10-725/36-725: Convex Optimization **Fall 2015**

Lecture 19: November 5

Lecturer: Ryan Tibshirani *Scribes: Hyun Ah Song*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

19.1 Last time: review of optimization toolbox

Throughout the course, we have learned about several optimizaiton tools, including first order methods, second order methods, and interior point methods. Using this toolbox of optimization, we can solve interesting class of problems. Last time, we went through a big algorithm of table that tells us what method to use for what types of problems. In this lecture, we will go through high-level reasoning for regularized estimation problems.

19.2 Sum of norms regularization

The class of problem we will go through in this lecture is called *sums of norms regularization* problem. It has a standard following form.

$$\min_{\beta} \quad f(\beta) + \lambda \sum_{j=1}^{J} \|D_j\beta\|_{q_j}. \tag{19.1}$$

Here, $f : \mathbb{R}^p \to \mathbb{R}$ is a smooth, convex function, $D_j \in \mathbb{R}^{m_j \times p}$ is a penalty matrix, q_j is a norm parameter for $j = 1, \ldots, J$, and $\lambda \geq 0$ is a regularization parameter.

19.2.1 A special example: Lasso

With $J = 1, D = I, q = 1$, the sums of norms regularization problem becomes a form of lasso problem.

$$f(\beta) = \|y - X\beta\|_2^2. \tag{19.2}$$

Unpenalized intercept can be included by adding a column of zeros to D.

19.3 Fused lasso or total variation denoising

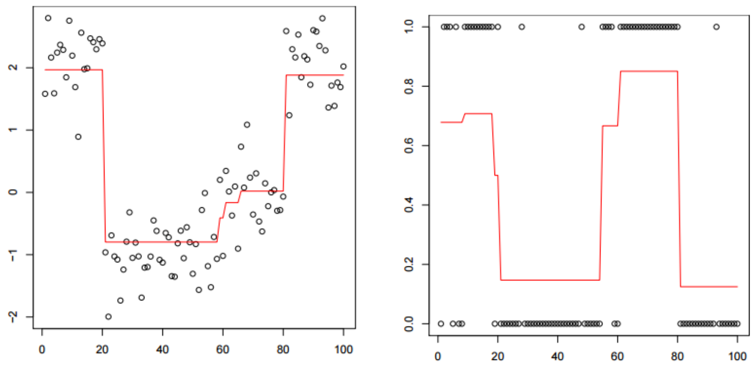
19.3.1 1d

If we let $J = 1, q = 1$, D as following where we have -1 in the diagonal, 1 in the super-diagonal, and 0 elsewhere,

$$D = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{19.3}$$

then $\|D\beta\|_1 = \sum_{i=1}^{n-1} |\beta_i - \beta_{i+1}|$. This problem is called *fused lasso* or *total variation denoising* in *1d*. Here, we are regularing the *differences* in β, (i.e. promoting sparsity in the differences $\hat{\beta}_i - \hat{\beta}_{i+1}$, so that there are $\hat{\beta}_i = \hat{\beta}_{i+1}$ in many places. Here we are not sparsifying β itself. Then the solution $\hat{\beta}$ we get is piecewise constant.

This setting is used commonly for *signal approximation*, where we want to estimate the average of the observations in $\hat{\beta}$. In Figure 19.1, solution $\hat{\beta}$ of the cases where f is defined for (a) Gaussian loss $f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_i)^2$ and (b) logistic loss $f(\beta) = \sum_{i=1}^{n}(-y_i\beta_i + \log(1+e^{\beta_i}))$ is shown. We see that in the solution β, it captures the input pattern with structure in the signal in form of piecewise constant. For the logistic loss case, we also observe that $\hat{\beta}$ is approximated to either 0 or 1.

