ADMM Algorithms

October 9, 2024

1 Case Study: Fused Lasso

Consider the gray spikes in Figure 1, the results of a comparative genomic hybridization (CGH) experiment. Each of these represents the (log base 2) relative copy number of a gene in a cancer sample relative to a control sample; these copy numbers are plotted against the chromosome order of the gene.

- These data are very noisy, so that some kind of smoothing is essential.
- Biological considerations dictate that it is typically segments of a chromosome – rather than individual genes—that are replicated.
- Consequently, we might expect that the underlying vector of true copy numbers to be piecewise-constant over contiguous regions of a chromosome.

The **fused lasso** signal approximator exploits such structure within a signal, and is the solution of the following optimization problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \left\{ \frac{1}{2}\sum_{i=1}^{N}(y_i-\theta_i)^2 + \lambda_1 \sum_{i=1}^{n}|\theta_i| + \lambda_2 \sum_{i=2}^{n}|\theta_i-\theta_{i-1}| \right\} \tag{1}$$

- The first penalty is the familiar ℓ_1-norm, and serves to shrink the θ_i toward zero.
- The second penalty encourages neighboring coefficients θ_i to be similar, and will cause some to be identical.

There are more general forms of the fused lasso; we mention two here.

- We can generalize the notion of neighbors from a linear ordering to more general neighborhoods, for examples adjacent pixels in an image. This leads to a penalty of the form
$$\lambda_2 \sum_{i \sim i'} |\theta_i - \theta_{i'}|, \tag{2}$$
where we sum over all neighboring pairs $i \sim i'$.
- In (1) every observation is associated with a coefficient. More generally
$$\min_{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^p} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \right\},$$
Here the covariates x_{ij} and their coefficients β_j are indexed along some sequence j for which neighborhood clumping makes sense; (1) is clearly a special case.

Problem (1) and its relatives are all convex optimization problems, and so all have well-defined solutions. As in other problems of this kind, here we seek efficient path algorithms for finding solutions for a range of values for the tuning parameters. Although coordinate descent is one of our favorite algorithms for lasso-like problems, it need not work for the fused lasso (1), because the difference penalty is not a separable function of the coordinates.

We begin by considering the structure of the optimal solution $\hat{\boldsymbol{\theta}}(\lambda_1, \lambda_2)$ of the fused lasso problem (1) as a function of the two regularization parameters λ_1 and λ_2. The following result due to Friedman et al. (2007) provides some useful insight into the behavior of this optimum:

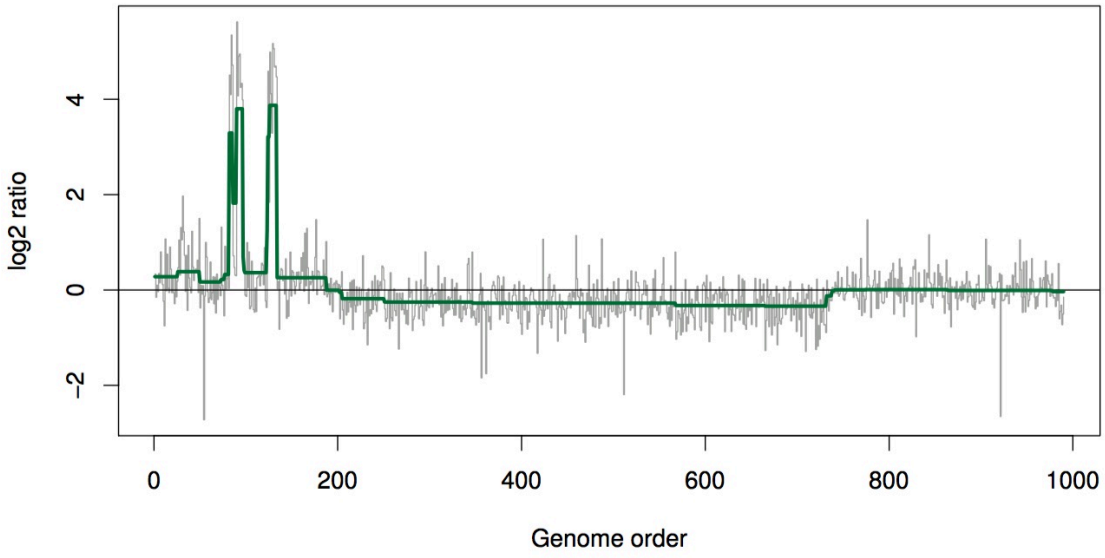

Figure 1: Fused lasso applied to CGH data. Each spike represents the copy number of a gene in a tumor sample, relative to that of a control (on the log base-2 scale). The piecewise-constant green curve is the fused lasso estimate.

