Gradient descent $\leftarrow$ linear regression, ridge regression $\rightarrow$ closed form solution

coordinate descent $\leftarrow$ lasso regression, logistic regression $\rightarrow$ optimization

Gradient Descent

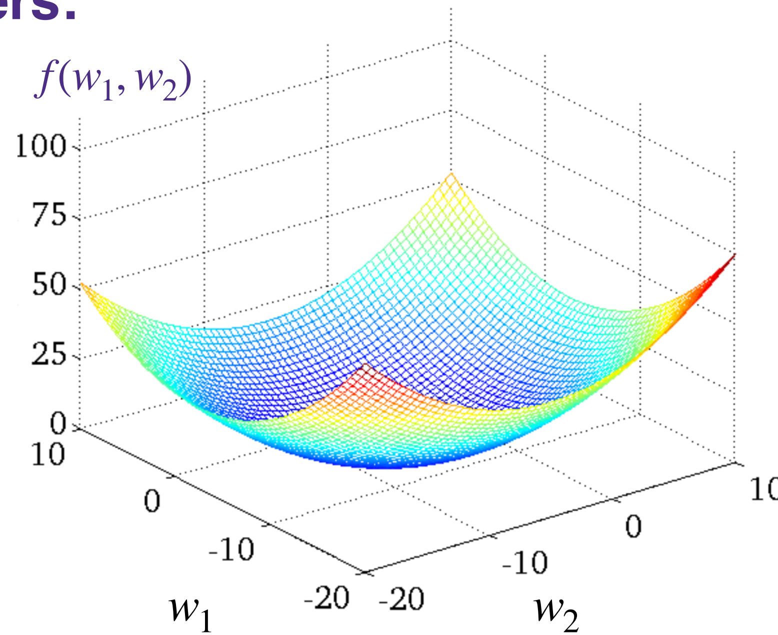
Running example: linear regression

- **Given data:**

$$\{(x_i, y_i)\}_{i=1}^{n} \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- **Learning a model's parameters:**

$$\hat{w}_{\mathrm{LS}} = \arg\min_{w \in \mathbb{R}^d} \underbrace{\|\mathbf{y} - \mathbf{X}w\|_2^2}_{f(w)}$$



Gradient descent

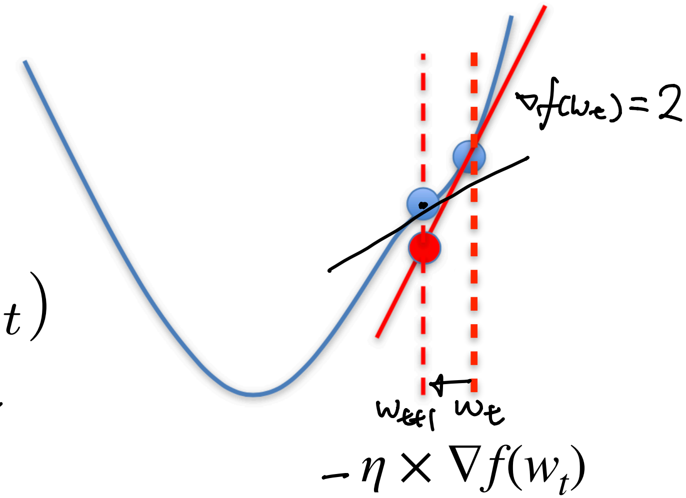
Initialize: $w_0 = 0$

for $t = 1, 2, \ldots$

$$w_{t+1} = \underset{\mathbb{R}^d}{w_t} - \underset{\mathbb{R}}{\eta} \underset{\mathbb{R}^d}{\nabla f(w_t)}$$

Learning rate or step-size,
a hyper-parameter to be chosen by the analyst

Example of a general non-convex $f(w)$



Gradient descent for linear regression

Initialize: $w_0 = 0$

for $t = 1, 2, \ldots$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

For linear regression, we have

$$\hat{w}_{LS} = \arg\min_{w \in \mathbb{R}^d} \underbrace{\|\mathbf{y} - \mathbf{X}w\|_2^2}_{f(w)}$$

$$w^* = (X^T X)^{-1} X^T y$$

$$\nabla f(w_t) = -2 \cdot X^T (y - X w_t)$$

$$w_{t+1} = w_t + 2\eta X^T (y - X w_t),$$

$$w_{t+1} - w^* = (\mathbb{I} - 2\eta X^T X) w_t + 2\eta X^T y - w^*$$

$$= (\mathbb{I} - 2\eta X^T X)(w_t - w^*) + \underbrace{2\eta X^T y - 2\eta \underbrace{X^T X \cdot w^*}_{2\eta \cancel{(X^T X)} \cancel{(X^T X)^{-1}} X^T y}}_{=0}$$

$$w_{t+1} - w^* = \underbrace{(\mathbb{I} - 2\eta X^T X)}_{A} (w_t - w^*)$$

Contraction: $\|A\|_{spectral} < 1$,

Gradient descent for linear regression

Initialize: $w_0 = 0$

for $t = 1, 2, \ldots$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

For linear regression, we have

$$\hat{w}_{\text{LS}} = \arg\min_{w \in \mathbb{R}^d} \underbrace{\|\mathbf{y} - \mathbf{X}w\|_2^2}_{f(w)}$$

$$\nabla f(w_t) = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}w_t)$$

$$w_{t+1} = w_t + \eta 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}w_t) = (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})w_t + 2\eta\mathbf{X}^T\mathbf{y}$$

