Scalable Kernel Methods via Doubly Stochastic Gradients

Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj,
Maria-Florina Balcan, Le Song. NIPS-2014

Zoltán Szabó

Machine Learning Journal Club, Gatsby

October 20, 2014

Outline

- Motivation.
- Notations, objective.
- Algorithm.
- Error bounds.
- Numerical experiences.

Motivation

- Large-scale, efficient neural nets: $\approx$ no theory.
- **Goal**: scale kernel methods up.
- Previous work:
  - low-rank approximation, RND features:
    - limited generalization ability,
    - rank/#of RND features can be $\mathcal{O}$(sample#).
  - BCD in the dual form: one might have to store all SVs for testing (=whole training set!).

Notations: kernel

- Kernel: $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, if $\exists \varphi : \mathcal{X} \to H$(ilbert) such that

$$k(a, b) = \langle \varphi(a), \varphi(b) \rangle_H . \tag{1}$$

- H not necessarily unique, but $\exists! H = H(k)$ RKHS:
  1. generators: $k(x, \cdot) \in \mathcal{H}$ $(\forall x \in \mathcal{X})$,
  2. reproducing property: $\langle f, k(x, \cdot) \rangle_H = f(x)$ $(\forall f \in H)$.

RND feature based kernel construction

- Let
  - $\mathbb{P}_\Omega$: measure on Ω,
  - $\phi_\omega : \mathcal{X} \to \mathbb{R}$, $\phi_\omega \in L^2(\Omega, \mathbb{P})$.

  Then

$$k(x, x') = \int_\Omega \phi_\omega(x)\phi_\omega(x')\mathrm{d}\mathbb{P}_\Omega(\omega)$$

  is a kernel on $\mathcal{X}$.
- Example: for $\Omega = \mathbb{R}^d$, $\phi_\omega(x) = e^{i\omega^T x}$, we get the translation invariant kernels (Bochner T.).

Problem

- Objective: We want to solve ($\lambda > 0$)

$$R(f) = \mathbb{E}_{(x,y)}\left[l(f(x), y)\right] + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \to \min_{f \in \mathcal{H} = \mathcal{H}(k)},$$

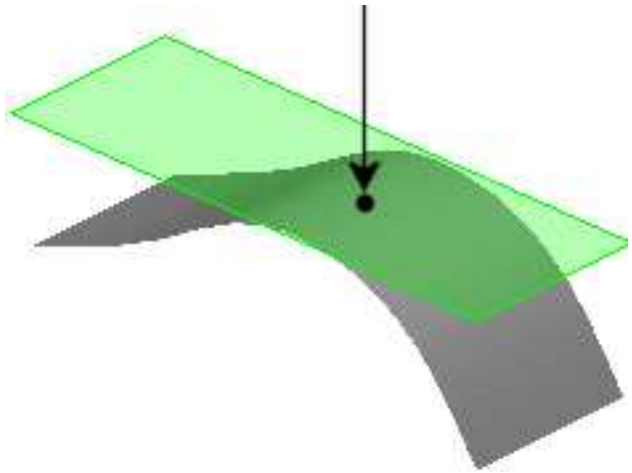
  where the $l(u, y) \in \mathbb{R}$ loss is convex in u.
- Examples (l):
  - $l(u, y) = |u - y|_\epsilon$: SVMR,
  - $l(u, y) = \ln(1 + e^{-uy})$: logistic regression,
  - $l(u, y) = (u - y)^2$: ridge regression.

Functional gradient

- Convexity of $l \Rightarrow \exists$ subgradient of l w.r.t. $u =: l'(u, y)$.
- Optimization: (doubly) stochastic gradient descent.
- Functional gradient $[\nabla R(f)]$: the gradient of $R : \mathcal{H} \to \mathbb{R}$ at $f \in \mathcal{H}$

$$R(f + \epsilon g) = R(f) + \epsilon \langle \nabla R(f), g \rangle_{\mathcal{H}} + \mathcal{O}\left(\epsilon^2\right).$$

- View: $\nabla R(f) \in \mathcal{H}$ (Riesz repr. T).



