Preconditioned Conjugate Gradient Methods in Truncated Newton Frameworks for Large-scale Linear Classification

Wei-Lin Chiang

Department of Computer Science

National Taiwan University

Joint work with Chih-Yang Hsia and Chih-Jen Lin

Outline

1. Introduction & Proposed Methods
2. Experiments
3. Discussions & Conclusions

Linear Classification & Its Optimization

- Linear classification is important for many applications, but training large data may still be time-consuming
- Training data $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{l}, y_i \in \{-1, 1\}, \boldsymbol{x}_i \in \mathbb{R}^n$;
  l: #instances, n: #features
- We solve

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}), \text{where } f(\boldsymbol{w}) \equiv \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l} \xi(y_i \boldsymbol{w}^T \boldsymbol{x}_i)$$

C: regularization parameter

ξ: loss functions such as logistic loss

Newton Method for Linear Classification

- We consider Newton method for large-scale linear classification (Lin et al., 2008)
- In each iteration, Newton method considers the quadratic approximation at iterate $\boldsymbol{w}$ to find a direction $\boldsymbol{s}$ by solving a sub-problem

$$\min_{\boldsymbol{s}} \quad \frac{1}{2}\boldsymbol{s}^T \underbrace{\nabla^2 f(\boldsymbol{w})}_{\text{Hessian}} \boldsymbol{s} + \underbrace{\nabla f(\boldsymbol{w})^T}_{\text{Gradient}} \boldsymbol{s}, \tag{1}$$

- To solve the sub-problem, we take the derivative and solve a linear system

$$\nabla^2 f(\boldsymbol{w})\boldsymbol{s} = -\nabla f(\boldsymbol{w}) \tag{2}$$

Hessian-free Newton Method

- However, $\nabla^2 f(\boldsymbol{w})$ is often too large to be stored

$$\nabla^2 f(\boldsymbol{w}) \in \mathbb{R}^{n \times n}, n: \text{number of features}$$

- Without $\nabla^2 f(\boldsymbol{w})$, how to solve the linear system?

Hessian-free Newton Method

- However, $\nabla^2 f(\boldsymbol{w})$ is often too large to be stored

$$\nabla^2 f(\boldsymbol{w}) \in \mathbb{R}^{n \times n}, n: \text{number of features}$$

- Without $\nabla^2 f(\boldsymbol{w})$, how to solve the linear system?
- In linear classification, $\nabla^2 f(\boldsymbol{w})$ has a special structure

$$\nabla^2 f(\boldsymbol{w}) = I + CX^T DX$$

  where D is a diagonal matrix and $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l]^T$ is the data matrix
- Hessian-vector product can be calculated by

$$\nabla^2 f(\boldsymbol{w})\boldsymbol{s} = (I + CX^T DX)\boldsymbol{s} = \boldsymbol{s} + CX^T(D(X\boldsymbol{s}))$$

Hessian-free Newton Method (Cont'd)

- Iterative methods such as conjugate gradient (CG) can be used to solve each Newton linear system. CG involves a series of Hessian-vector products

$$\underbrace{\nabla^2 f(\boldsymbol{w})\boldsymbol{s}_1,\ \nabla^2 f(\boldsymbol{w})\boldsymbol{s}_2,\ldots}_{\#\text{CG steps}} \rightarrow \text{solution } \boldsymbol{s} \text{ is obtained}$$

- The cost of Newton method becomes proportional to

$$\begin{aligned}&\#\text{CG steps of Newton iteration 1}\\ +&\#\text{CG steps of Newton iteration 2}\\ +&\ldots\end{aligned}$$

Hessian-free Newton Method (Cont'd)

- How many #CG steps are needed?
- When solving $A\boldsymbol{x} = \boldsymbol{b}$, a smaller condition number of A, $\texttt{cond}(A)$, usually leads to fewer #CG steps
- Preconditioning techniques (Concus et al., 1976) can possibly reduce the condition number of A

Preconditioned Conjugate Gradient (PCG)

Suppose we want to solve $A\boldsymbol{x} = \boldsymbol{b}$.

- PCG finds a preconditioner $M = EE^T$ to approximate A and transforms

$$A\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

  to a new linear system $\bar{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}$. Specifically,

$$\underbrace{(E^{-1}AE^{-T})}_{\bar{A}}\underbrace{(E^T\boldsymbol{x})}_{\bar{\boldsymbol{x}}} = \underbrace{E^{-1}\boldsymbol{b}}_{\bar{\boldsymbol{b}}}. \tag{4}$$

  Then PCG solves the transformed linear system
- If $M \approx A$, $\mathrm{cond}(E^{-1}AE^{-T}) \approx \mathrm{cond}(I) < \mathrm{cond}(A)$. Then fewer #CG steps are needed

Preconditioned Conjugate Gradient (PCG)

Suppose we want to solve $A\boldsymbol{x} = \boldsymbol{b}$.