Let the least-squares solution be $w^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

$$\begin{aligned} w_{t+1} - w^* &= (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})w_t + 2\eta\mathbf{X}^T\mathbf{y} - w^* \\ &= (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})(w_t - w^*) + 2\eta\mathbf{X}^T\mathbf{y} - 2\eta\mathbf{X}^T\mathbf{X}w^* \\ &= (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})(w_t - w^*) \end{aligned}$$

Gradient descent for linear regression

$$w_{t+1} = w_t - \eta \nabla f(w_t) \implies w_{t+1} - w^* = (\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X})(w_t - w^*)$$

$$= (I - 2\eta X^T X)(I - 2\eta X^T X)(w_{t-1} - w^*)$$

$$\vdots$$

$$= \underbrace{(I - 2\eta X^T X)^{t+1}} \underline{(w_0 - w^*)}$$

Gradient descent for linear regression

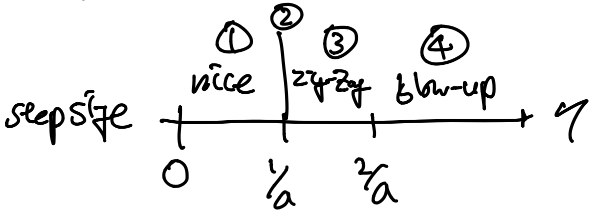
$$w_{t+1} = w_t - \eta \nabla f(w_t) \implies w_{t+1} - w^* = (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})(w_t - w^*)$$

$$= (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})^2(w_{t-1} - w^*)$$

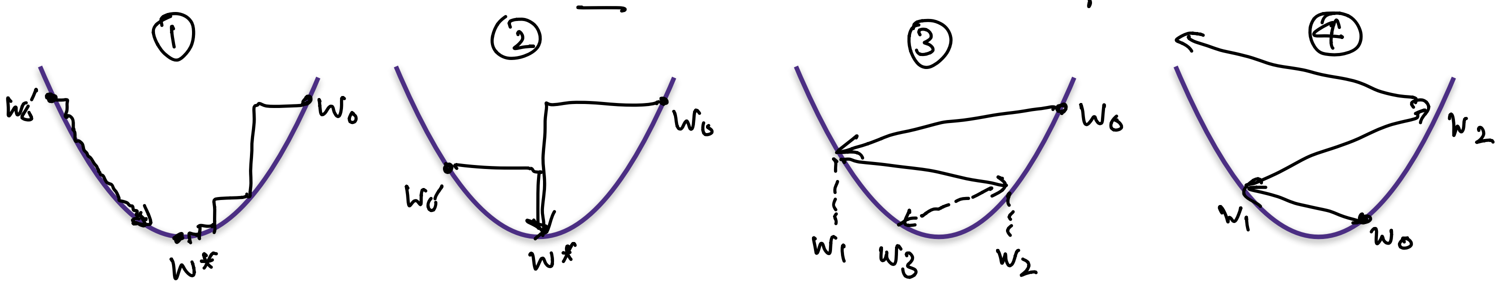
$$\vdots$$

$$= (\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})^{t+1}(w_0 - w^*)$$

$$= \underbrace{(1 - \eta \cdot a)}^{t+1}(w_0 - w^*)$$



In one dimension, $2\mathbf{X}^T\mathbf{X} = \underline{\underline{a}}$ $\in \mathbb{R}$ is a scalar



$0 < \eta < 1/a$ | $\eta = 1/a$ | $1/a < \eta < 2/a$ | $\eta > 2/a$

stepsize

$0 < \boxed{1 - \eta \cdot a} < 1$

$= 1/2$

$\underline{1 - \eta a = 0}$

$-1 < 1 - \eta a < 0$

$1 - \eta a < -1$

Gradient descent for linear regression

$$w_{t+1} = w_t - \eta \nabla f(w_t) \quad \Longrightarrow \quad w_{t+1} - w^* = (\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X})^{t+1}(w_0 - w^*)$$

In multi dimensions, **eigenvalues** of $\mathbf{X}^T\mathbf{X}$ are important

(you will see why I say $\mathbf{X}^T\mathbf{X}$ instead of $\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}$ in couple of slides)

Let the eigenvalue decomposition of $\underline{\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}}$ be $Q^{-1}DQ$

$$w_{t+1} - w^* = Q^{-1}D\cancel{Q} \times \cancel{Q^{-1}}D\cancel{Q} \times \cdots \times \cancel{Q^{-1}}DQ \cdot (w_0 - w^*)$$

$$= Q^{-1} \cdot D^{t+1} \cdot Q(w_0 - w^*)$$

$$D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_d \end{bmatrix}, \quad Q = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \end{bmatrix}$$

Right Eigen Vector

Eigen Values

$$Q(w_{t+1} - w^*) = D^{t+1} \cdot Q(w_0 - w^*)$$

$$\begin{bmatrix} q_1^T(w_{t+1} - w^*) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \lambda_1^{t+1} & & & \\ & \lambda_2^{t+1} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} q_1^T(w_0 - w^*) \\ q_2^T(w_0 - w^*) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \boxed{\lambda_1^{t+1} \cdot} \, q_1^T(w_0 - w^*) \\ \vdots \\ \vdots \end{bmatrix}$$

