LECTURE 16: LASSO AND COORDINATE DESCENT

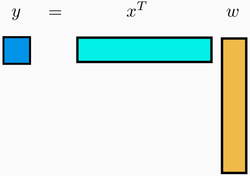
STAT 598z: Introduction to computing for statistics

Vinayak Rao

Department of Statistics, Purdue University

March 20, 2019

Problem: Given training data $(X, y) \equiv \{x_i, y_i\}$,
minimize $\mathcal{L}(w) = \frac{1}{2}(Y - X^T w)^2$



Bias-variance and regularization

Problem: Given training data $(\mathsf{X}, \mathsf{y}) \equiv \{\mathsf{x}_i, y_i\}$,
minimize $\mathcal{L}(\mathsf{w}) = \frac{1}{2}(\mathsf{Y} - \mathsf{X}^T\mathsf{w})^2$

To reduce variance (i.e. sensitivity to small changes in training data) , add a penalty $\Omega(\mathsf{w})$:

$$\hat{\mathsf{w}} = \text{argmin } \mathcal{L}(\mathsf{w}) + \lambda\Omega(\mathsf{w})$$

Problem: Given training data $(\mathbf{X}, \mathbf{y}) \equiv \{\mathbf{x}_i, y_i\}$,
minimize $\mathcal{L}(\mathbf{w}) = \frac{1}{2}(\mathbf{Y} - \mathbf{X}^T\mathbf{w})^2$

To reduce variance (i.e. sensitivity to small changes in training data) , add a penalty $\Omega(\mathbf{w})$:

$$\hat{\mathbf{w}} = \text{argmin } \mathcal{L}(\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

Ridge regression/L_2 regression:

- $\Omega(\mathbf{w}) = \|\mathbf{w}\|_2^2$
- $\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \quad$ (Shrinkage)

Bias-variance and regularization

Problem: Given training data $(\mathrm{X}, \mathrm{y}) \equiv \{\mathrm{x}_i, y_i\}$,
minimize $\mathcal{L}(\mathrm{w}) = \frac{1}{2}(\mathrm{Y} - \mathrm{X}^T\mathrm{w})^2$

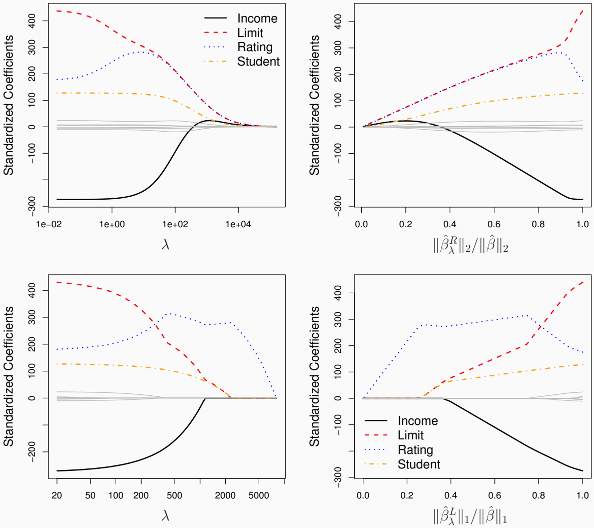
To reduce variance (i.e. sensitivity to small changes in training data) , add a penalty $\Omega(\mathrm{w})$:

$$\hat{\mathrm{w}} = \mathrm{argmin}\ \mathcal{L}(\mathrm{w}) + \lambda\Omega(\mathrm{w})$$

LASSO:

- $\Omega(\mathrm{w}) = \|\mathrm{w}\|_1 \quad (\|\mathrm{w}\|_1 = |w_1| + |w_2| + \cdots + |w_p|)$
- Shrinkage and selection
  (w is sparse with some components equal to 0)
- No simple closed-form solution

Credit data set (average credit card debt)



(top) ridge, (bottom) LASSO. (James, Witten, Hastic and Tibshirani)

Regularization as constrained optimization

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_2^2$ is equivalent to

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t. \; \|\mathbf{w}\|_2^2 \leq \gamma$

(Note: γ will depend on data)