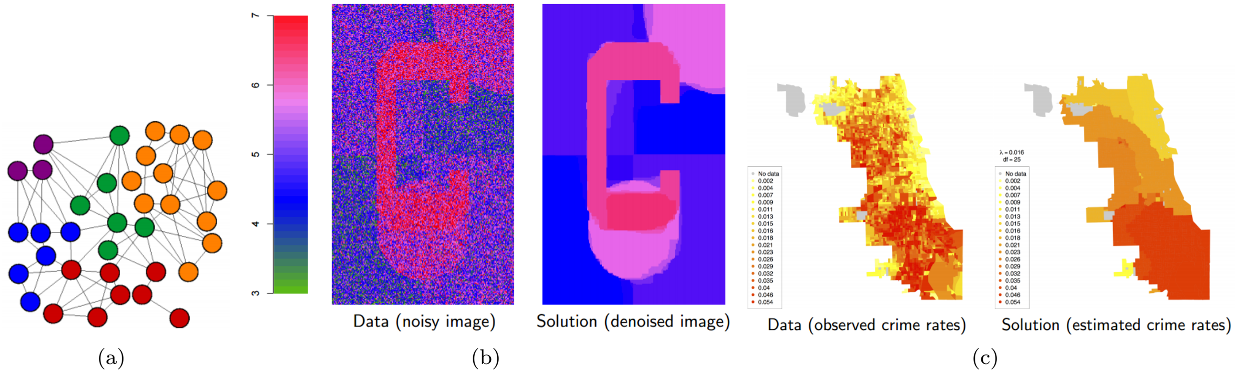


(a) (b)

Figure 19.1: Fused lasso example - 1d. (a) Gaussian loss, and (b) Logistic loss

19.3.2 Graph

We can generalize D and design the fused lasso (or total variation denoising) where it is associated with a *graph*. Given a graph $G = (\{1, \dots, n\}, E)$, let D be a $|E| \times n$ matrix, where $D_l = (0, \dots, -1, \dots, 1, \dots, 0)$ for the lth row if $e_l = (i, j)$. Then we have $\|D\beta\|_1 = \sum_{(i,j)\in E} |\beta_i - \beta_j|$. As shown in Figure 19.2 (a), we see that we have solution is piecewise constant over grpah, $\hat{\beta}_i = \hat{\beta}_j$ for $(i, j) \in E$. In Figure 19.2 (b) and (c), we have cases where we have a Gaussian loss $f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_i)^2$ for 2d graph of an image, and Chicago crime rate graph, respectively. We see the estimation shows smoothened version of the original noisy data, where it shows distinct regions of a uniform color across the vertices within each region. This seems reasonable as solution for such cases of image or regional analysis.



Data (noisy image) Solution (denoised image) Data (observed crime rates) Solution (estimated crime rates)

(a) (b) (c)

Figure 19.2: Fused lasso example for graph.

19.4 Fused lasso with a regressor matrix

Until now, we have been looking at signal approximation problems, where we seek an output that is a smoothened approximation of the input data. Now, we will consider a more complicated problem where we have a regressor matrix, so that instead of imposing a structure in the output, we are imposing a structure in the weights that correspond to the coefficients of the regressors, which forces coefficients to be constant over graph. This enables us to find the groups of related predictors.

Let $X \in \mathbb{R}^{n \times p}$ a predictor matrix with structure in its columns that have been measured over nodes of a graph. We can consider the case where $J = 1, q = 1$ with fused lasso D over graph, and Gaussian loss function $f(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2$ or logistic loss function $f(\beta) = \sum_{i=1}^{n}(-y_i x_i^T\beta + \log(1 + \exp(x_i^T\beta)))$. Here x_i is ith row of matrix X.

In Figure 19.3, we have an experimental result of fused lasso with a regressor matrix problem. The goal in this experiment is to predict whether a patient has Alzheimer's disease or not given MRI brain images of the patient. We have pictures of the models of MRI brain images that are used to make predictions. The first column shows models trained using fused lasso penalty that encourages voxels to have constant values that are adjacent to each other. The second and third columns are the models of the simple lasso penalty that encourages learning sparse set of voxels but not necessarily forcing voxels to have spatial property. The first row shows best models that show the voxels predictive of the Alzheimer's disease, and the second row shows features that are most predictive of the Alzheimer's disease. We observe that the colored region in the second row corresponds to the hippocampus region, which is known to be relevant to Alzheimer.

19.5 Algorithms for the fused lasso

Now let's learn how to solve the following fused lasso problem.

$$\min_{\beta} \quad f(\beta) + \lambda\|D\beta\|_1. \tag{19.4}$$

Let's examine several different methods to solve this problem.