Project onto q_i direction

Gradient descent for linear regression

$$w_{t+1} = w_t - \eta \nabla f(w_t) \quad \Longrightarrow \quad w_{t+1} - w^* \; = \; (\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X})^{t+1}(w_0 - w^*)$$

In multi dimensions, **eigenvalues** of $\mathbf{X}^T\mathbf{X}$ are important

(you will see why I say $\mathbf{X}^T\mathbf{X}$ instead of $\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}$ in couple of slides)

Let the eigenvalue decomposition of $\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}$ be $Q^{-1}DQ$

Then, $w_{t+1} - w^* \; = \; (Q^{-1}DQ)^{t+1}(w_0 - w^*)$

$$= \underbrace{Q^{-1}DQQ^{-1}DQ\cdots Q^{-1}DQ}_{t+1 \text{ times}}(w_0 - w^*)$$

$$= Q^{-1}D^{t+1}Q\,(w_0 - w^*)$$

$$Q(w_{t+1} - w^*) \; = \; D^{t+1}Q\,(w_0 - w^*)$$

$$Q(w_{t+1} - w^*) \;=\; D^{t+1} Q\,(w_0 - w^*)$$

$$\Downarrow$$

each $i \in \{1, \dots, d\}$

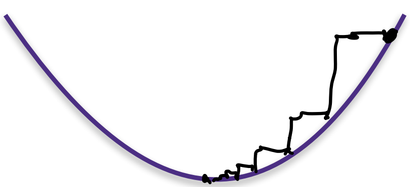
$$q_i^T(w_{t+1} - w^*) = \lambda_i^{t+1} \cdot q_i^T(w_0 - w^*)$$

$$|\lambda_i| < 1.$$

Gradient descent for linear regression

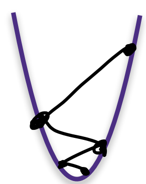
$$w_{t+1} = w_t - \eta \nabla f(w_t) \quad \Longrightarrow \quad w_{t+1} - w^* \;=\; (\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X})^{t+1}(w_0 - w^*)$$

$$Q(w_{t+1} - w^*) \;=\; D^{t+1} Q\,(w_0 - w^*)$$



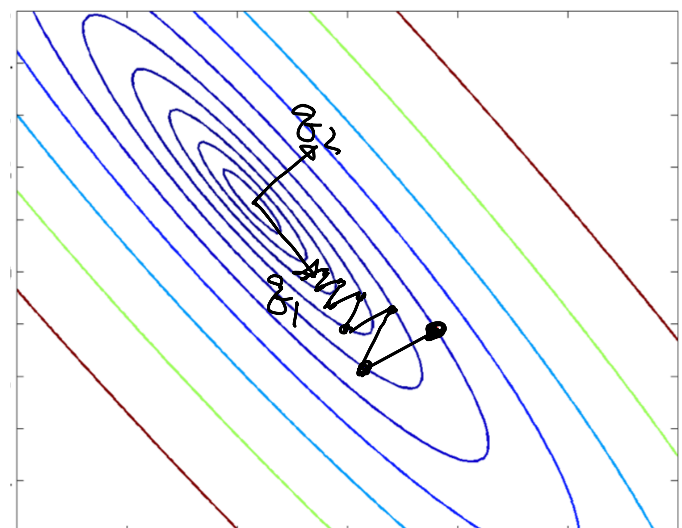
In direction q_1

$0 < \lambda_1(\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X})$



In direction q_2

$-1 < \lambda_2(\mathbf{I} - 2\eta \mathbf{X}^T\mathbf{X}) < 0$



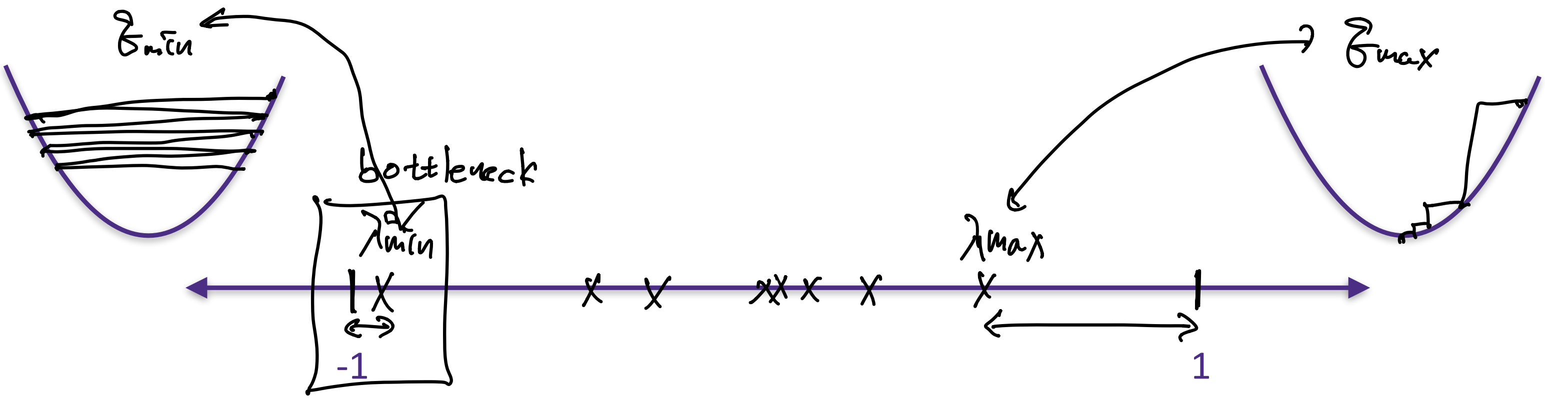
Gradient descent for linear regression

Recall that in each eigen direction

$$q_i^T(w_{t+1} - w^*) \;=\; \Big(\overbrace{\lambda_i(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})}^{\text{i-th eigen value of } \mathbf{I}-2\eta X^TX}\Big)^{t+1} q_i^T(w_0 - w^*)$$

