Why Line-Search When You Can Plane-Search?

ISMP 2024, Montreal

Betty Shea, Mark Schmidt
University of British Columbia

Getting more without spending (much) more

THE UNIVERSITY OF BRITISH COLUMBIA

We could be using better step sizes

For many common machine learning (ML) problems, these step size strategies all have the same asymptotic cost

- fixed step size,
- line search,
- plane search

Plane search PHB

Polyak's Heavy Ball Method (PHB)

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \alpha_t \nabla f(\mathbf{x_t}) + \beta_t(\mathbf{x_t} - \mathbf{x_{t-1}})$$

- Usually $\alpha_t = \alpha$ and $\beta_t = \beta$

Plane search PHB

Polyak's Heavy Ball Method (PHB)

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \alpha_t \nabla f(\mathbf{x_t}) + \beta_t(\mathbf{x_t} - \mathbf{x_{t-1}})$$

- Usually $\alpha_t = \alpha$ and $\beta_t = \beta$
- But you can do a plane search for the same cost

$$\alpha_t, \beta_t = \arg\min_{\alpha,\beta} f(\mathbf{x_t} - \alpha\nabla f(\mathbf{x_t}) + \beta(\mathbf{x_t} - \mathbf{x_{t-1}}))$$

Plane search PHB

Polyak's Heavy Ball Method (PHB)

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \alpha_t \nabla f(\mathbf{x_t}) + \beta_t(\mathbf{x_t} - \mathbf{x_{t-1}})$$

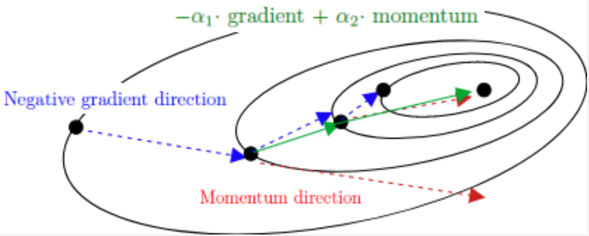
- Usually $\alpha_t = \alpha$ and $\beta_t = \beta$
- But you can do a plane search for the same cost

$$\alpha_t, \beta_t = \arg\min_{\alpha,\beta} f(\mathbf{x_t} - \alpha \nabla f(\mathbf{x_t}) + \beta(\mathbf{x_t} - \mathbf{x_{t-1}}))$$

- Potentially more progress at every iteration
- No hyperparameter tuning

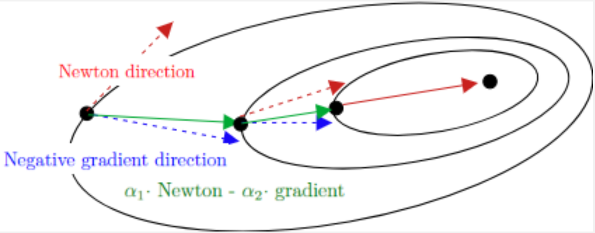
Plane search PHB

Adaptable gradient and momentum learning rates



Another use case: combination methods

Hybrid first- and second- order methods



Plane search for k directions

Subproblem to solve at every iteration t

Step sizes and search directions are

$$\boldsymbol{\alpha}_t = \begin{bmatrix} \alpha_{t,1} \\ \vdots \\ \alpha_{t,k} \end{bmatrix}, P_t = \begin{bmatrix} | & & | \\ p_{t,1} & \dots & p_{t,k} \\ | & & | \end{bmatrix}$$

Plane search for k directions

Subproblem to solve at every iteration t

Step sizes and search directions are

$$\boldsymbol{\alpha}_t = \begin{bmatrix} \alpha_{t,1} \\ \vdots \\ \alpha_{t,k} \end{bmatrix}, P_t = \begin{bmatrix} | & & | \\ p_{t,1} & \dots & p_{t,k} \\ | & & | \end{bmatrix}$$

The update becomes

$$\mathbf{x_{t+1}} = \mathbf{x_t} + P_t \boldsymbol{\alpha}_t$$

Plane search for k directions

Subproblem to solve at every iteration t

Step sizes and search directions are

$$\boldsymbol{\alpha}_t = \begin{bmatrix} \alpha_{t,1} \\ \vdots \\ \alpha_{t,k} \end{bmatrix}, P_t = \begin{bmatrix} | & & | \\ p_{t,1} & \dots & p_{t,k} \\ | & & | \end{bmatrix}$$

The update becomes

$$\mathbf{x_{t+1}} = \mathbf{x_t} + P_t \boldsymbol{\alpha}_t$$

Plane search solves this problem

$$\boldsymbol{\alpha}_t \in \arg\min_{\alpha} f\left(\mathbf{x_t} + P_t \boldsymbol{\alpha}_t\right)$$

When is plane search efficient?

Matrix-vector multiplications are bottlenecks

Linear composition problems (LCPs)

$$f(\mathbf{x}) = g(A\mathbf{x})$$

Examples:

- least squares: $g(A\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{y}\|^2$
- logistic regression: $g(A\mathbf{x}) = \log\left(1 + \exp(-\mathbf{y} \circ A\mathbf{x})\right)$

*Narkiss and Zibulevsky (2005) SESOP

Linear Composition Problem (LCP)

Cost of calculating function value

$$f(\mathbf{x_{t+1}}) = g(A(\mathbf{x_t} + P_t\boldsymbol{\alpha}_t)) = g(\mathbf{v_t} + \tilde{P}_t\boldsymbol{\alpha}_t)$$

where

$$\mathbf{v_t} = A\mathbf{x_t} \in \mathbb{R}^m \text{ and } \tilde{P}_t = AP_t \in \mathbb{R}^{m\times k}$$

Matrix-vector multiplications:

