A consensus-based global optimization method for high dimensional machine learning problems

Yuhua Zhu

Joint work with Jose Carrilo, Shi Jin, Lei Li

Oct 22, 2019
Motivations

The Model and algorithm

Numerical experiments
Motivations

The Model and algorithm

Numerical experiments
Goal: find $x^* = \arg\min_x L(x)$, $L(x)$ is a non-convex function.

For example: $L(x) = \frac{1}{n} \sum_i l_i(x)$

Why non-gradient method?

- Gradient is hard to calculate
- Objective function is non-smooth
- Flat local minimum
It is hard for gradient based method to escape from flat local minimum

GD: \( X'(t) = -\nabla L(X(t)) \)

SGD: \( dX_t = -\nabla L(X_t) + \sqrt{\frac{1}{\beta}} dB_t \)

p.d.f of SGD: \( \partial_t p(t, x) = \nabla \cdot \left[ \nabla L(x)p + \frac{2}{\beta} \nabla p \right] \)

\( p^\infty(x) = \frac{1}{Z} e^{-\frac{\beta}{2} L(x)} \)

When \( \frac{\sqrt{\det H_1}}{\sqrt{\det H_2}} < \frac{e^{\frac{\beta}{2} L_1}}{e^{\frac{\beta}{2} L_2}} \), SGD is more likely to converge to the flat local minimum.

\[ \frac{\mathbb{P}(\text{converge to } X^1)}{\mathbb{P}(\text{converge to } X^2)} = \frac{\sqrt{\det H_2}}{\det H_1} e^{\frac{\beta}{2} (L_2 - L_1)} \]

\[ > 1 \quad < 1 \]
It is hard for gradient based method to escape from flat local minimum

Example:

\[ \ell(x, \hat{x}_i) = e^{\sin(2x^2)} + \frac{1}{10}(x - \hat{x}_i - \frac{\pi}{2})^2, \quad \hat{x}_i \sim N(0, 0.1) \]

\[ L(x) = \frac{1}{n} \sum_i \ell(x, \hat{x}_i) \]

Success rate for SGD to find the correct global minimum is 18%
Motivations

The Model and algorithm

Numerical experiments
For \( j = 1, \cdots, N \)

\[
dX^j = -\lambda(X^j - \bar{x}^*)H^e(L(X^j) - L(\bar{x}^*))\, dt + \sigma|X^j - \bar{x}^*|dW^j
\]

where \( \bar{x}^* = \frac{1}{\sum_{j=1}^{N} e^{-\beta L(X^j)}} \sum_{j=1}^{N} X^j e^{-\beta L(X^j)}. \)

Relax to their weighted average, in the meantime, explore their surrounding environment.

Require \( \lambda \sim O(d) \) to guarantee the convergence of the method

Bad for high-dimensional problems
\[ dX^j = -\lambda (X^j - \bar{x}^*) \, dt + \sigma \sum_{k=1}^{d} (X^j - \bar{x}^*)_k dW^j_k \bar{e}_k \]

First improvement

- Intuitively, now the diffusivity allows the particles to explore each dimension with different rate, so more possible to find the global minimum.

component-wise geometric Brownian motion
Previous model

Assume $x^* = a$ is a constant.

$$dX = -\lambda (X - a) \, dt + \sigma |X - a| dW^j$$

For each dimension $i$

$$d[(X)_i - (a)_i] = -\lambda [(X)_i - (a)_i] \, dt + \sigma |X - a| d(W^j)_i$$

By Ito’s formula and then take expectation

$$dE[(X)_i - (a)_i]^2 = -2\lambda E[(X)_i - (a)_i]^2 dt + \sigma^2 E|X - a|^2 dt$$

Sum over all dimension

$$\frac{d}{dt} E|X - a|^2 = -2\lambda E|X - a|^2 + \sigma^2 \sum_{i=1}^{d} E|X - a|^2 = (-2\lambda + \sigma^2 d) E|X - a|^2$$

New model

$$dX = -\lambda (X - a) \, dt + \sigma \sum_{k=1}^{d} (X^j - a)_k dW^j_k \bar{e}_k$$

For each dimension $i$

$$d[(X)_i - (a)_i] = -\lambda [(X)_i - (a)_i] \, dt + \sigma [(X)_i - (a)_i] d(W^j)_i$$

By Ito’s formula and then take expectation

$$dE[(X)_i - (a)_i]^2 = -2\lambda E[(X)_i - (a)_i]^2 dt + \sigma^2 E[(X)_i - (a)_i]^2 dt$$

Sum over all dimension

$$\frac{d}{dt} E|X - a|^2 = -2\lambda E|X - a|^2 + \sigma^2 \sum_{i=1}^{d} E(X - a)_i^2 = (-2\lambda + \sigma^2) E|X - a|^2$$

$2\lambda > d\sigma^2$

$2\lambda > \sigma^2$

[Carrillo-Choi-Totzeck-Tse, 18]
Mean field limit of the continuous model

\[ dX^j = -\lambda (X^j - \bar{x}^*) \, dt + \sigma \sum_{k=1}^{d} (X^j - \bar{x}^*)_k dW^j_k \bar{e}_k \]

\[ N \to \infty \]

\[ dX = -\lambda (X - X^*) \, dt + \sigma \sum_{i=1}^{d} \bar{e}_i (X - X^*)_i dW_i \]

with \( X^* = \frac{\mathbb{E}(X e^{-\beta L(X)})}{\mathbb{E}(e^{-\beta L(X)})} \).