We want the error to decay fast in all directions, whose bottleneck is the largest and the smallest eigen values:

$$\lambda_{\min}(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}) \text{ and } \lambda_{\max}(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})$$



We want to choose the learning rate η such that

what is optimal η^*?

$$-1 \ll \overset{\lambda_{min}}{\lambda_d}(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}) \le \cdots \le \overset{\lambda_{max}}{\lambda_1}(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}) \ll 1$$

$\lambda_i \leftrightarrow I - 2\eta X^TX$, $\tilde{\lambda}_i \leftrightarrow 2X^TX$ $\quad$ $q_i^T \lambda_i(X^TX)\cdot q_i = q_i^T 2X^TX q_i$

Gradient descent for linear regression

$\lambda_i(X^TX) = \frac{2 q_i^T X^T X q_i}{q_i^T q_i}$

Recall that in each eigen direction

$$\overline{q_i^T(w_{t+1} - w^*) \;=\; \lambda_i(\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X})^{t+1}\, q_i^T(w_0 - w^*)}$$

I will not prove the facts that $\|q_i\|_2 = 1$ and $q_i^T q_j = 0$

Claim: $\lambda_i(\boxed{\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X}}) = 1 - 2\eta\lambda_i(\mathbf{X}^T\mathbf{X})$ $\quad$ $2X^TX = \tilde{Q}^{-1}\tilde{D}\tilde{Q}$

d-i+1-th eigenvalue

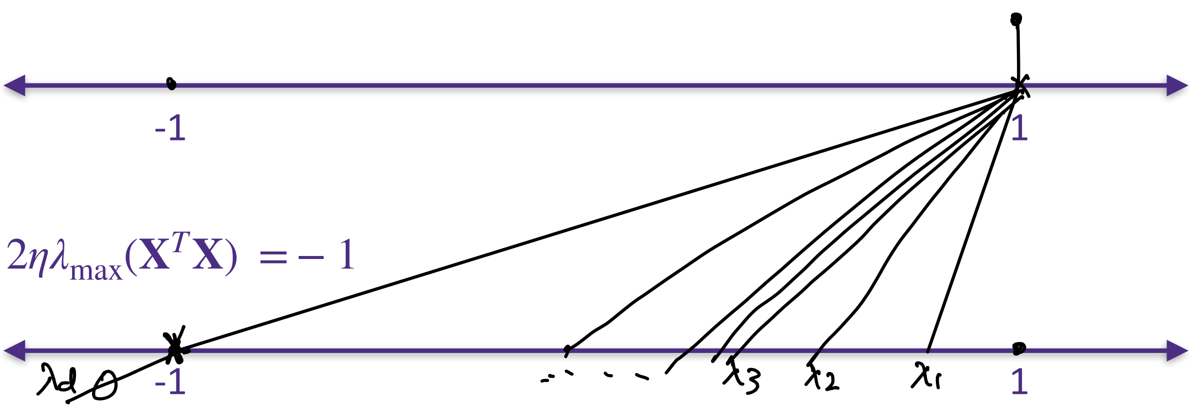
$$\mathbf{I} - 2\eta\mathbf{X}^T\mathbf{X} = \tilde{Q}^{-1}\tilde{Q} - \eta\tilde{Q}^{-1}\tilde{D}\tilde{Q}$$

$$= \tilde{Q}^{-1}\underbrace{(I - \eta\tilde{D})}_{\text{Diagonal}}\tilde{Q} \implies \begin{bmatrix} 1-\eta\tilde{\lambda}_1 & & \\ & 1-\eta\tilde{\lambda}_2 & \\ & & \ddots \end{bmatrix} \leftarrow \lambda_i(I - 2\eta X^TX)$$

