Newton's Method for Constrained Norm Minimization and Its Application to Weighted Graph Problems

Mahmoud El Chamie and Giovanni Neglia
INRIA - Sophia Antipolis
France

American Control Conference 2014 (Portland, OR, USA, June 4-6)

Acknowledgement

Konstantin Avrachenkov
k.avrachenkov@inria.fr

Giovanni Neglia
giovanni.neglia@inria.fr

Talk Outline

- The General Optimization Framework
- Contribution of the Paper
- Newton's Method for the General Framework
- Newton's Method for Weighted Graph Optimization (a case study)
- Performance Evaluation

The General Optimization Framework

- $$\mathrm{Argmin}_X \, f(X) = \underbrace{\mathrm{E}(X, \hat{y})}_{\text{Loss Function}} + \gamma \underbrace{\mathrm{R}(X)}_{\text{Regularization}}$$
  - $X \in \Re^{n_1, n_2}$
  - γ : scalar for the trade-off between the two terms

- For underdetermined systems (more variables than the equations): the *minimal norm interpolation problem [A. Argyriou et al., 2010]*,

$$\mathrm{Argmin}_X \;\; \mathrm{R}(X)$$
$$\text{subject to} \;\; \mathrm{L}(X) = \hat{y}$$

  - $\mathrm{L}(X) = A\mathrm{vext}(X)$ is a linear function (A has $r \times n_1 n_2$ dimensions , r rank of A)

Schatten-p Norm

$$\mathrm{R}(X) = \| X \|_{\sigma p} = \left(\sum_i \sigma_i^p\right)^{1/p}$$

- σ_i is the i-th singular value of X
- $p=1$ is the Nuclear norm
- $p=2$ is the Frobeniuos norm
- $p=\infty$ is the Spectral norm

- Differentiable for even $p=2q$, $\| X \|_{\sigma p}^p = \mathrm{Tr}((XX^T)^q)$
- The problem is then convex

Contribution

$$\underset{X}{\text{minimize}} \;\; h(X) = \mathrm{Tr}((XX^T)^q)$$
$$\text{subject to} \;\; A\,\mathrm{vect}(X) = \hat{y}$$

- Transform into unconstrained optimization
- Find close form solutions for the gradient and the Hessian
- Apply Newton's method
- Show that the optimization as applications in graph optimization problems
  - Multi-agents consensus problems
  - Hessian can be sparse and exact step-size can be easily calculated
- Study the convergence time of different optimization methods for this problem

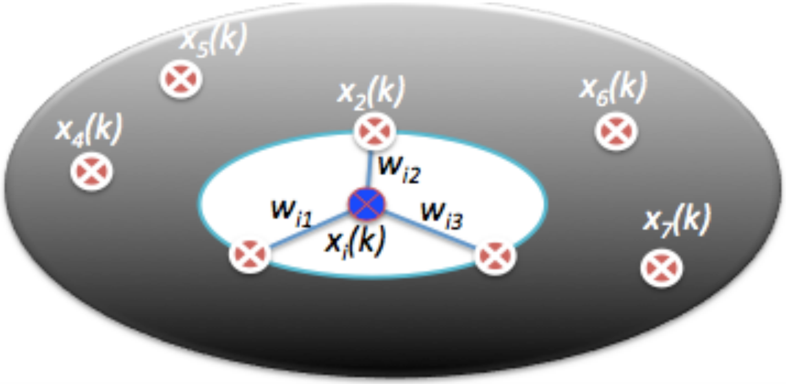
Weighted Graph Optimization: Weight Selection in Consensus Protocols

- Consensus algorithm
  - Given a network $G=(V, E)$
  - Each node has a local variable $x_i(k)$, where k is the iteration number and $x_i(0)$ are initial values
  - Each node performs weighted sum of its value and its neighbors' values:

$$x_i(k+1) = w_{ii} \times x_i(k) + \sum_{\text{neighbors } j} w_{ij} \times x_j(k)$$

  - In matrix form:

$$x(k) = W^k x(0)$$



Weighted Graph Optimization: Weight Selection in Consensus Protocols

- Consensus algorithm

$$x_i(k+1) = w_{ii} \times x_i(k) + \sum_{\text{neighbors } j} w_{ij} \times x_j(k)$$