**Theorem:** [Carrilo-Jin-Li-Z, 19]

Under some condition on the initial distribution of \( X \) and \( \lambda, \sigma, X(t) \to \tilde{x} \) exponentially fast and and,

\[ L(\tilde{x}) \leq -\frac{1}{\beta} \log \mathbb{E} e^{-\beta L(X(0))} + \frac{\log 2}{\beta} \leq L(x^*) + O(\beta^{-1}) \]

- The initial law of \( X \)
- The largeness of \( \beta \)
Numerical method
A gradient-free optimization method

Goal: find $x^* = \arg\min_x L(x) = \arg\min_x \frac{1}{n} \sum_i l_i(x)$

Algorithm [Carrillo-Jin-Li-Z-19]

Initially, randomly generate $N$ particles $X^j$, at each step we randomly update $M$ particles.

- Calculate $L(X^j), j = 1, \ldots, N$.

$$\hat{L}(X^j) = \frac{1}{m} \sum_{i \in b} L(x), b \subset \{1, \ldots, n\}.$$  \[O(n)\]

- Find a weighted average: $\bar{X}^* = \frac{1}{\sum_{j=1}^N \mu^j} \sum_{j=1}^N X^j \mu^j$, $\mu^j = e^{-\beta \hat{L}(X^j)}$  \[O(N)\]

- Let $X^j$ move towards $X^*$ and explore their neighbor at the same time.

$$X^j \leftarrow X^j - \lambda \gamma (X^j - \bar{X}^*) + \sigma \sqrt{\gamma} \sum_{i=1}^d \tilde{e}_i (X^j - \bar{X}^*)_i z_i, \quad z_i \sim \mathcal{N}(0, 1)$$  \[O(1)\]
Motivations

The Model and algorithm

Numerical experiments
Example:

\[ \ell(x, \hat{x}_i) = e^{\sin(2x^2)} + \frac{1}{10}(x - \hat{x}_i - \frac{\pi}{2})^2, \quad \hat{x}_i \sim N(0, 0.1) \]

\[ L(x) = \frac{1}{n} \sum_i \ell(x, \hat{x}_i) \]

Success rate of our method is 98%!
(with N = 100, M = 20)
\[ L(x) = \frac{1}{d} \sum_{i=1}^{d} \left[ (x_i - B)^2 - 10 \cos(2\pi(x_i - B)) + 10 \right] + C \]

**Table 2.** Rastrigin function in \( d = 20 \) with \( \alpha = 30 \).

<table>
<thead>
<tr>
<th>( x^* )</th>
<th>( \frac{1}{d} \mathbb{E} [|v_f(T) - x^*|^2] )</th>
<th>( N = 50 )</th>
<th>( N = 100 )</th>
<th>( N = 200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>success rate</td>
<td>34.2%</td>
<td>61.1%</td>
<td>62.2%</td>
</tr>
<tr>
<td></td>
<td>( \frac{1}{d} \mathbb{E} [|v_f(T) - x^*|^2] )</td>
<td>3.12e-1</td>
<td>2.47e-1</td>
<td>2.42e-1</td>
</tr>
<tr>
<td>1</td>
<td>success rate</td>
<td>34.5%</td>
<td>57.1%</td>
<td>61.6%</td>
</tr>
<tr>
<td></td>
<td>( \frac{1}{d} \mathbb{E} [|v_f(T) - x^*|^2] )</td>
<td>3.09e-1</td>
<td>2.52e-1</td>
<td>0.244e-1</td>
</tr>
<tr>
<td>2</td>
<td>success rate</td>
<td>35.5%</td>
<td>54.8%</td>
<td>62.4%</td>
</tr>
<tr>
<td></td>
<td>( \frac{1}{d} \mathbb{E} [|v_f(T) - x^*|^2] )</td>
<td>3.06e-1</td>
<td>2.51e-1</td>
<td>2.44e-1</td>
</tr>
</tbody>
</table>

[Pinna-Totzeck-Tse-Martin, 17]
Learning MNIST data with two layer Neural Network

\[ X \in \mathbb{R}^{7290} \]

Only using \( N = 100, M = 10 \)
How parameters affect the performance

![Graph showing the effect of parameters on performance](image)
Future Directions

• Ongoing work: Constrained optimization problem
• How to choose all the parameters?
• Theory for the numerical method.
Thanks!