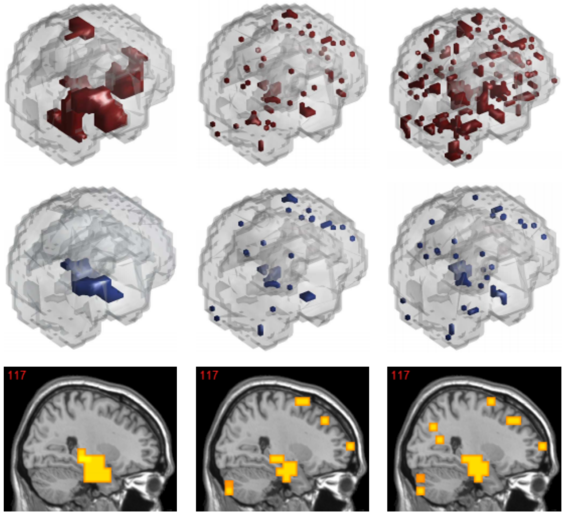


Figure 19.3: Fused lasso with a regressor matrix - brain imaging example.

19.5.1 Subgradient method

The subgradient of the equation 19.4 is

$$g = \nabla f(\beta) + \lambda D^T \gamma \tag{19.5}$$

where $\gamma_i \in \partial\|x\|_1$ is evaluated at $x = D\beta$

$$\gamma_i \in \begin{cases} \{sign((D\beta)_i)\} & \text{if } (D\beta)_i \neq 0 \\ [-1, 1] & \text{if } (D\beta)_i = 0 \end{cases} \tag{19.6}$$

- Downside: convergence is slow.
- Upside: g is easy to compute, when we know ∇f. In this case, $g = \nabla f(\beta) + \lambda \sum_{i \in S} sign((D\beta)_i) \cdot D_i$ where $S = supp(D\beta)$; we ignore all nonzeros in β and sum up all active components.

19.5.2 Proximal gradient descent

We can compute the proximal operator as following.

$$\text{prox}_t(\beta) = \text{argmin}_z \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\|Dz\|_1 \tag{19.7}$$

Here, we see that the solving proximal operator is a fused lasso signal approximation problem with a Gaussian loss. This is not easy since we have D in the L1 norm penalty unlike the soft-thresholding case where $D = I$.

We can reparameterize L1 norm $\|D\beta\|_1$ to linear inequality constraints by introducing a new set of variables corresponding to the absolute value of the components. Then we can use interior point method solve the problem. But in this approach, we have twice as many variables as the original problem.

It is better to consider formulating this problem into dual problem to solve it, where we will explain in the following section.

19.6 Fused lasso dual probelm

Our original primal problem is

$$\min_{\beta} \quad f(\beta) + \lambda\|D\beta\|_1. \tag{19.8}$$

By introducing a new variable z,

$$\begin{aligned} \min_{\beta,z} \quad & f(\beta) + \lambda\|z\|_1. \\ \text{subject to} \quad & z = D\beta \end{aligned} \tag{19.9}$$

The Lagrangian is given as $L(\beta, z, u) = f(\beta)+\lambda\|z\|_1+u^T(D\beta-z)$, so the Lagrangian dual function becomes

$$\begin{aligned} &\min_{\beta,z} f(\beta) + \lambda\|z\|_1 + u^T(D\beta - z) \\ &= \min_{\beta} [f(\beta) + u^T(D\beta)] + \min_{z} [\|z\|_1 - u^T z] \\ &= -f^*(-D^T u) - I(\|z\|_\infty \leq \lambda) \end{aligned} \tag{19.10}$$

Thus, we have dual problem,

$$\begin{aligned} \min_{u} \quad & f^*(-D^T u) \\ \text{subject to} \quad & \|u\|_\infty \leq \lambda \end{aligned} \tag{19.11}$$

Here f^* is the conjugate function of f, $u \in \mathbb{R}^m, \beta \in \mathbb{R}^n$ where m is the number of rows in D.

By KKT conditions, we see that the primal and dual solutions $\hat{\beta}, \hat{u}$ are linked:

$$\nabla f(\hat{\beta}) + D^T\hat{u} = 0 \tag{19.12}$$

$$\hat{u}_i = \begin{cases} \{\lambda\} & \text{if } (D\hat{\beta})_i > 0 \\ \{-\lambda\} & \text{if } (D\hat{\beta})_i < 0, i = 1, \ldots, m \\ [-\lambda, \lambda] & \text{if } (D\hat{\beta})_i = 0 \end{cases} \tag{19.13}$$

From the second property, we see that the sparsity pattern in the primal and dual variales are reversed: if u_i is at the boundary of $(-\lambda, \lambda)$, then $\hat{\beta}$ is inside of L1 norm (nonzero), and if $\hat{\beta} = 0$ then $\hat{u}_i$ is inside the $(-\lambda, \lambda)$ (nonzero).

