Proximal Newton DC programming for non-convex problems

Gilles GASSO

Joint work with A. Rakotomamonjy, R. Flamary and S. Canu

1 Day France/Japan Meeting

September 25, 2017

litis

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Setting

General machine learning problem

- Dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{N}$
- Learn a functional relation $f : \mathcal{X} \to \mathcal{Y}$

$$\min_{f \in \mathcal{C}} \quad \underbrace{L(f, \mathcal{S})}_{\text{fitting error}} \quad + \quad \underbrace{\lambda\, \Omega(f)}_{\text{regularization term}}$$

- $\mathcal{C} \subseteq \mathcal{H}$: space of functions

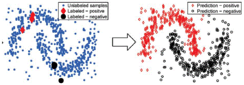
Common issues

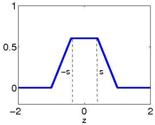
- Choice of the loss function L
- Specification of the regularization term Ω
- Optimization algorithm

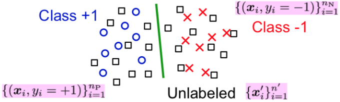
Why non-convex problems

Non-convex loss function L

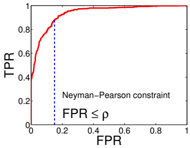
- Weakly supervised learning
  - Semi-supervised learning
  - PU classification and variants







- Probability constraint
  - Imbalanced classification
  - Neyman-Pearson constraint

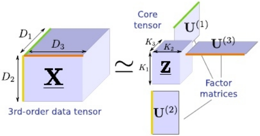


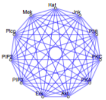
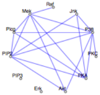
Why non-convex problems

Sparsity constraint on $f \longrightarrow$ non-convex regularization Ω

- High dimensional problems
  - Signal denoising
  - Compressive sensing
  - Bioinformatics ...


- Structure inference
  - Matrix/Tensor decomposition (low rank structure)
  - Graphical model inference (sparse graph structure) ...



Sparsity

Sparse Learning problem

- Desired model f depends on parameter vector $\mathbf{w} \in \mathbb{R}^d$
- Simple sparse learning problem

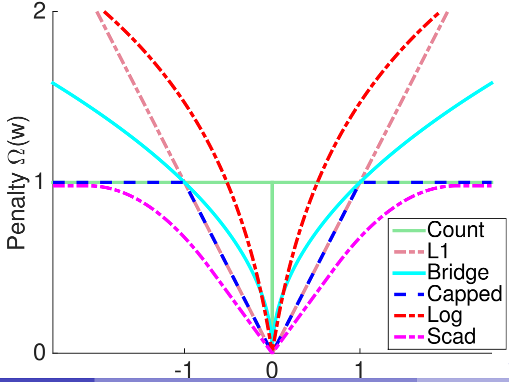
$$\min_w \quad L(\mathbf{w}) \quad + \quad \lambda \, \|\mathbf{w}\|_0$$

Counting norm

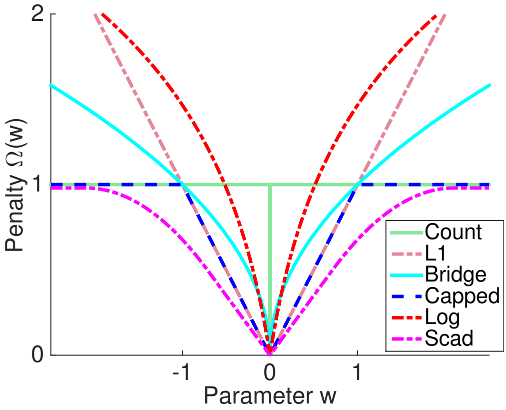
1. Count: $\Omega(\mathbf{w}) = \sum_{j=1}^{d} \mathbb{I}_{\mathbf{w}_j \neq 0}$
2. Number of non-zeros components of $\mathbf{w}$