Regularization as constrained optimization

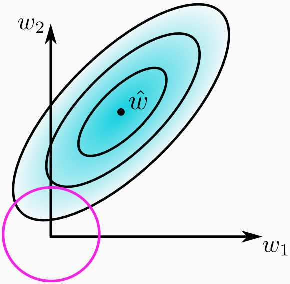
$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_2^2$ is equivalent to

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t. \; \|\mathbf{w}\|_2^2 \leq \gamma$

(Note: γ will depend on data)

First problem: regularized optimization

Second problem: constrained optimization



Regularization as constrained optimization

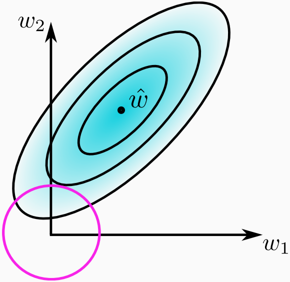
$\operatorname{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_2^2$ is equivalent to

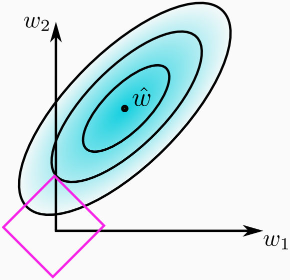
$\operatorname{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t.\ \|\mathbf{w}\|_2^2 \leq \gamma$

(Note: γ will depend on data)

First problem: regularized optimization

Second problem: constrained optimization





Lasso

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t. \; \|\mathbf{w}\|_1 \leq \gamma$

$\|\mathbf{w}\|_1 = \sum_{j=1}^{p} |w_j|$ is the ℓ_1-norm.

Lasso

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t.\ \|\mathbf{w}\|_1 \leq \gamma$

$\|\mathbf{w}\|_1 = \sum_{j=1}^{p} |w_j|$ is the ℓ_1-norm.

Lasso: least absolute shrinkage and selection operator.

$$\hat{\mathbf{w}} = \text{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_1$$

Lasso

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t. \ \|\mathbf{w}\|_1 \leq \gamma$

$\|\mathbf{w}\|_1 = \sum_{j=1}^{p} |w_j|$ is the ℓ_1-norm.

Lasso: least absolute shrinkage and selection operator.

$$\hat{\mathbf{w}} = \text{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_1$$

- Penalizes small w_j more than ridge regression.
- Tolerates larger w_j more than ridge regression.

Lasso

$\text{argmin}(\mathbf{y} - \mathbf{X}^\top \mathbf{w})^2 \quad s.t.\ \|\mathbf{w}\|_1 \leq \gamma$

$\|\mathbf{w}\|_1 = \sum_{j=1}^{p} |w_j|$ is the ℓ_1-norm.

Lasso: least absolute shrinkage and selection operator.

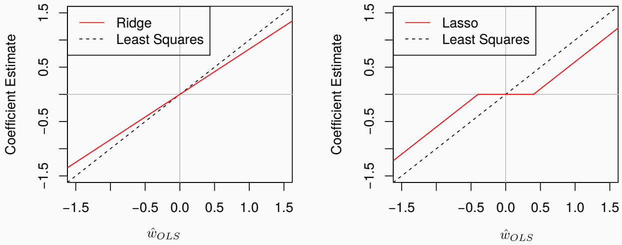
$$\hat{\mathbf{w}} = \text{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_1$$

- Penalizes small w_j more than ridge regression.
- Tolerates larger w_j more than ridge regression.

Result:

- $\hat{\mathbf{w}}_{LASSO}$ has some components *exactly* equal to zero.
- Performs feature selection.

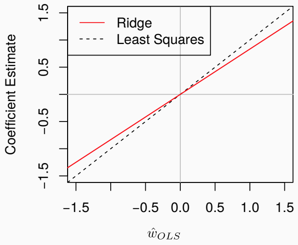
The 1-D case

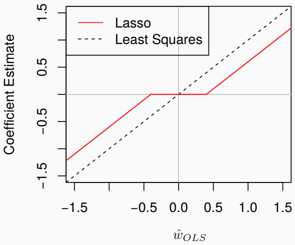


In the 1-d case, $(\mathbf{x}, \mathbf{y}) \equiv \{x_i, y_i\}$

Least-squares solution: $\hat{w}_{ols} = \frac{\mathbf{x}^\top \mathbf{y}}{\mathbf{x}^\top \mathbf{x}}$

