# LASSO and Its Extensions

Xiaoning Qian (xqian@ece.tamu.edu)
ECE, Texas A&M University

ECEN689.603 Machine Learning with Networks

February 12, 2015

# Sparse Linear Models

## LASSO (Least Absolute Shrinkage and Selection Operator)

Corresponding Bayesian model: Laplace prior [$\rightarrow$ Gaussian Scale Models (GSM) for "easy" computation]

$$\min_{\mathbf{w}} \left\{ J(\mathbf{w}) = NLL(\mathbf{w}) + \lambda |\mathbf{w}|_{l_1} \right\},$$

in which $|\mathbf{w}|_{l_1} = \sum_i |w_i|$.

The objective function is convex but non-smooth (not differentiable)!

## Optimization algorithms

For most regularized problems (either by $l_1$ or $l_0$ norm and many others: Bridge, SCAD, etc.), there are two typical solution strategies:

1. Approximate the regularization terms by convex and smooth functions;
2. Adopt tricks in convex programming, for example, subgradient-based methods and proximal methods.

## Reference

M Schmidt, G Fung, R Rosales (2007) "FastOptimization Methods for L1 Regularization: A Comparative Study and Two New Approaches," European Conference on Machine Learning (ECML).

# Shooting Algorithm

## Subgradient

A subgradient or subderivative of a convex function $f(\mathbf{w})$ at $\mathbf{w}_0$ can be any scalar $g(\mathbf{w}_0) \in G(\mathbf{w}_0)$ such that $f(\mathbf{w}) - f(\mathbf{w}_0) \geq g(\mathbf{w}_0)(\mathbf{w} - \mathbf{w}_0)$.

## Toy example

For simplicity, we are fitting a univariate function:

$$\min_w \frac{1}{2}(\mathbf{y} - w\mathbf{x})^T(\mathbf{y} - w\mathbf{x}) + \lambda|w|$$

Compute the subgradient:

$$G(w) = \begin{cases} \mathbf{x}^T\mathbf{x}w - \mathbf{x}^T\mathbf{y} + \lambda & w > 0 \\ \left[-\mathbf{x}^T\mathbf{y} - \lambda, \quad -\mathbf{x}^T\mathbf{y} + \lambda\right] & w = 0 \\ \mathbf{x}^T\mathbf{x}w - \mathbf{x}^T\mathbf{y} - \lambda & w < 0 \end{cases}$$

If $0 \in G(w)$, we have the minimum of the convex function. Hence,

$$w = \begin{cases} \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} - \frac{\lambda}{\mathbf{x}^T\mathbf{x}} & \mathbf{x}^T\mathbf{y} > \lambda \\ 0 & |\mathbf{x}^T\mathbf{y}| < \lambda \\ \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} + \frac{\lambda}{\mathbf{x}^T\mathbf{x}} & \mathbf{x}^T\mathbf{y} < -\lambda \end{cases}$$

# Shooting Algorithm

Shooting algorithm is an iterative soft thresholding algorithm:

$$w = soft\left(\frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}}, \frac{\lambda}{\mathbf{x}^T\mathbf{x}}\right) = sign\left(\frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}}\right)\left(|\frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}}| - \frac{\lambda}{\mathbf{x}^T\mathbf{x}}\right)_+$$

.

For general problems, let $\mathbf{y}_{-j} = \mathbf{y} - \mathbf{X}_{-j}\mathbf{w}_{-j}$, then update $w_j$ iteratively.

---

## Shooting algorithm

1. Initialize by ridge regression: $\mathbf{w} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$.
2. **repeat**
3. **for** $j = 1 : D$ **do**
4. $a_j = \sum_n (x_j^n)^2$
5. $c_j = \sum_n x_j^n (y^n - \mathbf{w}^T\mathbf{x}^n + w_j x_j^n)$
6. $w_j = soft(\frac{c_j}{a_j}, \frac{\lambda}{a_j})$
7. **until** converged.