- PCG finds a preconditioner $M = EE^T$ to approximate A and transforms

$$A\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

to a new linear system $\bar{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}$. Specifically,

$$\underbrace{(E^{-1}AE^{-T})}_{\bar{A}}\underbrace{(E^T\boldsymbol{x})}_{\bar{\boldsymbol{x}}} = \underbrace{E^{-1}\boldsymbol{b}}_{\bar{\boldsymbol{b}}}. \tag{4}$$

Then PCG solves the transformed linear system

- If $M \approx A$, $\text{cond}(E^{-1}AE^{-T}) \approx \text{cond}(I) < \text{cond}(A)$. Then fewer #CG steps are needed

Preconditioned Conjugate Gradient (PCG)

Suppose we want to solve $A\boldsymbol{x} = \boldsymbol{b}$.

- PCG finds a preconditioner $M = EE^T$ to approximate A and transforms

$$A\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

  to a new linear system $\bar{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}$. Specifically,

$$\underbrace{(E^{-1}AE^{-T})}_{\bar{A}}\underbrace{(E^{T}\boldsymbol{x})}_{\bar{\boldsymbol{x}}} = \underbrace{E^{-1}\boldsymbol{b}}_{\bar{\boldsymbol{b}}}. \tag{4}$$

  Then PCG solves the transformed linear system
- If $M \approx A$, $\text{cond}(E^{-1}AE^{-T}) \approx \text{cond}(I) < \text{cond}(A)$. Then fewer #CG steps are needed

Challenges of PCG for Solving One Linear System

Finding a good preconditioner is not easy

1. Preconditioning sometimes reduces #CG steps, but not always
2. Applying preconditioning incurs extra costs. Fewer #CG steps may not imply less running time

New Challenges of PCG in Newton

1. Newton method solves a series of linear systems depending on current $\boldsymbol{w}_k$

$$\text{Newton iteration 1: } \nabla^2 f(\boldsymbol{w}_1)\boldsymbol{s} = -\nabla f(\boldsymbol{w}_1)$$
$$\text{Newton iteration 2: } \nabla^2 f(\boldsymbol{w}_2)\boldsymbol{s} = -\nabla f(\boldsymbol{w}_2)$$
$$\vdots$$

   A preconditioner may be useful for some linear systems, but not for others

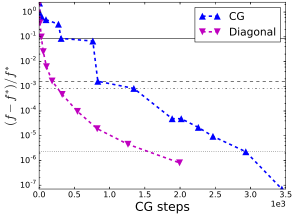
   Most past PCG studies focus on one linear system

2. We don't explicitly have $\nabla^2 f(\boldsymbol{w}_k)$. Many existing preconditioners can not be applied

Difficulties of Applying PCG in Newton

Let's try a diagonal preconditioner, which is doable in Hessian-free scenarios

kddb



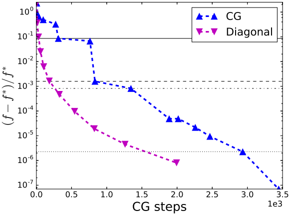
PCG better

- Preconditioning can be very useful

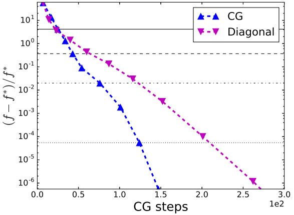
Difficulties of Applying PCG in Newton

Let's try a diagonal preconditioner, which is doable in Hessian-free scenarios

kddb



news20



PCG better

PCG worse

- Preconditioning can be very useful, but not always
- Can we improve the worse case?

Making Preconditioning More Robust

- Recall that in Newton, we solve a series of linear systems with the following matrices

	CG	PCG
Newton iteration 1	$\nabla^2 f(\boldsymbol{w}_1)$	$E_1^{-1}\nabla^2 f(\boldsymbol{w}_1)E_1^{-T}$
Newton iteration 2	$\nabla^2 f(\boldsymbol{w}_2)$	$E_2^{-1}\nabla^2 f(\boldsymbol{w}_2)E_2^{-T}$
$\vdots$	$\vdots$	$\vdots$

  A preconditioner may improve the conditions of some matrices but not others
- Can we have a setting achieving better robustness for the overall procedure?