The 1-D case



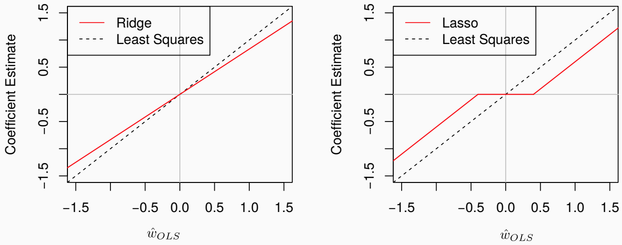


In the 1-d case, $(\mathbf{x}, \mathbf{y}) \equiv \{x_i, y_i\}$

Least-squares solution: $\hat{w}_{ols} = \frac{\mathbf{x}^\top \mathbf{y}}{\mathbf{x}^\top \mathbf{x}}$

Ridge regression solution: $\hat{w}_{ridge} = \frac{\mathbf{x}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x} + \lambda}$

The 1-D case



In the 1-d case, $(\mathbf{x}, \mathbf{y}) \equiv \{x_i, y_i\}$

Least-squares solution: $\hat{w}_{ols} = \frac{\mathbf{x}^\top \mathbf{y}}{\mathbf{x}^\top \mathbf{x}}$

Ridge regression solution: $\hat{w}_{ridge} = \frac{\mathbf{x}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x} + \lambda}$

LASSO solution?

Optimization in R

Use the `optim` function

Syntax:

```
optim(par, fn, gr = NULL, ...,
  method = c('Nelder-Mead', 'BFGS', 'CG', 'L-BFGS-B', 'SANN',
             'Brent'),
  lower = -Inf, upper = Inf,
  control = list(), hessian = FALSE)
```

`fn`: function to be optimized

`gr`: gradient function (calculate numerically if `NULL`)

`par`: initial value of parameter to be optimized (should be first argument of `fn`)

LASSO IN 1-D

$$\hat{w} = \operatorname{argmin} \mathcal{L}(w) = \operatorname{argmin} \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$$

LASSO IN 1-D

$\hat{w} = \text{argmin}\ \mathcal{L}(w) = \text{argmin}\ \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$

At the minimum,

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w} = 0$$

LASSO IN 1-D

$\hat{w} = \operatorname{argmin} \mathcal{L}(w) = \operatorname{argmin} \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$

At the minimum,

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w} &= 0 \\ -\sum_{i=1}^{n}(y_i - wx_i)x_i + \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w} &= 0 \end{aligned}$$

LASSO IN 1-D

$\hat{w} = \text{argmin } \mathcal{L}(w) = \text{argmin } \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$

At the minimum,

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w} &= 0 \\ -\sum_{i=1}^{n}(y_i - wx_i)x_i + \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w} &= 0 \\ -\sum_{i=1}^{n}y_i x_i + w\sum_{i=1}^{n}x_i^2 + \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w} &= 0 \end{aligned}$$

$\hat{w} = \text{argmin}\ \mathcal{L}(w) = \text{argmin}\ \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$

At the minimum,

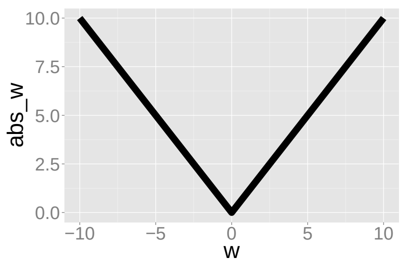
$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w} &= 0 \\
-\sum_{i=1}^{n}(y_i - wx_i)x_i + \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w} &= 0 \\
-\sum_{i=1}^{n} y_i x_i + w\sum_{i=1}^{n} x_i^2 + \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w} &= 0 \\
w = \frac{\sum_{i=1}^{n} y_i x_i - \lambda\frac{\mathrm{d}|w|}{\mathrm{d}w}}{\sum_{i=1}^{n} x_i^2}
\end{aligned}$$

Subgradients

$$w = \frac{\sum_{i=1}^{n} y_i x_i - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\sum_{i=1}^{n} x_i^2} = \frac{\mathbf{y}^\top \mathbf{x} - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\mathbf{x}^\top \mathbf{x}} : \quad \text{What is } \frac{\mathrm{d}|w|}{\mathrm{d}w}?$$