Usual relaxations of counting norm

1. Convex ℓ_1-norm: $\Omega(\mathbf{w}) = \sum_{j=1}^{d} |w_j|$
2. Bridge [Frank and Friedman, 1993]: $\Omega(\mathbf{w}) = \sum_{j=1}^{d} |w_j|^p$, $p \in (0,1)$
3. Log [Candes et al., 2008]: $\Omega(\mathbf{w}) = \sum_{j=1}^{d} \log(|w_j|^p + \epsilon)$,
4. Capped ℓ_1 [Zhang, 2008]: $\Omega(\mathbf{w}) = \sum_{j=1}^{d} \min(\eta, |w_j|)$
5. SCAD [Fan and Li, 2001]



Usual relaxations of counting norm



Issues

- Non-convex relaxations promote better sparsity. . .
- but their optimization is more challenging

Optimization approaches

- Coordinate wise optimization [Mazumder et al., 2011, Breheny and Huang, 2011]
- Active set methods [Jiao et al., 2013]
- Regularization path (SCAD and MCP) [Breheny and Huang, 2011]
- DC algorithm
- Proximal methods

Difference of convex approach

Recall general problem

Learning problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} J(\mathbf{w}) \quad \text{with} \quad J(\mathbf{w}) = L(\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

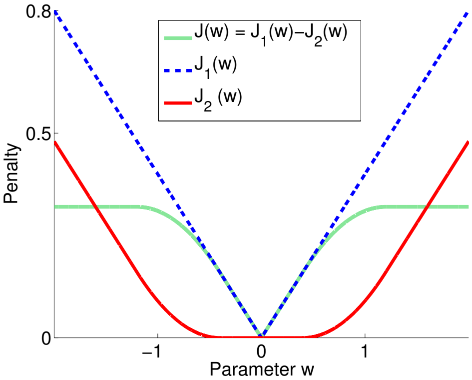
Difference of Convex (DC) Approach

- Dates to early 90's [Tao et al., 1988, Tao and Le Thi Hoai, 1994]
- Many further improvements (theory and algorithm) and applications
- Requires $J(\mathbf{w})$ to be a Difference of Convex functions

Difference of Convex functions

DC function

- Let $J_1(\mathbf{w}), J_2(\mathbf{w}) : \mathcal{C} \rightarrow]-\infty, +\infty]$ two convex, proper and lower semi-continuous functions
- $J(\mathbf{w})$ is a DC function if it can be expressed as $J(\mathbf{w}) = J_1(\mathbf{w}) - J_2(\mathbf{w})$.

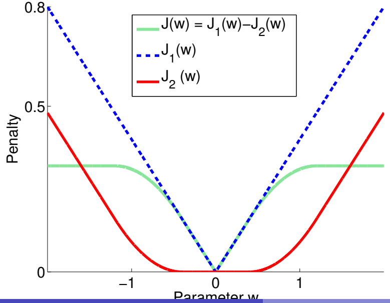


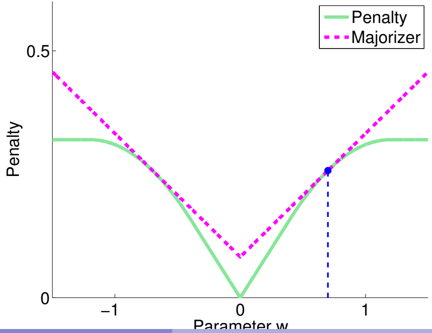
Properties of DC functions

Convex majorization

- Let $\partial J_2(\mathbf{w}_t) = \left\{\boldsymbol{\alpha}_t \in \mathbb{R}^d, J_2(\mathbf{w}) \geq J_2(\mathbf{w}_t) + \langle \mathbf{w} - \mathbf{w}_t, \boldsymbol{\alpha}_t\rangle, \forall \mathbf{w} \in \mathbb{R}^d\right\}$ the subdifferential of J_2 at $\mathbf{w}_t$.
- A convex majorization function of $J(\mathbf{w}) = J_1(\mathbf{w}) - J_2(\mathbf{w})$ at $\mathbf{w}_t$ is

$$J(\mathbf{w}) \leq J_1(\mathbf{w}) - J_2(\mathbf{w}_t) - \langle \mathbf{w} - \mathbf{w}_t, \boldsymbol{\alpha}_t\rangle$$