# Proximal Methods

## Proximal operator

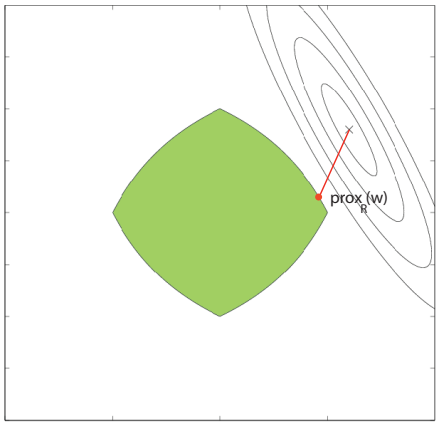
For more general regularization term $R(\mathbf{w})$, we may use proximal methods to solve the problem:

$$\min_{\mathbf{w}} \{J(\mathbf{w}) = NLL(\mathbf{w}) + \lambda R(\mathbf{w})\}.$$

For example, $R(\mathbf{w}) = 1_C(\mathbf{w})$ for any convex set $C$. The problem is in fact equivalent to:

$$\begin{aligned} &\min_{\mathbf{w}} NLL(\mathbf{w}) \\ &s.t. \quad \mathbf{w} \in C \end{aligned}$$

A proximal operator is defined as $prox_R(\mathbf{z}) = \arg\min_{\mathbf{w}}\{\frac{1}{2}\|\mathbf{z}-\mathbf{w}\|^2 + R(\mathbf{w})\}$. For a convex set $C$ and $R(\mathbf{w}) = 1_C(\mathbf{w})$, $prox_R(\mathbf{z}) = proj_C(\mathbf{z})$.




Special cases: soft thresholding, hard thresholding, bounding, normalization, etc.

# Proximal Gradient for LASSO

## Proximal gradient method for solving LASSO

Following the direction gradient descent, we are going to search for next update that minimizes the second-order approximation of the $NLL(\mathbf{w})$ at the current estimate:

$$\mathbf{w}_{k+1} = \arg\min_{\mathbf{z}} \left\{ R(\mathbf{z}) + NLL(\mathbf{w}_k) + (\mathbf{z}-\mathbf{w}_k)^T \nabla_{\mathbf{w}} NLL(\mathbf{w}_k) + \frac{1}{2\tau_k} \|\mathbf{z}-\mathbf{w}_k\|^2 \right\}$$

This is equivalent to solve

$$\mathbf{w}_{k+1} = \arg\min_{\mathbf{z}} \left\{ \tau_k R(\mathbf{z}) + \frac{1}{2} \|\mathbf{z} - \mathbf{w}_k + \tau_k \nabla_{\mathbf{w}} NLL(\mathbf{w}_k)\|^2 \right\}$$

Hence with $R(\mathbf{z}) = \lambda |\mathbf{z}|_{l_1}$,

$$\mathbf{w}_{k+1} = soft(\mathbf{w}_k - \tau_k \nabla_{\mathbf{w}} NLL(\mathbf{w}_k), \tau_k),$$

where $\tau_k$ can be taken the spectral stepsize known as Barzilai-Borwein (BB) method.

# Group LASSO

## $l_1/l_2$ norm regularization

There have been several papers, including some theoretical work on the proof of solution consistency, uniqueness, etc. Please read the corresponding references from the papers on eCampus if you are interested in the topic.

$$\min_{\mathbf{w}} NLL(\mathbf{w}) + \sum_{g} \lambda_g |\mathbf{w}_g|_{l_2}$$

It has many applications in multi-label classification, multi-task learning, and pathway-based analysis in bioinformatics.

We can also derive proximal gradient method for optimization. There are also other solution strategies.

# Fused LASSO

## Graph-based fused LASSO

$$\min_{\mathbf{w}} NLL(\mathbf{w}) + \lambda_1 |\mathbf{w}|_{l_1} + \lambda_2 \sum_{(i,j)\in E} |w_i - w_j|$$

Generally, optimization is more difficult. Bayesian learning and EM algorithm based on the GSM formulation can be implemented.
It can be also formulated differently as done in the "overlaping group LASSO".

## Two methods to further explore

1. GRACE: C Li and H Li (2008) "Network-constrained regularization and variable selection for analysis of genomic data," Bioinformatics, 24:1175-1118.
2. Difference Convex (DC) Algorithm: S Kim, W Pan, X Shen (2013) "Network-based penalized regression with application to genomic data," Biometrics, 69:582-593.

# Graph LASSO

## Gaussian Markov network model

Reminder: Learning multivariate Gaussian (Chapter 8).

Graph LASSO is "learning Gaussian with $l_1$ norm regularization":

$$\min_{\mathbf{\Gamma}} -\log\det(\mathbf{\Gamma}) + \text{trace}(\mathbf{S\Gamma}) + \lambda|\mathbf{\Gamma}|_{l_1},$$

in which $\mathbf{\Gamma} = \mathbf{\Sigma}^{-1}$ is the inverse covariance matrix and $\mathbf{S}$ is the sample covariance matrix.

Note that this is a convex programming problem and there exist good optimization algorithms.

Extensions to matrix decomposition problems ... ...

## Other interesting directions

For really large-scale problems, how can you "screen" features?

SAFE feature elimination in sparse supervised learning and several more.