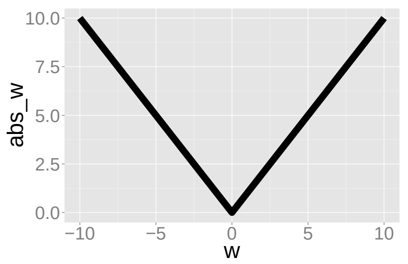
Subgradients

$$w = \frac{\sum_{i=1}^{n} y_i x_i - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\sum_{i=1}^{n} x_i^2} = \frac{\mathbf{y}^\top \mathbf{x} - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\mathbf{x}^\top \mathbf{x}} : \quad \text{What is } \frac{\mathrm{d}|w|}{\mathrm{d}w}?$$



Subgradients

$$w = \frac{\sum_{i=1}^{n} y_i x_i - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\sum_{i=1}^{n} x_i^2} = \frac{\mathbf{y}^\top \mathbf{x} - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\mathbf{x}^\top \mathbf{x}} : \quad \text{What is } \frac{\mathrm{d}|w|}{\mathrm{d}w}?$$



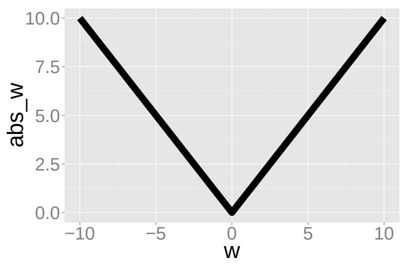
$$w > 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} = 1$$

$$w < 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} = -1$$

$$w = 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} \in (-1, 1)$$

Subgradients

$$w = \frac{\sum_{i=1}^{n} y_i x_i - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\sum_{i=1}^{n} x_i^2} = \frac{\mathbf{y}^\top \mathbf{x} - \lambda \frac{\mathrm{d}|w|}{\mathrm{d}w}}{\mathbf{x}^\top \mathbf{x}} : \quad \text{What is } \frac{\mathrm{d}|w|}{\mathrm{d}w}?$$



$$w > 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} = 1$$

$$w < 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} = -1$$

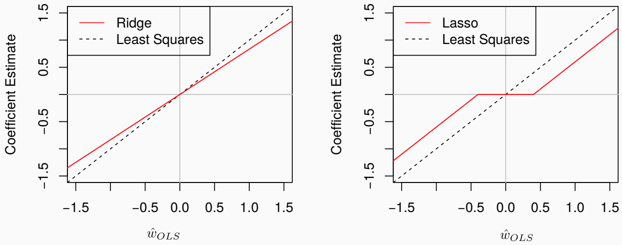
$$w = 0 \quad \leftrightarrow \frac{\mathrm{d}|w|}{\mathrm{d}w} \in (-1, 1)$$

$$w > 0 \quad \leftrightarrow w = \frac{\mathbf{y}^\top \mathbf{x} - \lambda}{\mathbf{x}^\top \mathbf{x}}$$

$$w < 0 \quad \leftrightarrow w = \frac{\mathbf{y}^\top \mathbf{x} + \lambda}{\mathbf{x}^\top \mathbf{x}}$$

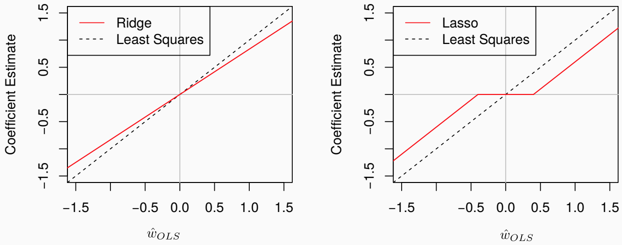
$$w = 0 \quad \leftrightarrow w = \text{otherwise}$$

The 1-D case



LASSO

First calculate: $\hat{w}_{ols} = \frac{\mathbf{y}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$



LASSO

First calculate: $\hat{w}_{ols} = \frac{\mathbf{y}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$

Soft threshold: $\hat{w}_{LASSO} = \text{sign}(\hat{w}_{ols})(|\hat{w}_{ols}| - \frac{\lambda}{\mathbf{x}^\top \mathbf{x}})_+$