DC Algorithm

Principle: successive convex relaxations

- At each iteration t, define the convex majorization function

$$J_{cvx}(\mathbf{w}) = J_1(\mathbf{w}) - J_2(\mathbf{w}_t) - \langle \mathbf{w} - \mathbf{w}_t, \boldsymbol{\alpha}_t \rangle \quad \text{with} \quad \boldsymbol{\alpha}_t \in \partial J_2(\mathbf{w}_t)$$

- Next solution: $\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}} J_{cvx}(\mathbf{w})$

Algorithm for solving $\min_{\mathbf{w}} J_1(\mathbf{w}) - J_2(\mathbf{w})$

Set $t = 0$, initialize $\mathbf{w}_t \in \text{dom} J_1$
repeat
  Select $\boldsymbol{\alpha}_t \in \partial J_2(\mathbf{w}_t)$
  Define $J_{cvx}(\mathbf{w})$ and solve $\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}} J_{cvx}(\mathbf{w})$
  $t = t + 1$
until convergence

DC algorithm in play: sparse signal recovery

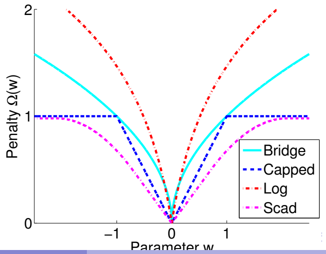
Optimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{y}-\mathbf{\Phi w}\|_2^2 + \lambda \sum_{j=1}^{d} \Omega(|w_j|)$$

- $\mathbf{y} \in \mathbb{R}^N$: noisy measurements, $\mathbf{\Phi} \in \mathbb{R}^{N\times d}$: given dictionary
- $\mathbf{w} \in \mathbb{R}^d$: sparse parameter vector

Non-convex penalties

1. Bridge: $\Omega(w_j) = |w_j|^p$, $p \in (0,1)$
2. Log: $\Omega(w_j) = \log(|w_j|^p + \epsilon)$,
3. Capped ℓ_1: $\Omega(w_j) = \min(\eta, |w_j|)$
4. SCAD



DC decomposition

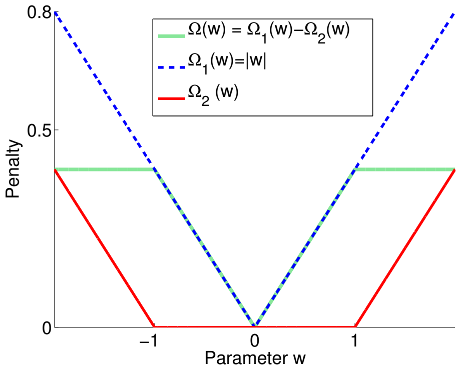
DC Decomposition of the penalty

- $\Omega(|w_j|) = \Omega_1(|w_j|) - \Omega_2(|w_j|)$
- $\Omega_1(|w_j|) = |w_j|$ and $\Omega_2(|w_j|) = |w_j| - \Omega(|w_j|)$

Example

For capped ℓ_1 penalty we have

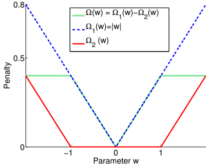
- $\Omega(|w_j|) = \min(\eta, |w_j|)$
- $\Omega_1(|w_j|) = |w_j|$
- $\Omega_2(|w_j|) = \max(0, |w_j| - \eta)$



DC decomposition

DC Decomposition of the penalty

- $\Omega(|w_j|) = |w_j| - \Omega_2(|w_j|)$
- $\Omega_2(|w_j|) = |w_j| - \Omega(|w_j|)$



