T}} H^{-1}$$



sub-optimal rate

- 10 players out of 100
- 20000 games (400 [game/pl.])

notes on solutions

- C_A and C_B are symmetric (only when $M = H$).
- $C_A H$ or $C_B H$ is a projection matrix:

$$\lambda = \begin{cases} 1, & v \in \operatorname{Im} A \text{ or } \operatorname{Ker} B, \\ 0, & \text{otherwise.} \end{cases}$$

- if k is small, calculation of C_A is more efficient than that of C_B
- only a few parameters are updated, however convergence is as good as optimal case
  (information loss is quite small in some case)

concluding remarks

- we have investigated:
  - dynamics of convergence phase of on-line learning,
  - conditions for optimal convergence rate,
  - optimal projection of gradients to subspaces,
- practical applications would be:
  - skill level rating systems,
  - on-line learning for Bradley-Terry model,
  - distributed control systems.