  - For some (easy to satisfy) condition on the weights, the algorithm is guaranteed to converge to the average of initial values
  - An approximation to the best weights (that achieve fastest convergence independent of initial values) is

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & Tr(W^p) \\ \text{subject to} \quad & W = W^T,\ W\mathbf{1}_n = \mathbf{1}_n,\ W \in \mathcal{C}_G, \end{aligned}$$

  - p identifies the error from optimal weights [M. El Chamie *et al.*, 2012]

Weighted Graph Optimization: Weight Selection in Consensus Protocols

- Consensus algorithm

$$x_i(k+1) = w_{ii} \times x_i(k) + \sum_{\text{neighbors } j} w_{ij} \times x_j(k)$$

  - For some (easy to satisfy) condition on the weights, the algorithm is guaranteed to converge to the average of initial values
  - An approximation to the best weights (that achieve fastest convergence independent of initial values) is

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & Tr(W^p) \\ \text{subject to} \quad & W = W^T,\ W\mathbf{1}_n = \mathbf{1}_n,\ W \in \mathcal{C}_G, \end{aligned}$$

Schatten p-norm of W

Linear constraints

  - p identifies the error from optimal weights [M. El Chamie *et al.*, 2012]

Newton's Method

- Form the unconstrained problem:

$$f(\mathbf{w}) = Tr(W^p)|_{W=I_n - Q\text{diag}(\mathbf{w})Q^T}$$

  - w : m by 1 weight vector (each link is given a variable)
  - Q : n by m incidence matrix of the graph G
- Form the gradient $\mathbf{g} = \nabla_w f \in \Re^m$, $l \leftrightarrow (a,b)$:

$$(\nabla_w f)_l = \frac{\partial f}{\partial w_l} = p((W^{p-1})_{a,b} + (W^{p-1})_{b,a} - (W^{p-1})_{a,a} - (W^{p-1})_{b,b})$$

Newton's Method

- Form the unconstrained problem:

$$f(\mathbf{w}) = Tr(W^p)|_{W=I_n - Q\mathrm{diag}(\mathbf{w})Q^T}$$

  - w : m by 1 weight vector (each link is given a variable)
  - Q : n by m incidence matrix of the graph G
- Form the Hessian $H = \nabla_w^2 f \in \Re^{m\times m}$:

$$(\nabla_{\mathbf{w}}^2 f)_{l,k} = p \sum_{z=0}^{p-2} \psi(z)\psi(p-2-z),$$

$$\psi(z) = (W^z)_{a,c} + (W^z)_{b,d} - (W^z)_{a,d} - (W^z)_{b,c}.$$

Newton's Direction

- We calculated the gradient g, and the Hessian H
  - Notice that H is positive semi-definite since f is convex
- Newton's Direction Δw is simply the solution of :

$$H\Delta\mathbf{w} = \mathbf{g}$$

Line Search

- For choosing a stepsize that guarantees sufficient decrease in the function

$$\phi(t) = f(\mathbf{w} - t\Delta\mathbf{w}) \quad , \quad t > 0$$

- Exact line search is usually complex and other simple (non-optimal) choice are usually used
  - Pure Newton $t=1$ (for all iterations)
  - Backtracking line search starts from $t=1$, and multiplicatively decrease t till sufficient decrease in the function
- However, the special structure of our problem allows for a simple exact line search

Exact Line Search

- For choosing a stepsize that guarantees sufficient decrease in the function

$$\phi(t) = f(\mathbf{w} - t\Delta\mathbf{w}) = h(W + tU)$$

where $U = Q\,\mathrm{diag}(\Delta \mathrm{w})\mathrm{Q}^T$ and let $Y = W + tU$

- Since h is convex, then $\phi(t)$ is also convex, and the first and second derivatives:

$$\phi'(t) = p\,\mathrm{Tr}(Y^{p-1}U) \quad \text{and} \quad \phi''(t) = p\,\mathrm{Tr}(\sum_r Y^{p-2-r}UY^rU)$$

- With Newton- Raphson (exact stepsize)

$$t_n \leftarrow t_{n-1} - \frac{\phi'(t_{n-1})}{\phi''(t_{n-1})}$$

Summary for Newton's Method

- Step 0: Initial start $W^{(0)}=I_n$, precision ε , $k=0$
- Step 1: Calculate gradient $\boldsymbol{g} = \nabla_{\boldsymbol{w}} f \in \Re^{m}$
- Step 2: Calculate Hessian $H = (\nabla^2_{\boldsymbol{w}} f + \gamma I_m) \in \Re^{m \times m}$
- Step 3: Calculate Newton's direction $\Delta \boldsymbol{w}^{(k)} = H^{-1} \boldsymbol{g}$