1. $\mathbf{v_t} = A\mathbf{x_t}$
2. $\tilde{P}_{t,1} = A^\intercal \mathbf{p_{t,1}}$
3. ...
4. $\tilde{P}_{t,k} = A^\intercal \mathbf{p_{t,k}}$

k direction plane search on LCP

THE UNIVERSITY OF BRITISH COLUMBIA

Subproblem at iteration t

Store $\mathbf{v_t}$ and $\tilde{P}_t$. Plane search is

$$\boldsymbol{\alpha}_t \in \arg\min_{\alpha} f(\mathbf{x_{t+1}}) = g(\mathbf{v_t} - \tilde{P}_t \boldsymbol{\alpha}) \tag{1}$$

- evaluating (1) requires no matrix-vector multiplication
- does require k vector-scalar multiplications
- significant gain if A is big and k is small
- buy 1 α get all the other αs for free

Plane search PHB on LCP

Subproblem at iteration t

PHB is a special case of k-direction plane search

$$\alpha_t = \begin{bmatrix} \alpha_{t,1} \\ \alpha_{t,2} \end{bmatrix}, \; P_t = \begin{bmatrix} | & | \\ -\nabla f(x_t) & (x_t - x_{t-1}) \\ | & | \end{bmatrix}$$

$$\boldsymbol{\alpha}_t \in \arg\min_{\boldsymbol{\alpha}} f(\mathbf{x_{t+1}}) = g(\mathbf{v_t} + \alpha_{t,1}\tilde{P}_{t,1} + \alpha_{t,2}(\mathbf{v_t} - \mathbf{v_{t-1}})) \quad (2)$$

Plane search PHB on LCP

Matrix-vector multiplications

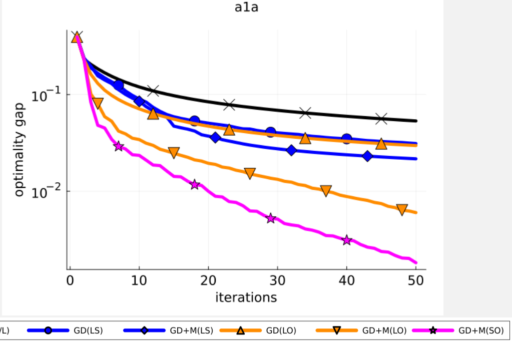
Matrix-vector multiplications

1. $\tilde{P}_{t,1} = -A^{\intercal}\nabla f(\mathbf{x_t})$.
2. $\mathbf{v_t} = A\mathbf{x_t}$. (Use $\mathbf{v_{t-1}}$ from previous iteration.)

Same as gradient descent with fixed step size and no momentum!

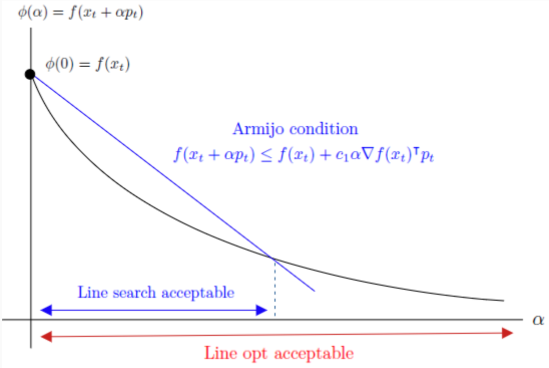
Experiments: GD+M, logistic regression

Fixed versus varying momentum weight



Line search blind spots

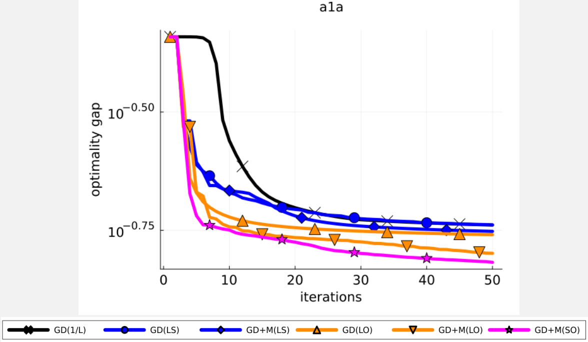
Armijo line search may rule out good step sizes



This affects Wolfe conditions too.

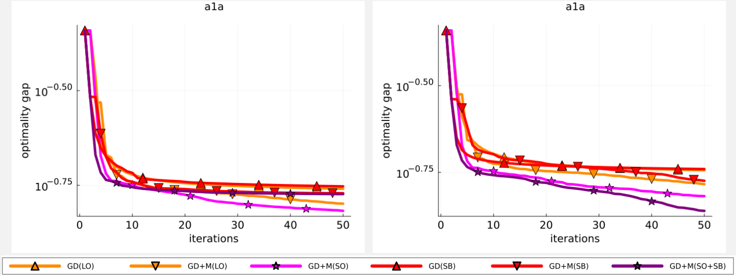
Experiments: GD+M, 2-layer NNs

No per-layer step sizes



Experiments: GD+M, 2-layer NNs

Per-layer step sizes without and with regularization



Experiments: observations

THE UNIVERSITY OF BRITISH COLUMBIA

Miscellaneous things we saw from the experiments

- large k (e.g. $k > 2$) does not appear to help much
- 3-term plane search implementation of Nesterov's acceleration
- can improve on Adam, quasi-Newton methods, etc.

Future directions

Software development: optimizers with built in plane search

- **Alyssa Zhang**,

  NSERC Undergraduate Student Research Award

Questions

- Does this work in the stochastic case?
- Does this work for modern architectures?

Thank you for listening

(or many) free step sizes

Experiments: GD+M, logistic regression

THE UNIVERSITY OF BRITISH COLUMBIA

Accelerated gradient methods