$$= 1 - 2\eta\cdot\lambda_i(2X^TX)$$

Claim: $\underline{\lambda_i(\mathbf{X}^T\mathbf{X}) \geq 0}$

$\{\lambda_i(I - 2\eta X^TX)\}$

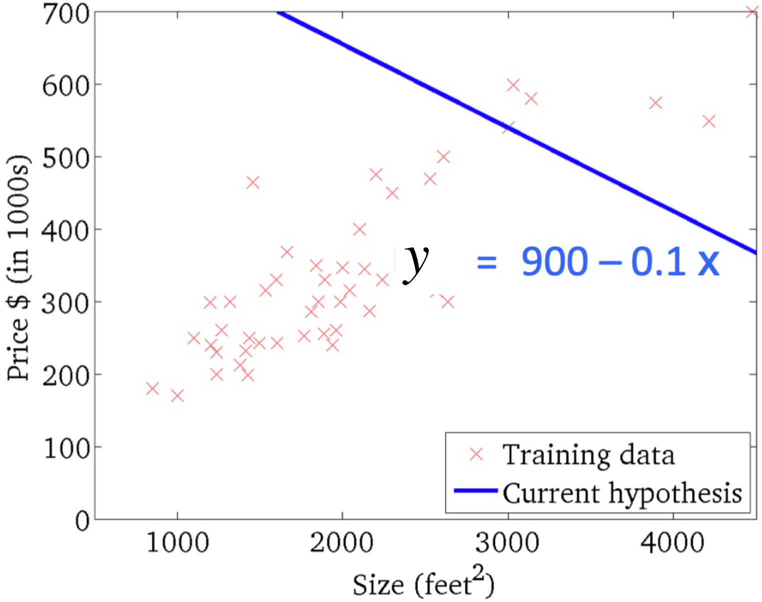


$\eta = 0$

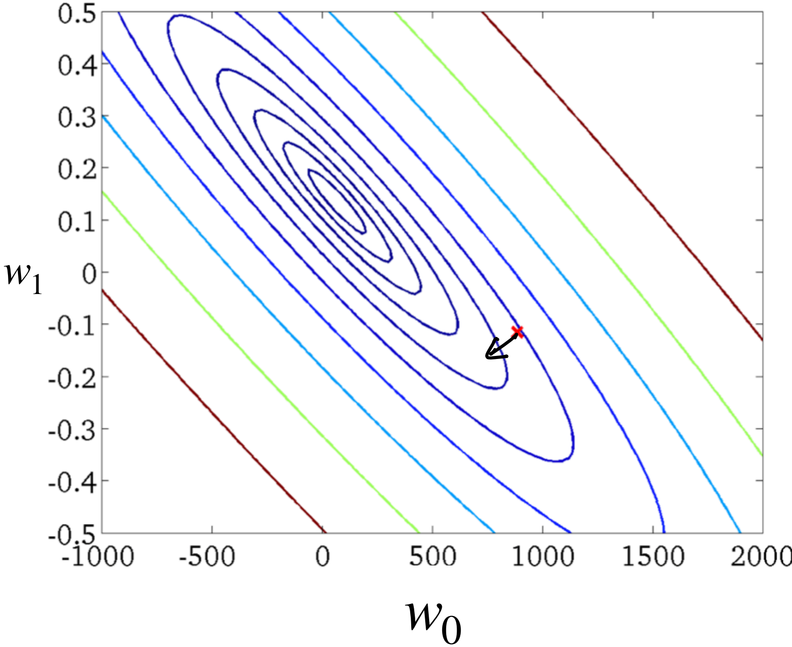
$1 - 2\eta\lambda_{\max}(\mathbf{X}^T\mathbf{X}) = -1$