Functional gradient: example-1

- Target (R): x is fixed; $R(f) = f(x)$.
- Gradient of R:

$$\begin{aligned} R(f + \epsilon g) &= (f + \epsilon g)(x) = f(x) + \epsilon g(x) \\ &= f(x) + \epsilon \langle k(x, \cdot), g \rangle_{\mathcal{H}} + 0. \end{aligned}$$

- Result: $\nabla R(f) = k(x, \cdot)$.

Functional gradient: example-2

- Target (R): $R(f) = \|f\|_{\mathcal{H}}^2$.
- Gradient of R:

$$\begin{aligned} R(f+\epsilon g) &= \|f+\epsilon g\|_{\mathcal{H}}^2 = \langle f+\epsilon g, f+\epsilon g\rangle_{\mathcal{H}} \\ &= \|f\|_{\mathcal{H}}^2 + 2\langle f, \epsilon g\rangle_{\mathcal{H}} + \epsilon^2 \|g\|_{\mathcal{H}}^2 \\ &= \|f\|_{\mathcal{H}}^2 + \epsilon \langle 2f, g\rangle_{\mathcal{H}} + \mathcal{O}\left(\epsilon^2\right). \end{aligned}$$

- Result: $\nabla R(f) = 2f$.

Back to our objective function

- Objective function: $R(f) = \mathbb{E}_{(x,y)}\left[l(f(x), y)\right] + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2$.
- Gradient:

$$\nabla R(f) = \mathbb{E}_{(x,y)}\left[l'(f(x), y)k(x, \cdot)\right] + \lambda f.$$

- Stochastic gradient: given $(x, y) \sim \mathbb{P}$

$$l'(f(x), y)k(x, \cdot) + \lambda f(\cdot) =: \xi(\cdot) + \lambda f(\cdot).$$

- Doubly stochastic gradient: given $(x, y) \sim \mathbb{P}$, $\omega \sim \mathbb{P}_\Omega$

$$l'(f(x), y)\phi_\omega(x)\phi_\omega(\cdot) + \lambda f(\cdot) =: \zeta(\cdot) + \lambda f(\cdot).$$

Functional gradient approximation: properties

$\xi \in \mathcal{H}$, $\zeta \notin \mathcal{H}$, unbiasedness:

$$\begin{aligned}
\xi(\cdot) &= l'(f(x), y)k(x, \cdot) \in \mathcal{H}, \\
\zeta(\cdot) &= l'(f(x), y)\phi_\omega(x)\phi_\omega(\cdot) \notin \mathcal{H}, \\
\xi(\cdot) &= \mathbb{E}_\omega[\zeta(\cdot)], \\
\nabla R(f) &= \mathbb{E}_{(x,y)}[\xi(\cdot)] + \lambda f(\cdot), \\
\nabla R(f) &= \mathbb{E}_{(x,y)}\mathbb{E}_\omega\left[\zeta(\cdot)\right] + \lambda f(\cdot),
\end{aligned}$$

Functional gradient descent

- $\gamma_i > 0$: learning rates.
- Stochastic gradient descent $[(f_{i-1}, x_i, y_i, \gamma_i) \to f_i]$:

$$\begin{aligned} f_i &= f_{i-1} - \gamma_i \nabla \hat{R}(f_{i-1}; x_i, y_i) \\ &= f_{i-1} - \gamma_i [l'(f_{i-1}(x_i), y_i) k(x_i, \cdot) + \lambda f_{i-1}] \\ &= (1 - \gamma_i \lambda) f_{i-1} - \gamma_i l'(f_{i-1}(x_i), y_i) k(x_i, \cdot). \end{aligned}$$

- Doubly stochastic gradient descent $[(f_{i-1}, x_i, y_i, \omega_i, \gamma_i) \to f_i]$:

$$\begin{aligned} f_i &= f_{i-1} - \gamma_i [l'(f_{i-1}(x_i), y_i) \phi_{\omega_i}(x_i) \phi_{\omega_i}(\cdot) + \lambda f_{i-1}] \\ &= (1 - \gamma_i \lambda) f_{i-1} - \gamma_i l'(f_{i-1}(x_i), y_i) \phi_{\omega_i}(x_i) \phi_{\omega_i}(\cdot). \end{aligned}$$