DC decomposition of the objective function

- Using additivity property of DC
- $J_1(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi w}\|_2^2 + \lambda \sum_{j=1}^d |w_j|$ and $J_2(\mathbf{w}) = \lambda \sum_{j=1}^d \Omega_2(|w_j|)$

Convex majorization at $\mathbf{w} = \mathbf{w}_t$

- Majorization of $-J_2(\mathbf{w})$

  $-\lambda \sum_{j=1}^d \Omega_2(|w_j|) \leq -\lambda \sum_{j=1}^d \alpha_j^t |w_j| + \text{const}$ with $\alpha_j^t \in \partial\Omega_2(|w_j|)$
- Majorization of the objective function: $J_1(\mathbf{w}) - \lambda \sum_{j=1}^d \alpha_j^t |w_j| + \text{const}$

Iterative re-weighted lasso

Iterative re-weigthed Lasso algorithm

Set $t = 0$, initialize $\mathbf{w}_t$

repeat

Select $\alpha_j^t \in \partial\Omega_2(|w_j|)$ for $\mathbf{w} = \mathbf{w}_t$

Find $\mathbf{w}_{t+1} = \text{argmin}_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_{j=1}^{d}(\lambda - \alpha_j^t)|w_j|$

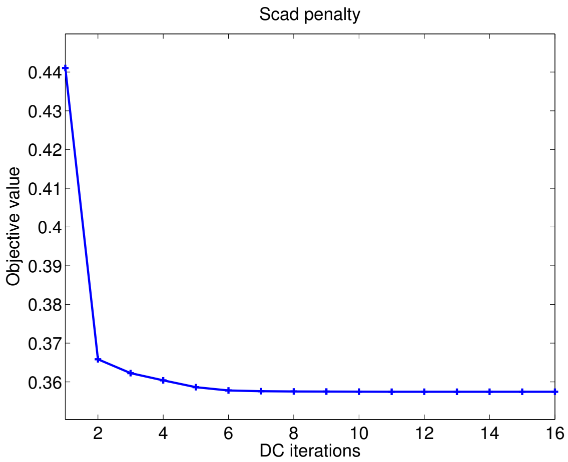
$t = t + 1$

until convergence

- Each iteration is a Lasso type problem
- Require any off-the-shelf Lasso solver

Empirical evaluation: convergence

- Typically few iterations for convergence in objective function

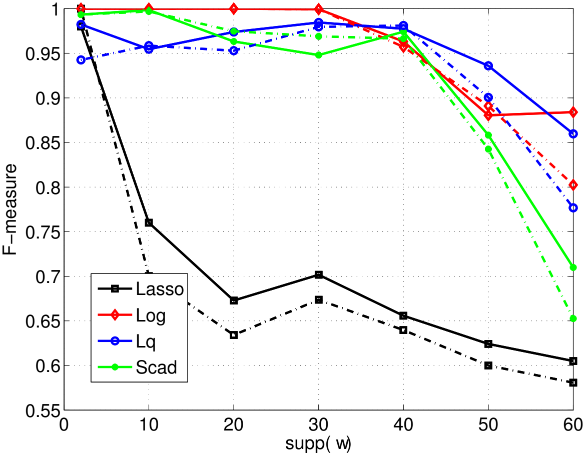


Performance measure

$$\text{Fmeasure} = 2\frac{|\text{supp}(\mathbf{w}^*) \cap \text{supp}(\hat{\mathbf{w}})|}{|\text{supp}(\mathbf{w}^*)| + |\text{supp}(\hat{\mathbf{w}})|}$$

- $\text{supp}(\mathbf{w}) = \{j, \mathbf{w}_j \neq 0\}$
- $\mathbf{w}^*$: true vector and $\hat{\mathbf{w}}$: estimated one
- Fmeasure close to 1 indicates a performing support recovery
- Comparison of Lasso with non-convex penalties