$\eta = \frac{1}{\lambda_{\max}(\mathbf{X}^T\mathbf{X})}$

$$y = w_0 + w_1 x$$

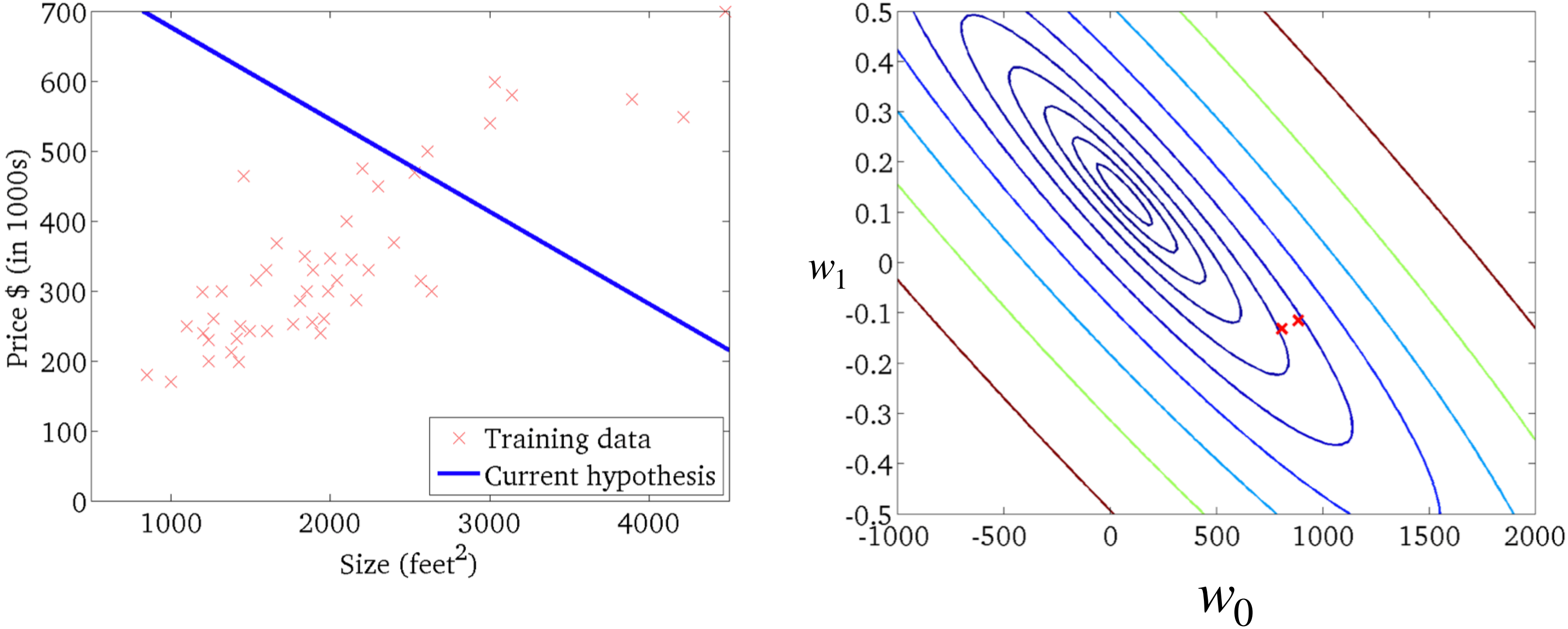


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
y = 900 – 0.1 x
Training data
Current hypothesis

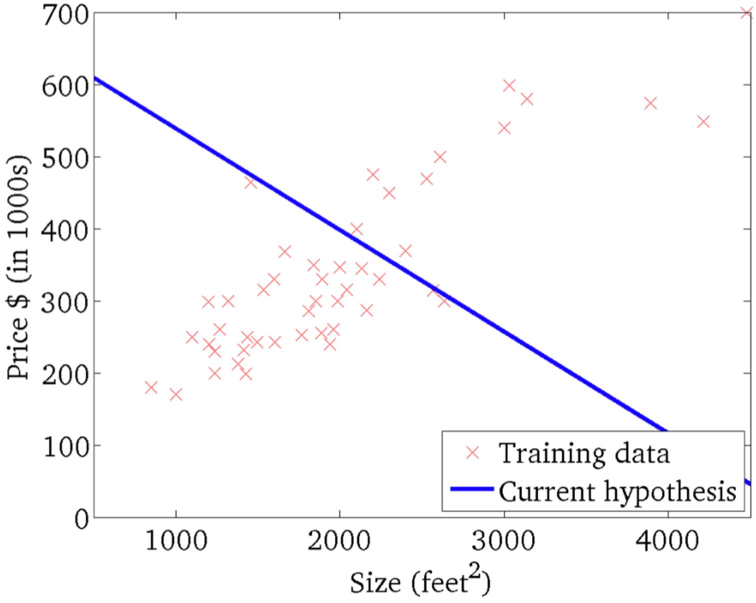


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
w_1
-1000
-500
0
500
1000
1500
2000
w_0