  Stopping Condition: $\| \Delta \boldsymbol{w}^{(k)} \| < \varepsilon$
- Step 4: Use exact line search for stepsize $t^{(k)}$
- Step 5: Update the weight matrix

$$W^{(k+1)} = W^{(k)} + t^{(k)} Q \,\mathrm{diag}(\Delta \boldsymbol{w}^{(k)}) Q^T$$

- Step 6: k=k+1 and go back to Step 1

Closed form solution for p=2

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & h(W) = Tr(W^2) = \sum_{i,j} w_{ij}^2 \\ \text{subject to} \quad & W = W^T, W\mathbf{1}_n = \mathbf{1}_n, W \in \mathcal{C}_G. \end{aligned}$$

- Since objective function is quadratic, pure newton converges in one iteration, starting from any feasible initial value
- Let $W^{(0)}=I_n$, then $\boldsymbol{g} = -4 \times \mathbf{1}_m$ and $H = 2(2I_m + Q^T Q)$
- Substitute in equation

$$\begin{aligned} W^{(k+1)} &= W^{(k)} + t^{(k)} Q \operatorname{diag}(\Delta \boldsymbol{w}^{(k)}) Q^T \\ &= I_n - Q \operatorname{diag}((I_m + 0.5 Q^T Q)^{-1} \mathbf{1}_m) Q^T \end{aligned}$$

Closed form solution for p=2

$$\underset{W}{\text{minimize}} \quad h(W) = Tr(W^2) = \sum_{i,j} w_{ij}^2$$

$$\text{subject to} \quad W = W^T, W\mathbf{1}_n = \mathbf{1}_n, W \in \mathcal{C}_G.$$

- Since objective function is quadratic, pure newton converges in one iteration, starting from any feasible initial value
- Let $W^{(0)}=I_n$, then $\boldsymbol{g} = -4 \times \mathbf{1}_m$ and $H = 2(2I_m + Q^T Q)$
- Substitute in equation

$$\begin{aligned} W^{(k+1)} &= W^{(k)} + t^{(k)} Q \operatorname{diag}(\Delta \boldsymbol{w}^{(k)}) Q^T \\ &= \boxed{I_n - Q \operatorname{diag}((I_m + 0.5 Q^T Q)^{-1} \mathbf{1}_m) Q^T} \end{aligned}$$

Closed Form

Simulation

T_{conv} (number of iterations)	$ER(n=100, Pr=0.07)$ $p=2$	$p=4$	$p=6$	$p=10$
Exact-Newton	1	5	5.7	6.1
Pure-Newton	1	9	11.1	13.9
Exact-GD	72.3	230.5	482.7	1500.5
Exact-Nesterov	130.2	422.8	811.3	1971.2
BT-GD or BT-Nesterov	> 5000	> 5000	> 5000	> 5000

TABLE I

CONVERGENCE TIME USING DIFFERENT OPTIMIZATION METHODS FOR PROBLEM (12).

- GD: Decent Gradient
- BT- : Backtracking line search
- Stopping Condition $\|g\| < 10^{-10}$

Simulation

T_{conv} (number of iterations)	$ER(n = 100, Pr = 0.07)$			
	$p = 2$	$p = 4$	$p = 6$	$p = 10$
Exact-Newton	1	5	5.7	6.1
Pure-Newton	1	9	11.1	13.9
Exact-GD	72.3	230.5	482.7	1500.5
Exact-Nesterov	130.2	422.8	811.3	1971.2
BT-GD or BT-Nesterov	> 5000	> 5000	> 5000	> 5000

TABLE I

CONVERGENCE TIME USING DIFFERENT OPTIMIZATION METHODS FOR PROBLEM (12).