Performance

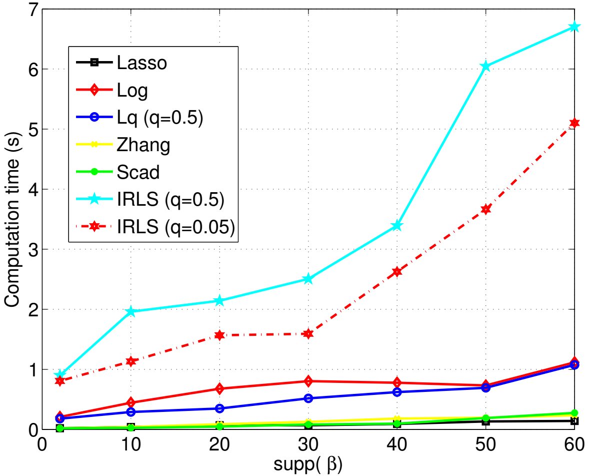


Dotted lines: highly correlated atoms, Solid lines: weak dependence of atoms

Non-convex penalties are effective than Lasso, especially log penalty

Computation time

- DC algorithm appears rather slow



DC proximal Newton

Proximal approach

General problem

$$\min_{\mathbf{w}} J(\mathbf{w}) := L(\mathbf{w}) + \Omega(\mathbf{w})$$

Assumptions

- $L(\mathbf{w})$ is either convex or is a DC function $L(\mathbf{w}) = L_1(\mathbf{w}) - L_2(\mathbf{w})$, lower bounded and twice differentiable
- We require $L_1(\mathbf{w})$ to be gradient Lipschitz
- $\Omega(\mathbf{w}) = \Omega_1(\mathbf{w}) - \Omega_2(\mathbf{w})$ is a DC function with $\Omega_k(\mathbf{w})$ lower semi-continuous, proper convex function
- $\Omega(\mathbf{w})$ may not be smooth

Proximal approach

General problem

$$\min_{\mathbf{w}} J(\mathbf{w}) := L(\mathbf{w}) + \Omega(\mathbf{w})$$

Solving algorithms

- Apply DC procedure to $L_1(\mathbf{w}) + \Omega_1(\mathbf{w}) - (L_2(\mathbf{w}) + \Omega_2(\mathbf{w}))$

  Might be slow if the convex relaxation problem is not easy to handle

- Apply proximal method
  - Generate sequence $\{\mathbf{w}_{t+1} = \text{argmin}_w \tilde{J}(\mathbf{w}, \mathbf{w}_t)\}$
  - $\tilde{J}(\mathbf{w}, \mathbf{w}_t) = \tilde{L}(\mathbf{w}, \mathbf{w}_t) + \tilde{\Omega}(\mathbf{w}, \mathbf{w}_t)$: convex quadratic majorization of $J(\mathbf{w})$ at $\mathbf{w}_t$
  - Exploit Lipschitz gradient property and DC convex linearization

Quadratic convex majorization

$$\min_{\mathbf{w}} L(\mathbf{w}) + \Omega(\mathbf{w})$$

Quadratic approximation of L

- $L(\mathbf{w}) = L_1(\mathbf{w}) - L_2(\mathbf{w})$ twice differentiable and L_1 gradient Lipschitz
- Let $\mathbf{w} = \mathbf{w}_t + \Delta\mathbf{w}$

$$\begin{aligned}\tilde{L}(\mathbf{w}, \mathbf{w}_t) &= L_1(\mathbf{w}_t) + \nabla L_1(\mathbf{w}_t)^\top \Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}_t \Delta\mathbf{w} \\ &\quad - L_2(\mathbf{w}_t) - \nabla L_2(\mathbf{w}_t)^\top \Delta\mathbf{w}\end{aligned}$$