$(x)_+ = x \; if \; x > 0, else \; 0$, and

$\text{sign}(x) = +1 \; if \; x > 0 \; else \; -1$

Lasso in higher (p) dimensions

Find $\mathbf{w}$ by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda\|\mathbf{w}\|_1 \tag{1}$$

(3)

Find $\mathbf{w}$ by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \quad (1)$$

$$= \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} w_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |w_j| \quad (2)$$

$$\quad (3)$$

Find $\mathbf{w}$ by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \tag{1}$$

$$= \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} w_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2}$$

$$= \sum_{i=1}^{n} (r_{id} - w_d x_{id})^2 + \lambda |w_d| + C \tag{3}$$

Find $\mathbf{w}$ by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \tag{1}$$

$$= \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} w_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2}$$

$$= \sum_{i=1}^{n}(r_{id} - w_d x_{id})^2 + \lambda |w_d| + C \tag{3}$$

Lasso in higher (p) dimensions

Find $\mathbf{w}$ by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda\|\mathbf{w}\|_1 \tag{1}$$

$$= \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} w_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2}$$

$$= \sum_{i=1}^{n}(r_{id} - w_d x_{id})^2 + \lambda |w_d| + C \tag{3}$$

Here r_{id} is the residual of obs. i:

$$r_{id} = y_i - \sum_{j \neq d} w_j x_{ij}$$

LASSO IN HIGHER (P) DIMENSIONS

Find **w** by coordinate descent

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \tag{1}$$

$$= \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} w_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{2}$$

$$= \sum_{i=1}^{n} (r_{id} - w_d x_{id})^2 + \lambda |w_d| + C \tag{3}$$

Here r_{id} is the residual of obs. i:

$$r_{id} = y_i - \sum_{j \neq d} w_j x_{ij}$$

Eq(3) is just 1d LASSO! Can solve for w_d by soft-thresholding.

Repeat

Initialize $\mathbf{w}$ to some arbitrary value

For dimension d, calculate the residual $\mathbf{r}_d = (r_{1d}, \cdots, r_{nd})$, $r_{id} = y_i - \sum_{j \neq d} w_j x_{ij}$ for each observation i

Set $\hat{w}_{ols} = \frac{(\mathbf{x}_d)^\top \mathbf{r}_d}{(\mathbf{x}_d)^\top \mathbf{x}_d}$ where $\mathbf{x}_d$ is the dth column of $\mathbf{X}$ and we have:

$$\hat{w}_d = \text{sign}(\hat{w}_{ols})(|\hat{w}_{ols}| - \frac{\lambda}{(\mathbf{x}_d)^\top \mathbf{x}_d})_+$$

Initialize $\mathbf{w}$ to some arbitrary value

For dimension d, calculate the residual $\mathbf{r}_d = (r_{1d}, \cdots, r_{nd})$, $r_{id} = y_i - \sum_{j \neq d} w_j x_{ij}$ for each observation i

Set $\hat{w}_{ols} = \frac{(\mathbf{x}_d)^\top \mathbf{r}_d}{(\mathbf{x}_d)^\top \mathbf{x}_d}$ where $\mathbf{x}_d$ is the dth column of $\mathbf{X}$ and we have:

$$\hat{w}_d = \text{sign}(\hat{w}_{ols})(|\hat{w}_{ols}| - \frac{\lambda}{(\mathbf{x}_d)^\top \mathbf{x}_d})_+$$

Repeat across dimensions d till convergence.

Co-ordinate descent

Initialize $\mathbf{w}$ to some arbitrary value

For dimension d, calculate the residual $\mathbf{r}_d = (r_{1d}, \cdots, r_{nd})$, $r_{id} = y_i - \sum_{j \neq d} w_j x_{ij}$ for each observation i

Set $\hat{w}_{ols} = \frac{(\mathbf{x}_d)^\top \mathbf{r}_d}{(\mathbf{x}_d)^\top \mathbf{x}_d}$ where $\mathbf{x}_d$ is the dth column of $\mathbf{X}$ and we have:

$$\hat{w}_d = \text{sign}(\hat{w}_{ols})(|\hat{w}_{ols}| - \frac{\lambda}{(\mathbf{x}_d)^\top \mathbf{x}_d})_+$$

Repeat across dimensions d till convergence.