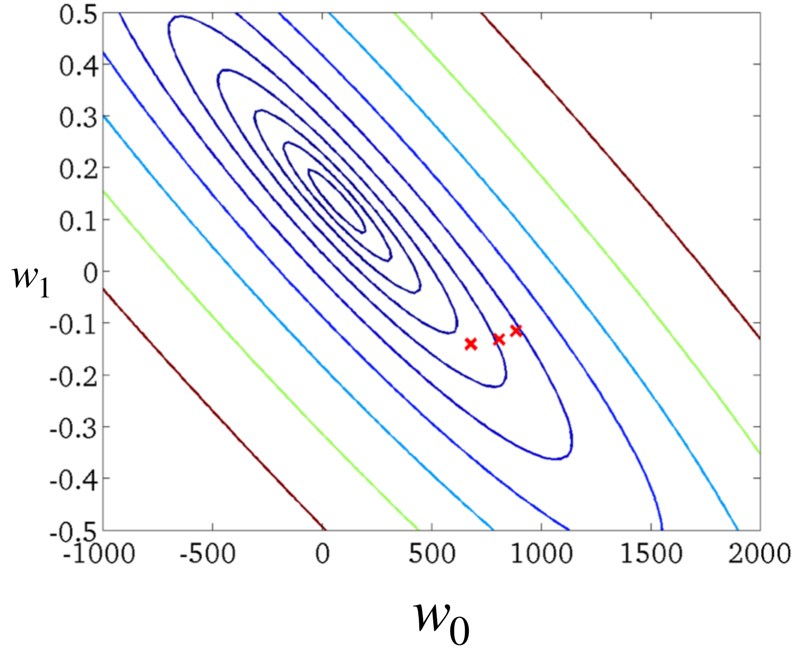
$$y = w_0 + w_1 x$$


700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
w_1
w_0

$$y = w_0 + w_1 x$$

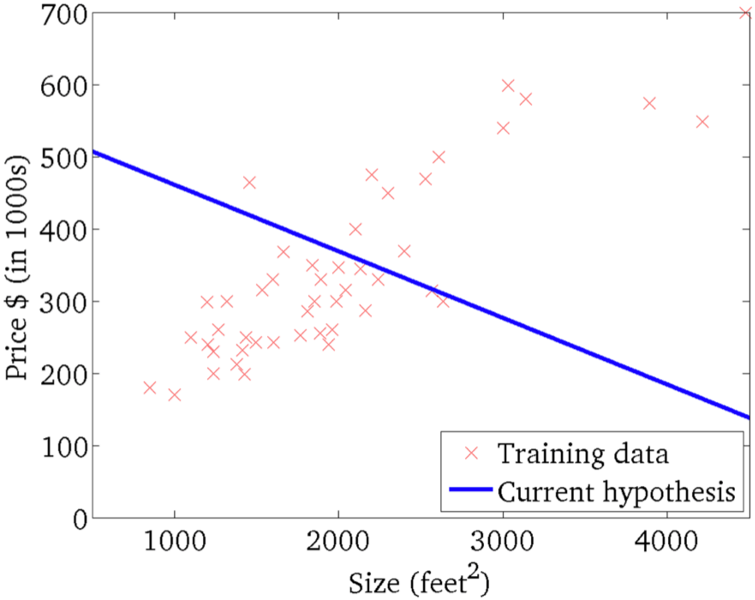


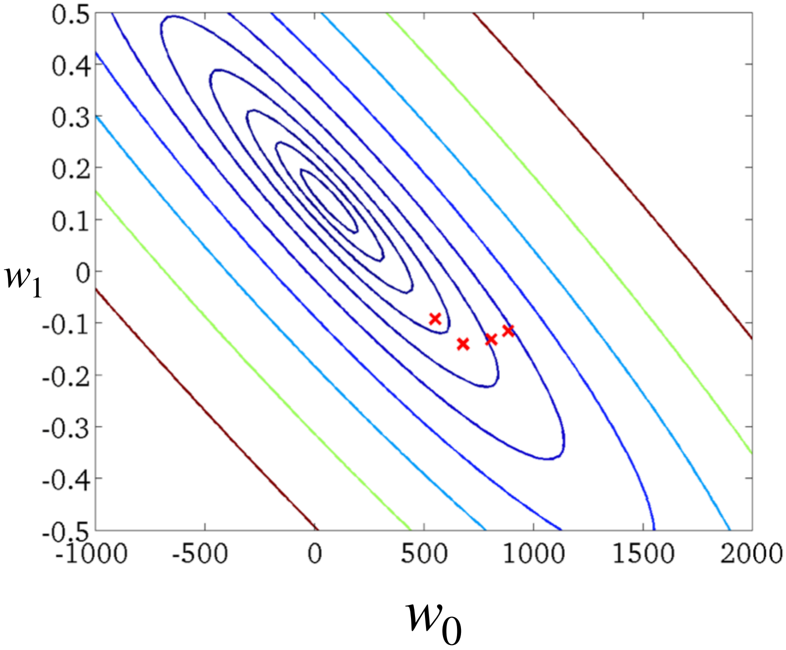
700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis



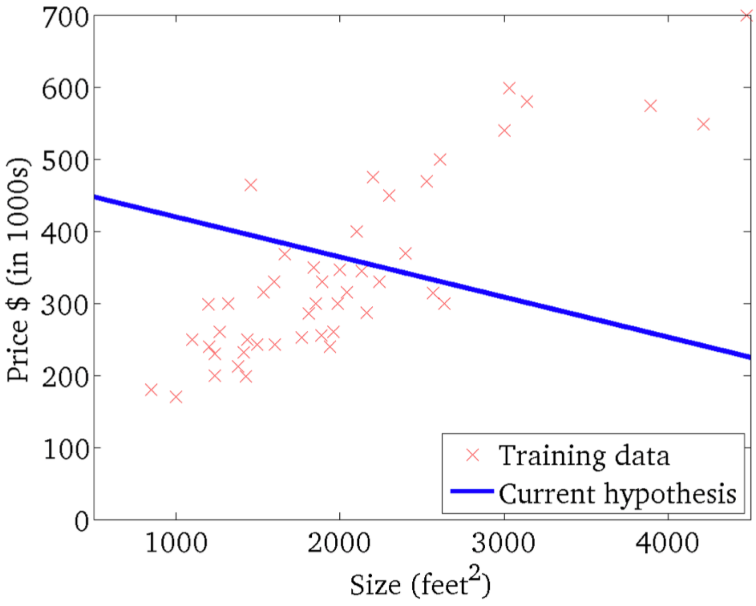
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
w_1
-1000
-500
0
500
1000
1500
2000
w_0