- $\mathbf{H}_t \succeq 0$: approximation of the Hessian of L_1

Linear approximation of $\Omega(\mathbf{w}) = \Omega_1(\mathbf{w}) - \Omega_2(\mathbf{w})$

$$\tilde{\Omega}(\mathbf{w}, \mathbf{w}_t) = \Omega_1(\mathbf{w}) - \Omega_2(\mathbf{w}_t) - \boldsymbol{\alpha}_t^\top \Delta\mathbf{w}, \quad \boldsymbol{\alpha}_t \in \partial\Omega_2(\mathbf{w}_t)$$

Quadratic convex majorization

Quadratic approximation of L

- Let $\mathbf{w} = \mathbf{w}_t + \Delta\mathbf{w}$

$$\begin{aligned}\tilde{L}(\mathbf{w}, \mathbf{w}_t) &= L_1(\mathbf{w}_t) + \nabla L_1(\mathbf{w}_t)^\top \Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}_t \Delta\mathbf{w} \\ &\quad - L_2(\mathbf{w}_t) - \nabla L_2(\mathbf{w}_t)^\top \Delta\mathbf{w}\end{aligned}$$

- $\mathbf{H}_t \succ 0$: approximation of the Hessian of L_1

Linear approximation of $\Omega(\mathbf{w}) = \Omega_1(\mathbf{w}) - \Omega_2(\mathbf{w})$

$$\tilde{\Omega}(\mathbf{w}, \mathbf{w}_t) = \Omega_1(\mathbf{w}) - \Omega_2(\mathbf{w}_t) - \boldsymbol{\alpha}_t^\top \Delta\mathbf{w}, \quad \boldsymbol{\alpha}_t \in \partial\Omega_2(\mathbf{w}_t)$$

Quadratic approximation of the objective function

$$\tilde{J}(\Delta\mathbf{w}) = \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}_t \Delta\mathbf{w} + \mathbf{v}_t^\top \Delta\mathbf{w} + \Omega_1(\mathbf{w}_t + \Delta\mathbf{w}) + \text{const}$$

with $\quad \mathbf{v}_t = \nabla L_1(\mathbf{w}_t) - \nabla\Omega_1(\mathbf{w}_t) - \boldsymbol{\alpha}_t$

Optimization scheme

General scheme

- At each iteration $\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma_t \Delta\mathbf{w}_t$ (γ_t is the step-size)
- Search direction: $\Delta\mathbf{w} = \text{argmin}_{\Delta\mathbf{w}} \tilde{J}(\Delta\mathbf{w})$

$$
\begin{aligned}
& \min_{\Delta\mathbf{w}} && \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}_t \Delta\mathbf{w} + \mathbf{v}_t^\top \Delta\mathbf{w} + \Omega_1(\mathbf{w}_t + \Delta\mathbf{w}) \\
\Leftrightarrow & \min_{\mathbf{z}} && \frac{1}{2}(\mathbf{z} - \mathbf{w}_t)^\top \mathbf{H}_t (\mathbf{z} - \mathbf{w}_t) + \mathbf{v}_t^\top (\mathbf{z} - \mathbf{w}_t) + \Omega_1(\mathbf{z}), \;\; \mathbf{z} = \mathbf{w}_t + \Delta\mathbf{w} \\
\Leftrightarrow & \min_{\mathbf{z}} && \frac{1}{2}\|(\mathbf{z} - \mathbf{w}_t) + \mathbf{H}_t^{-1}\mathbf{v}_t\|_{H_t}^2 + \Omega_1(\mathbf{z}) \quad \text{with} \quad \|\mathbf{z}\|_{\mathbf{H}}^2 = \mathbf{z}^\top \mathbf{H}\mathbf{z}
\end{aligned}
$$

Definition: Proximal Newton

$$
\textbf{prox}_{\Omega_1}^{\mathbf{H}}(\mathbf{w}) = \text{argmin}_{\mathbf{z}} \frac{1}{2}\|z - \mathbf{w}\|_{\mathbf{H}}^2 + \Omega_1(\mathbf{z})
$$