When we compare primal (equation 19.8) and dual problem (equation 19.11), we see that we have moved D from non-smooth term to the smooth term of the objective, which makes it easier for the proximal operator.

Let's consider two methods to solve this dual problem of equation 19.11: proximal gradient descent, and interior point method.

19.6.1 Dual proximal gradient descent

From dual problem, we have proximal operator,

$$\begin{aligned} \min_{u} \quad & \text{prox}_t(u) = \text{argmin}_z \frac{1}{2t}\|u - z\|_2^2 \\ \text{subject to} \quad & \|z\|_\infty \leq \lambda \end{aligned} \tag{19.14}$$

This is much easier to solve compared to the proximal operator of the original primal problem (equation 19.7). The solution is given by simply projecting the variable onto a m dimensional box $[-\lambda, \lambda]^m$:

$$\nabla f(\hat{\beta}) + D^T \hat{u} = 0 \tag{19.15}$$

$$\hat{z}_i = \begin{cases} \lambda & \text{if } u_i > \lambda \\ -\lambda & \text{if } u_i < \lambda, i = 1, \dots, m \\ u_i & \text{if } u_i \in [-\lambda, \lambda] \end{cases} \tag{19.16}$$

19.6.2 Dual interior point method

Let's rewrite the dual problem as

$$\begin{aligned} \min_u \quad & f^*(-D^T u) \\ \text{subject to} \quad & -\lambda \leq u_i \leq \lambda, i = 1, \dots, m \end{aligned} \tag{19.17}$$

In the interior point method, we substitute inequality constraints with log barrier function, so we have problem

$$\min_u \quad t \cdot f^*(-D^T u) + \phi(u) \tag{19.18}$$

where the log barrier function is $\phi(u) = -\sum_{i=1}^m (\log(\lambda - u_i) + \log(u_i + \lambda))$. Given above problem, we can solve it using Newton's method.

Now let's think of the efficiency of the Newton updates. We do the Newton updates with respect to direction $H^{-1}g$ where the gradient and Hessian of the dual criterion function $F(u) = tf^*(-D^T u) + \phi(u)$ is given as below.

$$g = \nabla F(u) = -t \cdot D(\nabla f^*(-D^T u)) + \nabla \phi(u) \tag{19.19}$$

$$H = \nabla^2 F(u) = t \cdot D(\nabla^2 f^*(-D^T u))D^T + \nabla^2 \phi(u) \tag{19.20}$$

From H, by inspection, we see that the second term (Hessian of log barrier, $\nabla^2 \phi(u)$) is simple diagonal matrix. Therefore, the first term (f^* and D) determines the overall difficulty of computing Newton updates. (If Hessian of loss term $\nabla^2 f^*(v)$ and D are structured then often $D\nabla^2 f^*(v)D^T$ is also structured, making it easy to solve Newton steps.)

19.7 Summary of the methods for solving fused lasso problem

- Primal subgradient method: cheap iterations, but slow convergence.
- Primal proximal graident: expensive iterations (evaluation of proximal operator), medium convergence.
- Dual proximal gradient: very cheap iterations, medium convergence.
- Dual iterior point method: may or may not be expensive iteration depending on f^*, D, rapid convergence.

19.8 Case study: fused lasso, Gaussian or logistic signal approximation

19.8.1 Problem formulation

Let's consider a fused lasso problem over a general graph, where we want to solve for D. The primal problems with Gaussian or logistic loss are given as followings.

$$\begin{aligned}&\min_{\beta} \quad \frac{1}{2}\sum_{i=1}^{n}(y_i-\beta_i)^2+\lambda\|D\beta\|_1\\&\min_{\beta} \quad \sum_{i=1}^{n}(-y_i\beta_i+\log(1+e^{\beta_i}))+\lambda\|D\beta\|_1\end{aligned} \tag{19.21}$$

If we want to get a solution of high accuracy, which method should we use? Primal methods (Primal subgradient and primal proximal gradient) are not good choices since they are slow and intractable, respectively. Let's formulate the dual problem, and inspect which of the dual methods (dual proximal gradient method or dual interior-point method) is desirable.