Lemma 1. *For any $\lambda_1' > \lambda_1$, we have*

$$\hat{\theta}_i(\lambda_1', \lambda_2) = \mathcal{S}_{\lambda_1'-\lambda_1}(\hat{\theta}_i(\lambda_1, \lambda_2)) \text{ for each } i = 1, \ldots, n,$$

One important special case of Lemma 1 is the equality

$$\hat{\theta}_i(\lambda_1, \lambda_2) = \mathcal{S}_{\lambda_1}(\hat{\theta}_i(0, \lambda_2)) \text{ for each } i = 1, \ldots, n.$$

Consequently, if we solve the fused lasso with $\lambda_1 = 0$, all other solutions can be obtained immediately by soft thresholding. This useful reduction also applies to the more general versions of the fused lasso (2). On the basis of Lemma 1, it suffices to focus our attention

on solving the problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n}\left\{\frac{1}{2}\sum_{i=1}^{N}(y_i-\theta_i)^2+\lambda\sum_{i=2}^{n}|\theta_i-\theta_{i-1}|\right\} \tag{3}$$

We consider ADMM algorithm to solving (3).

2 Alternating Direction Method of Multipliers (ADMM)

2.1 Dual (sub)gradient methods

What if we can't derive dual (conjugate) in closed form, but want to utilize dual relationship? Turns out we can still use dual-based subgradient or gradient methods.

Example: consider the problem

$$\min_x f(x) \qquad \text{subject to} \qquad Ax=b$$

Lagrangian is

$$f(x)+u^\top(Ax-b)$$

Dual function is

$$\begin{aligned}\min_x f(x)+u^\top(Ax-b) &= \min_x f(x)-(-A^\top u)x-b^\top u\\ &= -f^*(-A^\top u)-b^\top u\end{aligned}$$

Its dual problem is

$$\max_u -f^*(-A^\top u)-b^\top u$$

where f^* is conjugate of f. Defining $g(u) = -f^*(-A^\top u) - b^\top u$. Note that

$$\partial g(u) = A\partial f^*(-A^\top u) - b$$

and recall that

$$x \in \partial f^*(-A^\top u) \iff x \in \arg\min_z f(z) + u^\top Az$$

Therefore the dual subgradient method (for maximizing the dual objective) starts with an initial dual guess $u^{(0)}$, and repeats for $k = 1, 2, 3, \ldots,$

$$\begin{aligned}
x^{(k)} &\in \arg\min_x f(x) + (u^{(k-1)})^\top Ax \\
u^{(k)} &= u^{(k-1)} + t_k \partial g(u^{(k-1)}) \\
&= u^{(k-1)} + t_k(A\partial f^*(-A^\top u^{(k-1)}) - b) \\
&= u^{(k-1)} + t_k(Ax^{(k)} - b)
\end{aligned}$$

where t_k are step sizes. chosen in standard ways. Recall that if f is strictly convex, then f^* is differentiable and so we get dual gradient ascent, which for $k = 1, 2, 3, \ldots,$

$$\begin{aligned}
x^{(k)} &= \arg\min_x f(x) + (u^{(k-1)})^\top Ax \\
u^{(k)} &= u^{(k-1)} + t_k(Ax^{(k)} - b)
\end{aligned}$$

2.2 Dual decomposition

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \qquad \text{subject to} \qquad Ax = b$$

Here $x = (x_1, \ldots, x_B) \in \mathbb{R}^n$ divides into B blocks of variables, with each $x_i \in \mathbb{R}^{n_i}$. We can also partition A accordingly

$$A = [A_1, \ldots, A_B], \quad \text{where} \quad A_i \in \mathbb{R}^{m \times n_i}$$

Simple but powerful observation, in calculation of (sub)gradient:

$$\begin{aligned} x^+ &= \arg\min_x \sum_{i=1}^B f_i(x_i) + u^\top A x \\ \Longleftrightarrow x_i^+ &\in \arg\min_{x_i} f_i(x_i) + u^\top A_i x_i, \qquad i = 1, \ldots, B \end{aligned}$$

i.e., minimization decomposes into B separate problems.

Dual decomposition algorithm: repeat for $k = 1, 2, 3, \ldots,$

$$\begin{aligned} x_i^{(k)} &\in \arg\min_{x_i} f_i(x_i) + (u^{(k-1)})^\top A_i x_i, \qquad i = 1, \ldots, B \\ u^{(k)} &= u^{(k-1)} + t_k \Big(\sum_{i=1}^B A_i x_i^{(k)} - b \Big) \end{aligned}$$

Can think of these steps as

- Broadcast: send u to each of the B processors, each optimizes in parallel to find x_i.
- Gather: collect $A_i x_i$ from each processor. update the global dual variable u.