Doubly stochastic gradient based f_i

- Obtained update equation:

$$f_i = (1 - \gamma_i \lambda) f_{i-1} - \gamma_i l'(f_{i-1}(x_i), y_i) \phi_{\omega_i}(x_i) \phi_{\omega_i}(\cdot).$$

- Assuming

$$f_i(\cdot) = \sum_{j=1}^{i} \beta_j \hat{k}(x_j, \cdot) = \sum_{j=1}^{i} \left[\beta_j \phi_{\omega_j}(x_j)\right] \phi_{\omega_j}(\cdot) =: \sum_{j=1}^{i} \alpha_j \phi_{\omega_j}(\cdot),$$

our $f_{i-1} \to f_i$ update (in terms of α_j-s) is

$$\alpha_i = -\gamma_i l'(f_{i-1}(x_i), y_i) \phi_{\omega_i}(x_i),$$
$$\alpha_j = (1 - \gamma_i \lambda) \alpha_j \quad (j = 1, \ldots, i-1).$$

Algorithm: Training

- Given $\{(x_i, y_i)\}_{i=1}^t$ $[(x_i, y_i) \sim \mathbb{P}]$ compute $\{\alpha_i\}_{i=1}^t$.
- The RND number generation $(\Rightarrow \omega_i)$ is "cached" by seed i.

Algorithm 1 Train. $(\mathbb{P}, l, \lambda) \Rightarrow \{\alpha_i\}_{i=1}^t$.

---

for $i = 1, \ldots, t$ **do**

Sample $(x_i, y_i) \sim \mathbb{P}$.

Sample $\omega_i \in P_\Omega$ using seed i.

$f(x_i) = \mathit{Predict}(x_i, \{\alpha_j\}_{j=1}^{i-1})$.

$\alpha_i = -\gamma_i l'(f(x_i), y_i)\phi_{\omega_i}(x_i)$.

$\alpha_j = (1 - \gamma_i \lambda)\alpha_j \quad (j = 1, \ldots, i-1)$.

Algorithm: Predict

- Predict using *the same* seeds as in training $\Rightarrow$
- There is no need to store ω_i-s.

Algorithm 2 Predict. $(x, \{\alpha_i\}_{i=1}^t) \Rightarrow f(x)$.

Initialization: $f(x) = 0$.
for $i = 1, \ldots, t$ **do**
  Sample $\omega_i \in P_\Omega$ using seed i.
  $f(x) = f(x) + \alpha_i \phi_{\omega_i}(x)$.

Theoretical guarantees: conditions

- $\exists f_* = \arg\min_{f \in \mathcal{H}} R(f)$.
- $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ Lipschitz continuous in u: $\exists L$ such that

$$|l(u, y) - l(u', y)| \leq L|u - u'| \quad (\forall u, u', y \in \mathbb{R}).$$

- Bounded l': $\exists B_l$ such that $|l'(f_i(x_i), y_i)| \leq B_l$.
- Bounded kernel, RND feature: $\exists B_k, B_\phi$ such that

$$k(x, x') \leq B_k \quad (\forall x, x' \in \mathcal{X}),$$
$$|\phi_\omega(x)\phi_\omega(x')| \leq B_\phi \quad (\forall x, x' \in \mathcal{X}, \omega \in \Omega).$$

Example (Gaussian k): $B_k = 1$, $B_\phi = 2$.

Theoretical guarantees: in human-readable format

Let $\gamma_i = \frac{\theta}{i}$ with $\theta > 0$, $D^t = \{(x_i, y_i)\}_{i=1}^t$, $\omega^t = \{\omega_i\}_{i=1}^t$.