Search direction

$$
\Delta\mathbf{w} = \textbf{prox}_{\Omega_1}^{\mathbf{H}_t}(\mathbf{w}_t - \mathbf{H}_t^{-1}\mathbf{v}_t) - \mathbf{w}_t
$$

Algorithm

Non-convex second-order (Newton) Proximal algorithm

Set $t = 0$, initialize $\mathbf{w}_t$

repeat

Compute $\mathbf{v}_t = \nabla L_1(\mathbf{w}_t) - \nabla L_2(\mathbf{w}_t) - \boldsymbol{\alpha}_t$ with $\boldsymbol{\alpha}_t \in \partial\Omega_2(\mathbf{w}_t)$

Compute the Hessian $\mathbf{H}_t$

Solve for $\Delta\mathbf{w}_t = \mathbf{prox}_{\Omega_1}^{\mathbf{H}_t}(\mathbf{w}_t - \mathbf{H}_t^{-1}\mathbf{v}_t) - \mathbf{w}_t$

Compute the step-size γ_t by backtracking

$\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma_t\Delta\mathbf{w}_t$

Increase t

until convergence

Elements of convergence

Convergence guarantees

- Sufficient decrease of the objective function: for $\mathbf{H}_t \succ 0$ it holds

$$J(\mathbf{w}_{t+1}) - J(\mathbf{w}_t) \leq -\gamma_t \Delta\mathbf{w}_t^\top \mathbf{H}_t \Delta\mathbf{w}_t + O(\gamma_t^2)$$

- Existence of a step-size: for $\mathbf{H}_t \succ m\mathbf{I}$ and ζ the Lipschitz constant of ∇L_1 the decrease holds for

$$\gamma_t \leq \min\left(1, 2m\frac{1-\theta}{\zeta}\right), \quad \theta \in (0, 1/2)$$

- Convergence to a stationary point: if the previous conditions hold at each iteration t, any limit point of the sequence $\{\mathbf{w}_t\}$ is a stationary point of the optimization problem

Related method

General Iterative Shrinkage and Thresholding Algorithm (GIST) [Gong et al., 2013]

- First order proximal method
- Based on a non-convex majorization function

$$\tilde{F}(\mathbf{w}, \mathbf{w}_t) = L(\mathbf{w}_t) + \nabla L(\mathbf{w}_t)^\top \Delta\mathbf{w} + \frac{\gamma_t}{2} \Delta\mathbf{w}^\top \Delta\mathbf{w} + \Omega(\mathbf{w})$$

- $\mathbf{w}_{t+1} = \mathbf{prox}_\Omega \left(\mathbf{w}_t - \nabla L(\mathbf{w}_t)/\gamma_t\right)$ where
- $\mathbf{prox}_\Omega \left(\mathbf{w}\right) = \text{argmin}_{\mathbf{z}} \ \frac{1}{2}\|\mathbf{z} - \mathbf{w}\|_2^2 + \Omega(\mathbf{z})$ is a non-convex proximal
- Closed-form proximal solution exists for previously presented non-convex penalties

Applications

Classification problem

- Dataset: $\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^N$
- Loss function: $L(\mathbf{w}) = \sum_{i=1}^N \log(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w}))$ (convex function)
- Regularizer: $\Omega(\mathbf{w}) = \sum_{j=1}^d \min(\eta, |w_j|)$ (non-convex penalty)

		Class. Rate (%)			Time (s)		
dataset	d	DCA	GIST	DC-PN	DCA	GIST	DC-PN
la2	31472	91.32±0.9	91.67±0.9	91.81±0.9	36±11	45±26	**21±12**
sports	14870	97.86±0.4	97.94±0.3	97.94±0.3	89±70	161±162	**23±13**
classic	41681	96.93±0.6	97.33±0.5	97.38±0.5	**3.5±3.8**	310±11	17±7
ohscal	11465	87.05±0.6	87.99±0.6	**89.27±0.6**	320±134	44±21	**19±25**
real-sim	20958	95.16±0.3	**96.28±0.2**	96.05±0.2	63±96	382±813	**23±9**