19.8.2 Dual formulation

The conjugate functions f^* for Gaussian and logistic loss are given as following.

$$f^*(v)=\frac{1}{2}\sum_{i=1}^{n}y_i^2-\frac{1}{2}\sum_{i=1}^{n}(y_i+v_i)^2 \text{ (Gaussian loss case)} \tag{19.22}$$

$$f^*(v)=\sum_{i=1}^{n}((v_i+y_i)\log(v_i+y_i)+(1-v_i-y_i)\log(1-v_i-y_i)) \text{ (logistic loss case)} \tag{19.23}$$

From dual solutions, we can recover primal solutions as following.

$$\hat{\beta}=y-D^T\hat{u} \text{ (Gaussian loss case)} \tag{19.24}$$

$$\hat{\beta}_i=y_i\log(y_i(D^T\hat{u})_i)+y_i\log(1-y_i(D^T\hat{u})_i), i=1,\ldots,n \text{ (logistic loss case)} \tag{19.25}$$

Given this dual formulation, let's think of solving this dual problem by proximal gradient descent method ad interior-point method.

19.8.3 Dual proximal gradient method

- Cheap iterations: projection of $u+tD\nabla f^*(-D^Tu)$ on to a box repeatedly.
- Slow convergence: as shown in Figure 19.4, as we have more number of nodes in graph, the condition number in D increases, which means more iterations are required for proximal to converge.

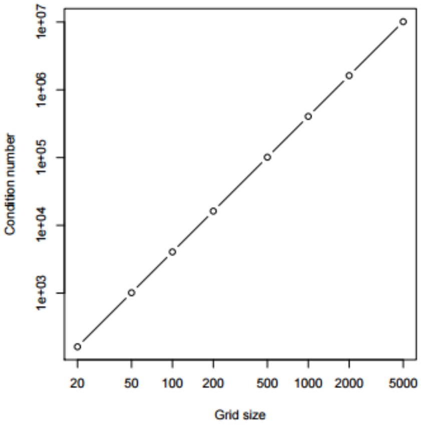
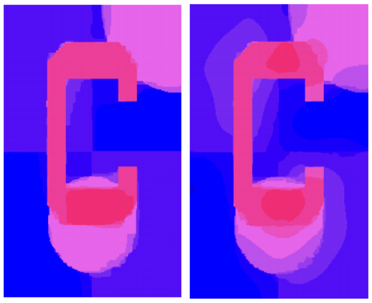


Figure 19.4: Condition numbers for a 2d gird graph



(a) (b)

Figure 19.5: (a) Dual interior point method (10 iterations) (b) Dual proximal gradient method (1000 iterations)

19.8.4 Dual interior-point method

In dual interior-point method, we need to solve for Newton updates, which requires computation of gradient and Hessian. The Hessians of the dual problem with Gaussian loss and logistic loss are given as follows.

$$\nabla^2 f^*(v) = I \text{ (Gaussian loss case)} \tag{19.26}$$

$$\nabla^2 f^*(v) = \text{diag}\left(\frac{1}{v_i + y_i} + \frac{1}{1 - v_i - y_i}, i = 1, \dots, m\right) \text{ (logistic loss case)} \tag{19.27}$$

From this we see that Newton steps involves solving a linear system $Hx = g$ in $H = DA(u)D^T + B(u)$. Here $A(u), B(u)$ are structured matrices (diagonal matrices). This makes the problem to be solved efficiently with $O(n)$ flops.

As a conclusion, to solve a fused lasso problem with high accuracy, dual interior-point method is desirable: cheap iterations and fast convergence to high accuracy. In Figure 19.5, we have solution for the image approximation problem using (a) dual interior-point method with 10 iterations and (b) dual proximal gradient method with 1000 iterations. We see that solution of the dual interior-point method is more accurate even with smaller number of iterations compared to that of dual proximal gradient method.

19.9 Case study: fused lasso, linear or logistic regression

In this section, let's think about fused lasso problem with regressor matrix $X \in \mathbb{R}^{n\times p}$ with losses $f(\beta) = \frac{1}{2}\sum_{i=1}^n (y_i - x_i^T\beta)^2$ or $f(\beta) = \sum_{i=1}^n(-y_i x_i^T\beta + \log(1+\exp(x_i^T\beta))$. If we denote the loss function $f(\beta) = h(X\beta)$ for simplicity, we have problem in primal and dual formulation as below.

$$\text{Primal: } \min_{\beta} \quad h(X\beta) + \lambda\|D\beta\|_1 \tag{19.28}$$

$$\begin{aligned} \text{Dual: } \min_{\beta} \quad & h^*(v) \\ \text{subject to} \quad & X^T v + D^T u = 0, \|u\|_\infty \le \lambda \end{aligned} \tag{19.29}$$

where $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ are dual variables.