- Convergence to the optimal function: for any $x \in \mathcal{X}$

$$\mathbb{E}_{D^t,\omega^t}\left[|f_{t+1}(x) - f_*(x)|^2\right] \leq \frac{C_1}{t},$$

$$|f_{t+1}(x) - f_*(x)|^2 \precsim \frac{C_2}{t} \text{ (with high probability).}$$

- Generalization error (risk): let $R_{true} = \mathbb{E}_{(x,y)}[l(f(x), y)]$,

$$R_{true}(f_{t+1}) - R_{true}(f_*) \precsim \frac{1}{\sqrt{t}} \text{ (with high probability).}$$

High-level proof idea

Recall the f_t-update, and define h_t as

$$f_{t+1}(\cdot) = f_t - \gamma_t[\zeta_t(\cdot) + \lambda f_t(\cdot)] = \sum_{i=1}^{t} a_t^i \zeta_i(\cdot), \quad (t > 1), \quad f_1(\cdot) = 0,$$

$$a_t^i = -\gamma_i \prod_{j=i+1}^{t} (1 - \gamma_j \lambda),$$

$$h_{t+1}(\cdot) = \sum_{i=1}^{t} a_t^i \xi_i(\cdot) = h_t - \gamma_t[\xi_t(\cdot) + \lambda h_t(\cdot)] \quad (t > 1), \quad h_1(\cdot) = 0,$$

Since $\xi_i(\cdot) \in \mathcal{H}$, $h_t \in \mathcal{H}$ $(\forall t \geq 0)$!

High-level proof idea: $f_{t+1} - f_*$ through h_{t+1}

$$|f_{t+1}(x) - f_*(x)|^2 \leq \underbrace{2|f_{t+1}(x) - h_{t+1}(x)|^2}_{\text{random functions}} + \underbrace{2\,\|h_{t+1} - f_*\|_{\mathcal{H}}^2\, B_k}_{\text{random data}}.$$

High-level proof idea: $f_{t+1} - f_*$ through h_{t+1}

$$
\begin{aligned}
|f_{t+1}(x) - f_*(x)|^2 &= |f_{t+1}(x) - h_{t+1}(x) + h_{t+1}(x) - f_*(x)|^2 \\
&\leq 2\left[|f_{t+1}(x) - h_{t+1}(x)|^2 + |h_{t+1}(x) - f_*(x)|^2\right], \\
|h_{t+1}(x) - f_*(x)|^2 &= |\left\langle h_{t+1}, k(x,\cdot)\right\rangle_{\mathcal{H}} - \left\langle f_*, k(x,\cdot)\right\rangle_{\mathcal{H}}|^2 \\
&= |\left\langle h_{t+1} - f_*, k(x,\cdot)\right\rangle_{\mathcal{H}}|^2 \\
&= [\|h_{t+1} - f_*\|_{\mathcal{H}} \|k(x,\cdot)\|_{\mathcal{H}}]^2 \\
&= \|h_{t+1} - f_*\|_{\mathcal{H}}^2 \left\langle k(x,\cdot), k(x,\cdot)\right\rangle_{\mathcal{H}} \\
&= \|h_{t+1} - f_*\|_{\mathcal{H}}^2 k(x,x) \\
&\leq \|h_{t+1} - f_*\|_{\mathcal{H}}^2 B_k. \Rightarrow \\
|f_{t+1}(x) - f_*(x)|^2 &\leq \underbrace{2|f_{t+1}(x) - h_{t+1}(x)|^2}_{\text{random functions}} + \underbrace{2\|h_{t+1} - f_*\|_{\mathcal{H}}^2 B_k}_{\text{random data}}.
\end{aligned}
$$

High-level proof idea

- Tricky part (due to the random functions, our focus):

$$|f_{t+1}(x) - h_{t+1}(x)|^2.$$

- Second term: stoch. approximation in RKHS ("standard").

By the definitions of f_{t+1} and h_{t+1}:

$$f_{t+1}(x) - h_{t+1}(x) = \sum_{i=1}^{t} a_t^i [\zeta_i(x) - \xi_i(x)] =: \sum_{i=1}^{t} V_i(x).$$

$\{V_i(x)\}_i$ is not i.i.d. (see a_t^i, f_i), but "almost" $\Rightarrow$ Concentration.

Martingal, martingal difference

$M_0, M_1, M_2, \ldots$ is *martingal* if

$$\begin{aligned} \mathbb{E}[|M_n|] &< \infty, \quad (\forall n), \\ \mathbb{E}[M_{n+1}|M_n, \ldots, M_1] &= M_n, \quad (\forall n). \Leftrightarrow \\ \mathbb{E}[M_{n+1} - M_n|M_n, \ldots, M_1] &= 0, \quad (\forall n). \end{aligned}$$

$M_n = \sum_{i=1}^{n} V_i$. Example: random walk.