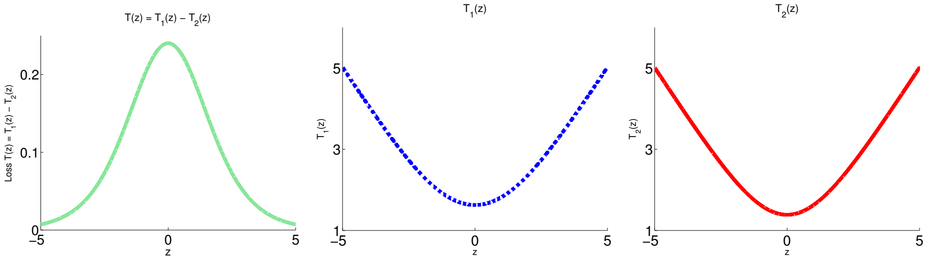
Proximal methods exploiting DC decomposition are faster than raw DC approach.
Proximal Newton is faster than the gradient counterpart.

Applications

Semi-supervised classification problem

- Labeled set: $\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^N$, Unabeled set: $\{\mathbf{z}_\ell \in \mathbb{R}^d\}_{\ell=1}^M$
- Loss function labeled set: $\sum_{i=1}^N \log(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w}))$ (convex)
- Loss function unlabeled set: $\sum_{j=1}^M T(\mathbf{z}_j^\top \mathbf{w})$ (non-convex)
- Regularizer: $\Omega(\mathbf{w}) = \sum_{j=1}^d \min(\eta, |w_j|)$ (non-convex penalty)

DC decomposition of $T(\cdot)$



Applications

Semi-supervised classification problem

- Labeled set: $\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^N$, Unabeled set: $\{\mathbf{z}_\ell \in \mathbb{R}^d\}_{\ell=1}^M$
- Loss function labeled set: $\sum_{i=1}^N \log(1 + \exp(-y_i \mathbf{x}_i^\top \mathbf{w}))$ (convex)
- Loss function unlabeled set: $\sum_{j=1}^M T(\mathbf{z}_j^\top \mathbf{w})$ (non-convex)
- Regularizer: $\Omega(\mathbf{w}) = \sum_{j=1}^d \min(\eta, |w_j|)$ (non-convex penalty)

				Classification Rate (%)	
dataset	d	N	M	Sparse Log	Sparse Transd.
la2	31472	61	2398	67.65±2.6	**70.23±3.1**
sports	14870	85	6778	81.26±5.0	**88.15±4.4**
classic	41681	70	5604	72.74±4.3	**86.97±2.2**
ohscal	11465	55	8873	70.35±2.4	**73.39±3.6**
real-sim	20958	723	57124	88.81±0.3	88.91±1.4
url	3.23×10^6	1000	40000	86.64±5.8	87.39±6.0

DC Proximal Newton can handle large scale and high-dimension data

Conclusion

- Non-convex problems: useful for certain machine learning applications
- DC proximal Newton able to handle efficiently large dimensional problems
- However computation of the gradient and Hessian remains costly $\longrightarrow$ use stochastic versions?
- Lack of theoretical analysis of local optimal solution

References

Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics*, 5(1):232, 2011.

E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing Sparsity by Reweighted ℓ_1 Minimization. *J Fourier Anal App*, 14:877–90, 2008.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456): 1348–1360, 2001.

Ildiko E. Frank and Jerome H. Friedman. A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, 35(2):109–135, 1993.

Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *Signal Processing, IEEE Transactions on*, 57(12):4686–4698, 2009.

Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proc. of ICML*, pages 37–45, 2013.

Yuling Jiao, Bangti Jin, and Xiliang Lu. A primal dual active set algorithm for a class of nonconvex sparsity optimization. Technical report, 2013.

Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*