Example with inequality constraints:

$$\min_x \sum_{i=1}^B f_i(x_i) \qquad \text{subject to} \qquad \sum_{i=1}^B A_i x_i \leq b$$

Dual decomposition (projected subgradient method) repeats for $k = 1, 2, 3, \ldots,$

$$
\begin{aligned}
x_i^{(k)} &\in \arg\min_{x_i} f_i(x_i) + (u^{(k-1)})^\top A_i x_i, \qquad i = 1, \ldots, B \\
v^{(k)} &= v^{(k-1)} + t_k(\sum_{i=1}^{B} A_i x_i^{(k)} - b) \\
u^{(k)} &= (v^{(k)})_+
\end{aligned}
$$

where $(\cdot)_+$ is component-wise thresholding, $(u_+)_i = \max\{0, u_i\}$.

2.3 Augmented Lagrangians

Disadvantage of dual (sub)gradient descent methods: require strong conditions to ensure primal iterates converge to solutions. Convergence properties can be improved by utilizing augmented Lagrangian. Transform primal:

$$
\begin{aligned}
&\min_x f(x) + \frac{\rho}{2}\|Ax - b\|_2^2 \\
&\text{subject to} \qquad Ax = b
\end{aligned}
$$

Clearly extra term $\frac{\rho}{2}\|Ax - b\|_2^2$ does not change problem. Use dual gradient ascent, repeat for $k = 1, 2, 3, \ldots,$

$$
\begin{aligned}
x^{(k)} &= \arg\min_x f(x) + (u^{(k-1)})^\top Ax + \frac{\rho}{2}\|Ax - b\|_2^2 \qquad (smooth) \\
u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} - b)
\end{aligned}
$$

when A has full column rank, primal is guaranteed strongly convex.

Notice step size choice $t_k = \rho$, for all k, in dual gradient ascent. Why? Since $x^{(k)}$

minimizes $f(x) + (u^{(k-1)})^\top Ax + \frac{\rho}{2}\|Ax - b\|_2^2$ over x, we have

$$\begin{aligned} 0 &\in \partial f(x^{(k)}) + A^\top(u^{(k-1)} + \rho(Ax^{(k)} - b)) \\ &= \partial f(x^{(k)}) + A^\top u^{(k)} \end{aligned}$$

This is the stationarity condition for the original primal problem. can show under mild conditions that $Ax^{(k)} - b$ approaches zero. i.e. primal iterates approach feasibility. hence in the limit, KKT conditions are satisfied and $x^{(k)}$, $u^{(k)}$ approach optimality.

- Advantage: much better convergence properties.
- Disadvantage: lose decomposability.

2.4 ADMM

good convergence properties of augmented Lagrangian + decomposability. Consider minimization problem

$$\min_x f_1(x_1) + f_2(x_2) \qquad \text{subject to } A_1x_1 + A_2x_2 = b$$

As before, we augment the objective

$$\begin{aligned} \min_x & f_1(x_1) + f_2(x_2) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2 \\ \text{subject to } & A_1x_1 + A_2x_2 = b \end{aligned}$$

Write the augmented Lagrangian as

$$L_\rho(x_1, x_2, u) = f_1(x_1) + f_2(x_2) + u^\top(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2$$

Now ADMM repeats the steps, for $k = 1, 2, 3, \ldots,$

$$
\begin{aligned}
x_1^{(k)} &= \arg\min_{x_1} L_\rho(x_1, x_2^{(k-1)}, u^{(k-1)}) \\
x_2^{(k)} &= \arg\min_{x_2} L_\rho(x_1^{(k)}, x_2, u^{(k-1)}) \\
u^{(k)} &= u^{(k-1)} + \rho(A_1 x_1^{(k)} + A_2 x_2^{(k)} - b)
\end{aligned}
$$

Note that the usual method of multipliers would have replaced the first two steps by

$$
(x_1^{(k)}, x_2^{(k)}) = \arg\min_{x_1, x_2} L_\rho(x_1, x_2, u^{(k-1)})
$$

2.4.1 ADMM in scaled form

It is often easier to express the ADMM algorithm in a scaled form, where we replace the dual variable u by a scaled variable $w = u/\rho$. In this parameterization, the ADMM steps are

$$
\begin{aligned}
x_1^{(k)} &= \arg\min_{x_1} f_1(x_1) + \frac{\rho}{2}\|A_1 x_1 + A_2 x_2^{(k-1)} - b + w^{(k-1)}\|_2^2 \\
x_2^{(k)} &= \arg\min_{x_2} f_2(x_2) + \frac{\rho}{2}\|A_1 x_1^{(k)} + A_2 x_2 - b + w^{(k-1)}\|_2^2 \\
w^{(k)} &= w^{(k-1)} + A_1 x_1^{(k)} + A_2 x_2^{(k)} - b
\end{aligned}
$$