Does this work?

Does co-ordinate descent work?

For convex differentiable functions: yes

Convex function f: local optimum is a global minimum.

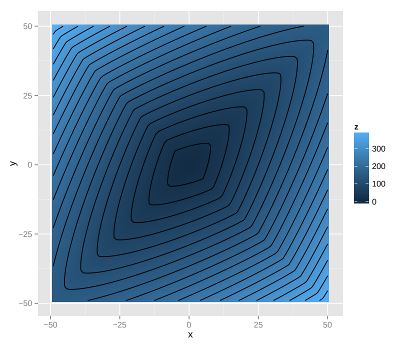
Local optimum for a differentiable function:

$$\nabla f(\mathsf{w}) = \left[\frac{\partial f}{\partial w_1}, \cdots, \frac{\partial f}{\partial w_p}\right] = 0$$

At a stationary point of coordinate descent, the RHS is true.

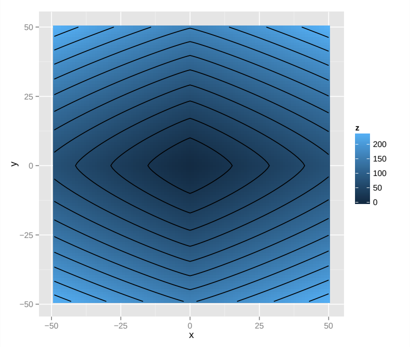
Does co-ordinate descent work?

For convex non-differentiable functions: in general, no!



Does co-ordinate descent work?

For functions of the form: $f(\mathbf{w}) = g(\mathbf{w}) + \sum_{i=1}^{p} h_i(w_i)$, where f is convex and differentiable, h_i's are convex but not differentiable: yes



Co-ordinate descent

Competitive with state-of-the-art optimization for LASSO

Co-ordinate descent

Competitive with state-of-the-art optimization for LASSO

Since objective function is convex, any initialization works (though some are better)

Co-ordinate descent

Competitive with state-of-the-art optimization for LASSO

Since objective function is convex, any initialization works (though some are better)

Obtains the solution $\hat{w}$ for any λ

Competitive with state-of-the-art optimization for LASSO

Since objective function is convex, any initialization works (though some are better)

Obtains the solution $\hat{\mathrm{w}}$ for any λ

Can repeat for different λ's (though some ways are better).

Pathwise co-ordinate descent

We want $\hat{\mathbf{w}}$'s for a set of λ's

Pathwise co-ordinate descent

We want $\hat{w}$'s for a set of λ's

Pick a smallest and largest λ (latter corresponding to $\hat{w} = 0$)
Divide into equidistant grid points (typ. on logscale)

Pathwise co-ordinate descent

We want $\hat{\mathbf{w}}$'s for a set of λ's

Pick a smallest and largest λ (latter corresponding to $\hat{\mathbf{w}} = 0$)
Divide into equidistant grid points (typ. on logscale)

Start with the largest λ (solution = 0).

Pathwise co-ordinate descent

We want $\hat{\mathbf{w}}$'s for a set of λ's

Pick a smallest and largest λ (latter corresponding to $\hat{\mathbf{w}} = 0$)
Divide into equidistant grid points (typ. on logscale)

Start with the largest λ (solution = 0).

Move to the next, using previous solution as initialization.

Pathwise co-ordinate descent

We want $\hat{\mathbf{w}}$'s for a set of λ's

Pick a smallest and largest λ (latter corresponding to $\hat{\mathbf{w}} = 0$)
Divide into equidistant grid points (typ. on logscale)

Start with the largest λ (solution = 0).

Move to the next, using previous solution as initialization.

Converges after a few sweeps

Repeat

Pathwise co-ordinate descent

We want $\hat{\mathbf{w}}$'s for a set of λ's

Pick a smallest and largest λ (latter corresponding to $\hat{\mathbf{w}} = 0$)
Divide into equidistant grid points (typ. on logscale)

Start with the largest λ (solution = 0).

Move to the next, using previous solution as initialization.

Converges after a few sweeps

Repeat

This kind of a guided search is often faster, even if we just want one λ.