- GD: Decent Gradient
- BT- : Backtracking line search
- Stopping Condition $||g|| < 10^{-10}$

- Observation 1 (intuitive)

On average, the number of iterations to converge of Newton much less than first order methods

Simulation

T_{conv} (number of iterations)	$ER(n = 100, Pr = 0.07)$ $p = 2$	$p = 4$	$p = 6$	$p = 10$
Exact-Newton	1	5	5.7	6.1
Pure-Newton	1	9	11.1	13.9
Exact-GD	72.3	230.5	482.7	1500.5
Exact-Nesterov	130.2	422.8	811.3	1971.2
BT-GD or BT-Nesterov	> 5000	> 5000	> 5000	> 5000

TABLE I

CONVERGENCE TIME USING DIFFERENT OPTIMIZATION METHODS FOR PROBLEM (12).

- GD: Decent Gradient
- BT- : Backtracking line search
- Stopping Condition $||g|| < 10^{-10}$

- Observation 2 (less intuitive)

Newton's method is less sensitive to stepsize (exact stepsize does not change much convergence)
In gradient methods, highly sensitive to stepsize

Simulation

T_{conv} (number of iterations)	$ER(n = 100, Pr = 0.07)$ $p = 2$	$p = 4$	$p = 6$	$p = 10$
Exact-Newton	1	5	5.7	6.1
Pure-Newton	1	9	11.1	13.9
Exact-GD	72.3	230.5	482.7	1500.5
Exact-Nesterov	130.2	422.8	811.3	1971.2
BT-GD or BT-Nesterov	> 5000	> 5000	> 5000	> 5000

TABLE I

CONVERGENCE TIME USING DIFFERENT OPTIMIZATION METHODS FOR PROBLEM (12).

- GD: Decent Gradient
- BT- : Backtracking line search
- Stopping Condition $||g||<10^{-10}$

- Observation 3 (not intuitive)

Decent Gradient method is faster than Nesterov

Conclusion

- Newton's method for Schatten p-norm minimization
- Its application to weighted graph problems
  - Sparse Hessian
  - Fast convergence
  - Robustness to stepsize
- Future work
  - General values of p (not just for even values)
  - Other graph optimization

Conclusion

- Newton's method for Schatten p-norm minimization
- Its application to weighted graph problems
  - Sparse Hessian
  - Fast convergence
  - Robustness to stepsize
- Future work
  - General values of p (not just for even values)
  - Other graph optimization

Thank you!

Questions?

Possible Approaches

$$\mathrm{Argmin}_X \ \ \mathrm{R}(X)$$
$$\text{subject to} \ \ \mathrm{L}(X) = \hat{y}$$

- First Order Methods
  - Gradient Method
  - Fast Gradient Method (Nestrov)
  - Drawbacks
    - Slow convergence
    - Step size selection (for unbounded gradient)
- Second Order Methods (for differentiable $\mathrm{R}(.)$)
  - Newton's Method
  - Drawbacks
    - Difficult to have closed form solutions
    - Complexity (forming and inverting the Hessian)

Newton's Method for Schatten-p Norm

$$\begin{array}{ll} \underset{X}{\text{minimize}} & h(X) = Tr\left(\left(XX^T\right)^q\right) \\ \text{subject to} & \left[\begin{array}{cc} I_r & B \end{array}\right] P \ \text{vect}(X) = \hat{\mathbf{y}} \end{array}$$

1. Substitute the constraints in the objective function,

$$\mathbf{x} = \left[\begin{array}{cc} 0^{n_1 n_2 - r, r} & I_{n_1 n_2 - r} \end{array}\right] P \ \text{vect}(X), \quad X = \text{vect}^{-1}\left(P^{-1}\left[\begin{array}{c} \hat{\mathbf{y}} - B\mathbf{x} \\ \mathbf{x} \end{array}\right]\right),$$

2. Reformulate into an unconstrained problem:

$$\underset{\mathbf{x}}{\text{minimize}} \, f(\mathbf{x}), \qquad f(\mathbf{x}) = Tr\left(\left(XX^T\right)^q\right)$$

3. Solve for the gradient $\boldsymbol{g} = \nabla_x f$ and Hessian $H = \nabla_x^2 f$ (closed form formulas are given in the paper, we only give the results here for a case study)
4. Find the newton's direction, iteratively update variable using stepsize (details in the upcoming case study)

D-regular graphs

- D-regular graphs are graphs where all vertices have the same degree D (cycles, complete graphs, …)
- Optimal values for p=2 is

$$w_l = \frac{1}{1+D} \ \forall l = 1, \ldots, m.$$

- Which gives the same weights as well known weight selection heuristics as Metropolis weight selection or the maximum degree.