Note that here the kth iterate $w^{(k)}$ is just given by a running sum of residuals:

$$
w^{(k)} = w^{(0)} + \sum_{i=1}^{k} A_1 x_1^{(i)} + A_2 x_2^{(i)} - b
$$

2.4.2 Example: lasso regression

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, recall the lasso problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

We can rewrite this as

$$\min_{\beta,\alpha} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\alpha\|_1 \qquad \text{subject to } \beta - \alpha = 0$$

The ADMM updates are

$$\begin{aligned}
\beta^+ &= \arg\min_{x_1} \frac{1}{2}\|y - X\beta\|_2^2 + \frac{\rho}{2}\|\beta - \alpha + w\|_2^2 \\
&= (X^\top X + \rho I)^{-1}(X^\top y + \rho(\alpha - w)) \\
\alpha^+ &= \arg\min_{x_1} \lambda\|\alpha\|_1 + \frac{\rho}{2}\|\beta^+ - \alpha + w\|_2^2 \\
&= \mathcal{S}_{\lambda/\rho}(\beta^+ + w) \\
w^+ &= w + \beta^+ - \alpha^+
\end{aligned}$$

2.4.3 Example: group lasso regression

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, recall the group lasso problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\beta_{(g)}\|_2$$

Rewrite as

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{G} c_g \|\alpha_{(g)}\|_2 \qquad \text{subject to } \beta - \alpha = 0$$

ADMM steps are:

$$\begin{aligned}
\beta^{(k)} &= (X^\top X + \rho I)^{-1}(X^\top y + \rho(\alpha^{(k-1)} - w^{(k-1)})) \\
\alpha_{(g)}^{(k)} &= \mathcal{R}_{c_g\lambda/\rho}(\beta_{(g)}^{(k)} + w_{(g)}^{(k-1)}), \qquad g = 1, \ldots, G \\
w^{(k)} &= w^{(k-1)} + \beta^{(k)} - \alpha^{(k)}
\end{aligned}$$

Notes:

- The matrix $X^\top X + \rho I$ is always invertible, regardless of X.
- If we compute a factorization (say Cholesky) in $O(p^3)$ flops, then each β update takes $O(p^2)$ flops.
- The α update applies the group soft-thresholding operator $\mathcal{R}_t$, which is defined as

$$\mathcal{R}_t(x) = \left(1 - \frac{t}{\|x\|}\right)_+ x$$

- Similar ADMM steps follow for a sum of arbitrary norms as regularizer, provided we know prox operator of each norm.
- ADMM algorithm can be rederived when groups have overlap (hard problem to optimize in general!).

2.5 Consensus ADMM

Consider a problem of the form:

$$\min_x \sum_{i=1}^{B} f_i(x)$$

The traditional method is to rewrite the problem as

$$\min_{x_1,\ldots,x_B} \sum_{i=1}^{B} f_i(x_i) \qquad \text{subject to } x_1 = x_2,\ x_2 = x_3,\ x_3 = x_4, \ldots, x_{B-1} = x_B$$

The variables are tangled together, not distributable! Instead, the consensus ADMM approach begins by reparametrizing:

$$\min_{x,x_1,\ldots,x_B} \sum_{i=1}^{B} f_i(x_i) \qquad \text{subject to } x_i = x,\ i = 1, \ldots, B$$

The consensus ADMM steps are:

$$\begin{aligned}
x_1^{(k)} &= \arg\min_{x_1} f_1(x_1) + \frac{\rho}{2}\|x_1 - x^{(k-1)} + w_1^{(k-1)}\|_2^2 \\
x_2^{(k)} &= \arg\min_{x_2} f_2(x_2) + \frac{\rho}{2}\|x_2 - x^{(k-1)} + w_2^{(k-1)}\|_2^2 \\
&\cdots\cdots \\
x_B^{(k)} &= \arg\min_{x_B} f_B(x_B) + \frac{\rho}{2}\|x_B - x^{(k-1)} + w_B^{(k-1)}\|_2^2 \\
x^{(k)} &= \frac{1}{B}\sum_{i=1}^{B}(x_i^{(k)} + w_i^{(k-1)}) \\
w_i^{(k)} &= w_i^{(k-1)} + x_i^{(k)} - x^{(k)},\ i = 1, \ldots, B
\end{aligned}$$