Martingal, martingal difference

$M_0, M_1, M_2, \ldots$ is *martingal* if

$$\begin{aligned} \mathbb{E}[|M_n|] &< \infty, \quad (\forall n), \\ \mathbb{E}[M_{n+1}|M_n, \ldots, M_1] &= M_n, \quad (\forall n). \Leftrightarrow \\ \mathbb{E}[M_{n+1} - M_n|M_n, \ldots, M_1] &= 0, \quad (\forall n). \end{aligned}$$

$V_n (= M_n - M_{n-1})$ is *martingal difference* if

$$\begin{aligned} \mathbb{E}[|V_n|] &< \infty, \\ \mathbb{E}[V_n|V_{n-1}, \ldots] &= 0. \end{aligned}$$

Equivalently, $M_n = \sum_{i=1}^{n} V_i$. Example: random walk.

Azuma-Hoeffding inequality

Let $\{V_i\}_i$ be a bounded martingal difference sequence ($|V_i| \leq c_i$). Then $\forall \epsilon > 0$

$$\mathbb{P}\left(\left|\sum_{i=1}^{t} V_i\right| \geq \epsilon\right) \leq 2e^{-\frac{\epsilon^2}{\sum_{i=1}^{t} c_i^2}}.$$

Let $V_i(x) = a_t^i[\zeta_i(x) - \xi_i(x)]$ (x: fixed). $V_i(x)$ is

1. bounded:

$$\begin{aligned}
|\zeta_i(x) - \xi_i(x)| &= \left|l'(f_i(x_i), y_i)k(x_i, x) - l'(f_i(x_i), y_i)\phi_{\omega_i}(x_i)\phi_{\omega_i}(x)\right| \\
&\leq \left|l'(f_i(x_i), y_i)\right| \left[|k(x_i, x)| + |\phi_{\omega_i}(x_i)\phi_{\omega_i}(x)|\right] \\
&\leq B_l(B_k + B_\phi) \Rightarrow \\
|V_i| &\leq |a_t^i| B_l(B_k + B_\phi) =: c_i.
\end{aligned}$$

2. mart. difference: $\mathbb{E}[V_i(x)|V_{i-1}(x), \ldots] = 0$ ($\Leftarrow$ unbiasedness).

Azuma-Hoeffding inequality applied to $\{V_i(x)\}_i$

$$\mathbb{E}\left[|f_{t+1}(x) - h_{t+1}(x)|^2\right] = \mathbb{E}\left[\left|\sum_{i=1}^{t} V_i\right|^2\right] = \int_0^\infty \mathbb{P}\left(\left|\sum_{i=1}^{t} V_i\right|^2 \geq \epsilon\right) \mathrm{d}\epsilon$$

$$\leq \int_0^\infty 2e^{-\frac{2\epsilon}{\sum_{i=1}^t c_i^2}} \mathrm{d}\epsilon = \sum_{i=1}^{t} c_i^2$$

using with $Z \geq 0$ (F_Z: cdf of Z)

$$\mathbb{E}[Z] = \int_0^\infty 1 - F_Z(z)\mathrm{d}z,$$

$$\mathbb{P}\left(\left|\sum_{i=1}^{t} V_i\right| \geq \epsilon\right) \leq 2e^{-\frac{\epsilon^2}{\sum_{i=1}^t c_i^2}},$$

$$\int_0^\infty 2e^{-\frac{2\epsilon}{c}} \mathrm{d}\epsilon = 2\left[-\frac{c}{2}e^{-\frac{2\epsilon}{c}}\right]_{\epsilon=0}^{\epsilon=\infty} = c.$$

- Thus, $\mathbb{E}\left[|f_{t+1}(x) - h_{t+1}(x)|^2\right] \leq \sum_{i=1}^{t} c_i^2$, where

$$c_i = |a_t^i| B_l (B_k + B_\phi), \qquad a_t^i = -\gamma_i \prod_{j=i+1}^{t} (1 - \gamma_j \lambda).$$