Making Preconditioning More Robust

If a preconditioner $M = EE^T$ is not always useful, our idea is to choose the better one between CG and PCG

Making Preconditioning More Robust

If a preconditioner $M = EE^T$ is not always useful, our idea is to choose the better one between CG and PCG

- We hope to find a new $\bar{M} = \bar{E}\bar{E}^T$ satisfies

$$
\begin{aligned}
&\operatorname{cond}(\underbrace{\bar{E}^{-1}\nabla^2 f(\boldsymbol{w})\bar{E}^{-T}}_{\text{PCG with } \bar{M}}) \\
\approx\; &\min\{\operatorname{cond}(\underbrace{\nabla^2 f(\boldsymbol{w})}_{\text{CG}}), \operatorname{cond}(\underbrace{E^{-1}\nabla^2 f(\boldsymbol{w})E^{-T}}_{\text{PCG with } M})\}
\end{aligned}
\tag{5}
$$

Making Preconditioning More Robust

If a preconditioner $M = EE^T$ is not always useful, our idea is to choose the better one between CG and PCG

- We hope to find a new $\bar{M} = \bar{E}\bar{E}^T$ satisfies

$$
\begin{aligned}
&\operatorname{cond}(\underbrace{\bar{E}^{-1}\nabla^2 f(\boldsymbol{w})\bar{E}^{-T}}_{\text{PCG with } \bar{M}}) \\
\approx &\min\{\operatorname{cond}(\underbrace{\nabla^2 f(\boldsymbol{w})}_{\text{CG}}), \operatorname{cond}(\underbrace{E^{-1}\nabla^2 f(\boldsymbol{w})E^{-T}}_{\text{PCG with } M})\}
\end{aligned}
\tag{5}
$$

- Proposed method 1: run CG and PCG in parallel and choose

$$
\bar{M} = \begin{cases} I, & \text{if CG uses fewer steps} \\ M, & \text{if PCG uses fewer steps} \end{cases}
$$

Making Preconditioning More Robust

- Parallelizaion may not be always possible. Then choosing the better between CG and PCG is difficult
- Therefore, we set a more modest goal

$$\begin{aligned} &\mathrm{cond}(\bar{E}^{-1}\nabla^2 f(\boldsymbol{w})\bar{E}^{-T}) \\ <&\max\{\mathrm{cond}(\nabla^2 f(\boldsymbol{w})), \mathrm{cond}(E^{-1}\nabla^2 f(\boldsymbol{w})E^{-T})\} \end{aligned} \tag{6}$$

That is, we hope to avoid the worse one

Making Preconditioning More Robust

- Parallelizaion may not be always possible. Then choosing the better between CG and PCG is difficult
- Therefore, we set a more modest goal

$$\begin{aligned} &\text{cond}(\bar{E}^{-1}\nabla^2 f(\boldsymbol{w})\bar{E}^{-T}) \\ <\ &\max\{\text{cond}(\nabla^2 f(\boldsymbol{w})), \text{cond}(E^{-1}\nabla^2 f(\boldsymbol{w})E^{-T})\} \end{aligned} \tag{6}$$

  That is, we hope to avoid the worse one
- Proposed method 2: a weighted average

$$\bar{M} = \alpha M + (1-\alpha)I, \text{ where } 0 < \alpha < 1$$

  We prove the new preconditioner $\bar{M}$ satisfies (6)

Outline

1 Introduction & Proposed Methods

2 Experiments

3 Discussions & Conclusions

Experiment Settings

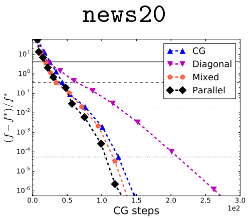
The following methods are considered

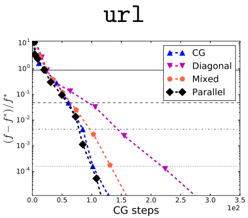
- `CG`: the standard CG without preconditioning
- `Diag`: the diagonal preconditioner

Proposed methods

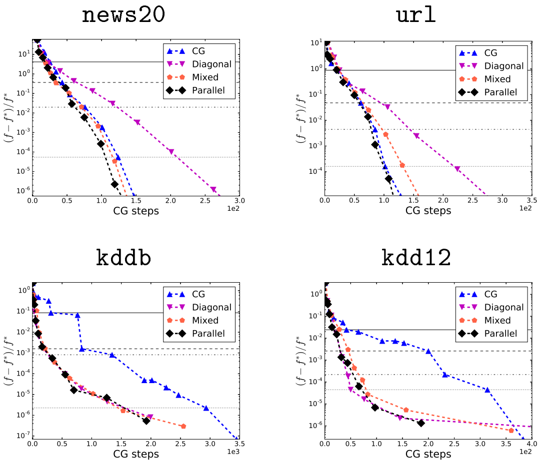
- `Parallel`: running CG and `Diag` in parallel
- `Mixed`: $\bar{M} = \alpha M + (1-\alpha)I$, where $\alpha = 0.01$ and M is `Diag`

More comparisons are in the paper





worse cases of `Diag` improved



worse cases of Diag improved

better cases of Diag remained

- The robustness is effectively improved
- The behavior of `Mixed` is not very sensitive to α

Outline

1 Introduction & Proposed Methods

2 Experiments

3 Discussions & Conclusions

Discussions & Conclusions

- Applying preconditioners on a sequence of linear systems in Hessian-free Newton is difficult
- We propose methods to improve the robustness
- The implementation is included in a linear classification package LIBLINEAR.[1] Many users are benefiting from this development

[1] https://www.csie.ntu.edu.tw/~cjlin/liblinear/