The primal and dual solutions $\hat{\beta}, \hat{u}, \hat{v}$ must satisfy below conditions.

$$\nabla h(X\hat{\beta}) - \hat{v} = 0 \text{ or equivalently } X^T\nabla h(X\hat{\beta}) + D^T\hat{u} = 0 \tag{19.30}$$

Computing primal variable $\hat{\beta}$ from dual variable $\hat{u}$ requires solving a linear system in X, and this is very expensive.

How should we solve this problem? Let's consider the four methods again.

- Dual proximal gradient descent: intractable. (Proximal operator is $\text{prox}_t(u,v) = \text{argmin}_{X^T w + D^T z = 0} \frac{1}{2t}\|u - z\|_2^2 + \frac{1}{2t}\|v - w\|_2^2 + \|u\|_\infty$. This is finding projection of (u,v) into the intersection of a plane and a lower-dimensional box.)
- Dual interior point method: not favorable. (We have equality constraint $X^T v = D^T u = 0$. Previously, the Hessian had nice sparse structure, but now we have dense X matrix in Hessian, so we cannot derive dual interior-point method with $O(n)$ flops.)
- Primal subgradient method: slow.
- Primal proximal gradient descent: best option. (The gradient $\nabla f(\beta) = X^T\nabla h(X\beta)$ can be computed easily by chain rule, and the proximal operator $\text{prox}_t(\beta) = \text{argmin}_z \frac{1}{2t}\|\beta - z\|_2^2 = \lambda\|Dz\|_1$ is exactly the problem of graph fused lasso with Gaussian loss without regressors. This prox operator can be evaluated by fast dual interior point method. Previously, we ruled out this method because the difficulty of solving the prox operator seemed equivalent to that of the original problem. However in this case with regressor, it is a good tradeoff between the difficulty of solving prox operator and being freed from solving linear system involving dense regressor X since prox operator does not involve regressor matrix)

19.10 Case study: 1d fused lasso, linear or logistic regression

Let's think about general case of fused lasso where it is applied to general chain graph with D given as 19.3.

The prox operator is

$$\text{prox}_t(\beta) = \text{argmin}_z \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\sum_{i=1}^n |z_i - z_{i+1}| \tag{19.31}$$

This can be computed by dyanmic programming or taut-string methods in $O(n)$ operations. In Figure 19.6, comparison on time consumption for solving the problem via dynamic programming versus baded matrix is plotted. We see that evaluating prox operation using dynamic programming is really fast.

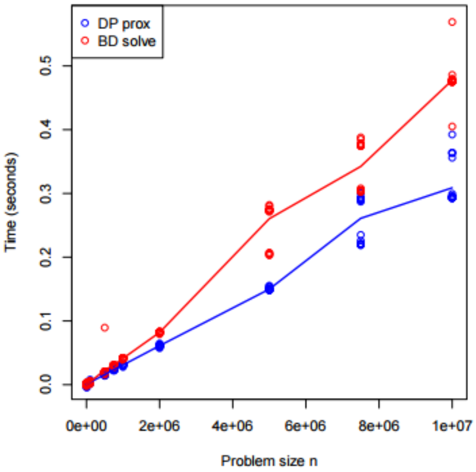


Figure 19.6: Dynamic programming vs. banded matrix solve

19.11 Case study: 1d fused lasso, Gaussian or logistic signal approximation

Let's compare primal proximal gradient and dual interior point method for 1d fused lasso problem, where both of them seem strong choices.

We have primal and dual problem as follows.

$$\text{Primal: } \min_{\beta} f(\beta) + \lambda \|D\beta\|_1 \tag{19.32}$$

$$\text{Dual: } \min_{u} f^*(-D^T u) \text{ subject to } \|u\|_\infty \leq \lambda \tag{19.33}$$

Then which algorithm should we use?

- Large λ: primal algorithm. (many of $(D\hat{\beta})_i = 0$ in primal, many of $\hat{u}_i \in (-\lambda, \lambda)$ in dual; fewer effective parameters in primal optimization)
- Small λ: dual algorithm. (many of $(D\hat{\beta})_i \neq 0$ in primal, many of $|\hat{u}_i| = \lambda$ in dual; fewer effective parameters in dual optimization)