- Freedom in the choice of γ_i!
- If $\gamma_i = \frac{\theta \geq 0}{i}$, where $\theta\lambda \in (1,2) \cup \mathbb{Z}^+$, then (induction) $|a_t^i| \leq \frac{\theta}{t}$.
- In this case

$$\sum_{i=1}^{t} |a_t^i|^2 \leq \sum_{i=1}^{t} \frac{\theta^2}{t^2} = \frac{\theta^2}{t},$$

$$\mathbb{E}\left[|f_{t+1}(x) - h_{t+1}(x)|^2\right] \leq [B_l(B_k + B_\phi)]^2 \frac{\theta^2}{t} = \mathcal{O}\left(\frac{1}{t}\right).$$

Error due to random data: $\mathbb{E}\left[\|h_t - f_*\|_{\mathcal{H}}^2\right]$

- Three gradient terms:

$$g_t := \xi_t + \lambda h_t = l'(f_t(x_t), y_t)k(x_t, \cdot) + \lambda h_t,$$
$$\hat{g}_t := \hat{\xi}_t + \lambda h_t := l'(h_t(x_t), y_t)k(x_t, \cdot) + \lambda h_t,$$
$$\bar{g}_t := \mathbb{E}[\hat{g}_t] = \mathbb{E}[l'(h_t(x_t), y_t)k(x_t, \cdot)] + \lambda h_t.$$

- By the definition of h_{t+1}: $h_{t+1} = h_t - \gamma_t g_t$ $(t \geq 1).\} \Rightarrow$
- Recursion to $A_{t+1} = \|h_{t+1} - f_*\|_{\mathcal{H}}^2$, and to its expectation $e_t = \mathbb{E}[A_t] = \mathbb{E}\left[\|h_t - f_*\|_{\mathcal{H}}^2\right]$. This gives $e_t = \mathcal{O}\left(\frac{1}{t}\right)$, similarly.

Generalization error

$$
\begin{aligned}
R_{true}(f_{t+1}) - R_{true}(f_*) &= \mathbb{E}_{(x,y)}\left[l(f_{t+1}(x), y)\right] - \mathbb{E}_{(x,y)}\left[l(f_*(x), y)\right] \\
&= \mathbb{E}_{(x,y)}\left[l(f_{t+1}(x), y) - l(f_*(x), y)\right] \\
&\leq \mathbb{E}_{(x,y)}\left[L|f_{t+1}(x) - f_*(x)|\right] \\
&= L\mathbb{E}_x|(f_{t+1}(x) - f_*(x)| \\
&= L\sqrt{\mathbb{E}_x|(f_{t+1}(x) - f_*(x)|^2} = L\left\|f_{t+1} - f_*\right\|_2 .
\end{aligned}
$$

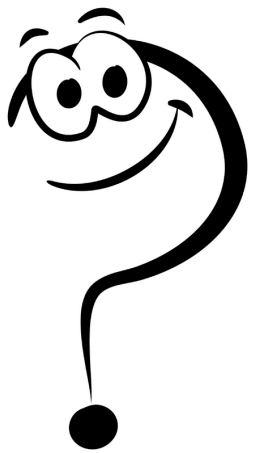
Similarly to the previous proof:

$$
\left\|f_{t+1} - f_*\right\|_2^2 \leq c_1\left\|f_{t+1} - h_{t+1}\right\|_2^2 + c_2\left\|h_{t+1} - f_*\right\|_{\mathcal{H}}^2 .
$$

Numerical experiences

- Problems: SVM, ridge regression, logistic regression.
- Baselines:
  - online kernel algorithms (NORMA, SDCA, Pegasos).
  - deep learning heuristics.
- Experience:
  - Similar performance, less computation/memory.
  - Mini-batching is useful.

Thank you for the attention!



Subgradient of a convex function

- Let $f : U \to \mathbb{R}$ be a convex function (U: convex). A vector v is called a *subgradient* of f at x_0 if

$$f(x) - f(x_0) \geq \langle v, x - x_0 \rangle \quad (\forall x \in U).$$

- $\partial f(x_0)$: Non-empty, convex, compact set.