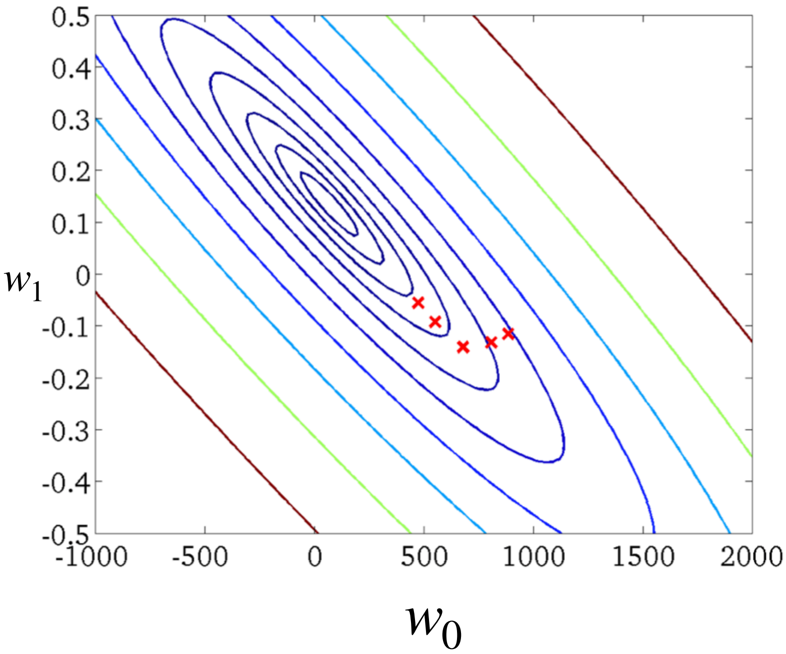
$$y = w_0 + w_1 x$$


700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis

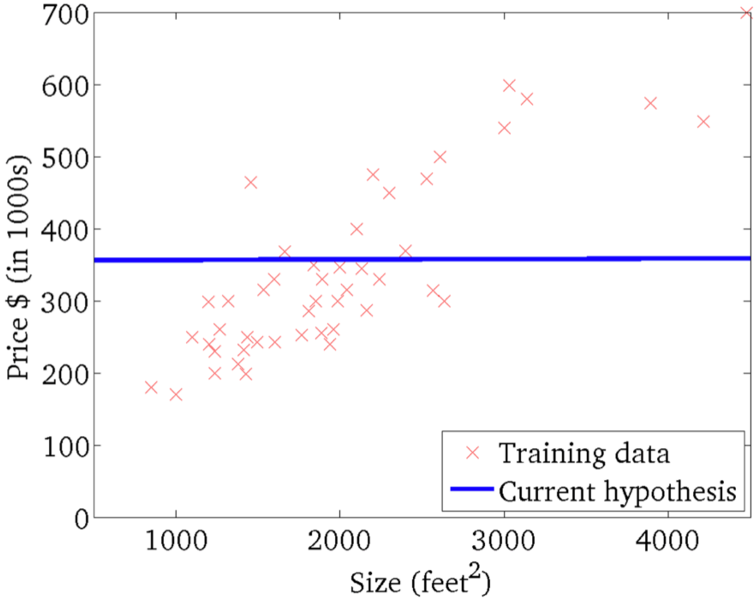

0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
w_1
w_0

$$y = w_0 + w_1 x$$

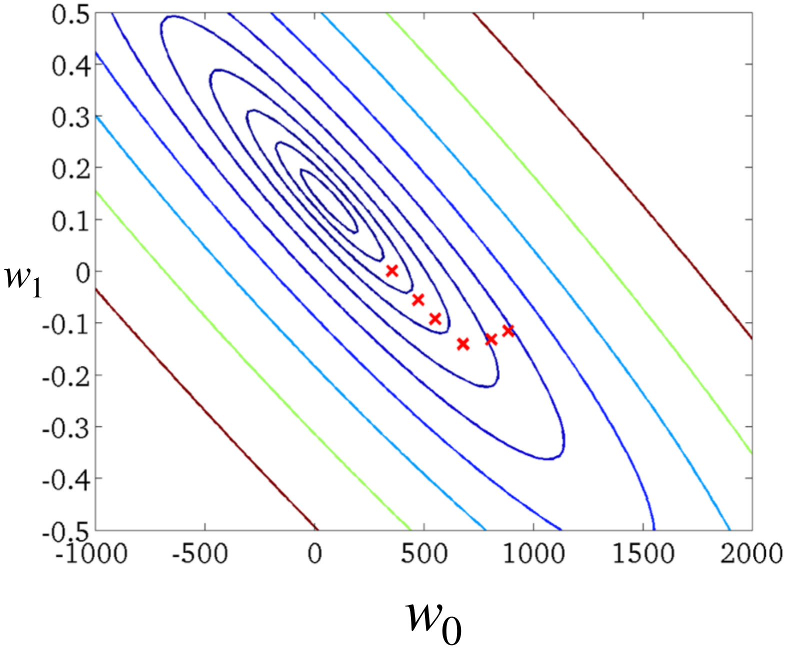

700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
w_1
w_0

$$y = w_0 + w_1 x$$

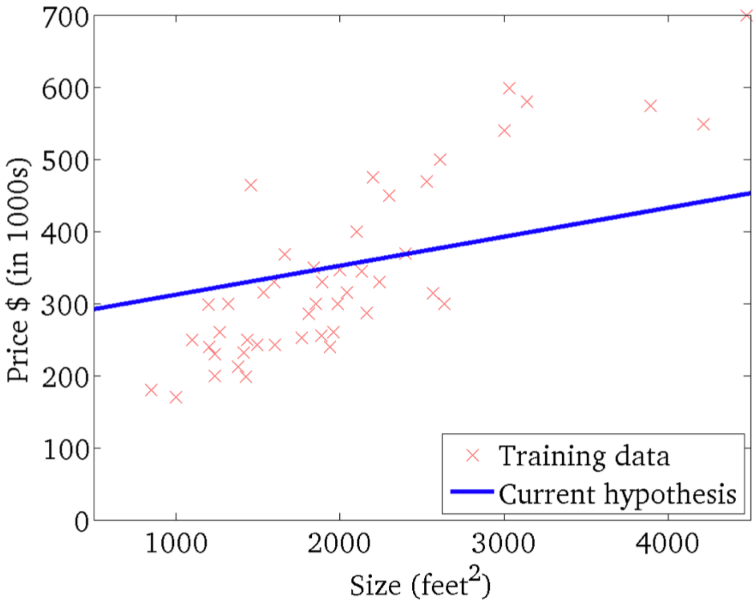


Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
700
600
500
400
300
200
100
0
1000
2000
3000
4000

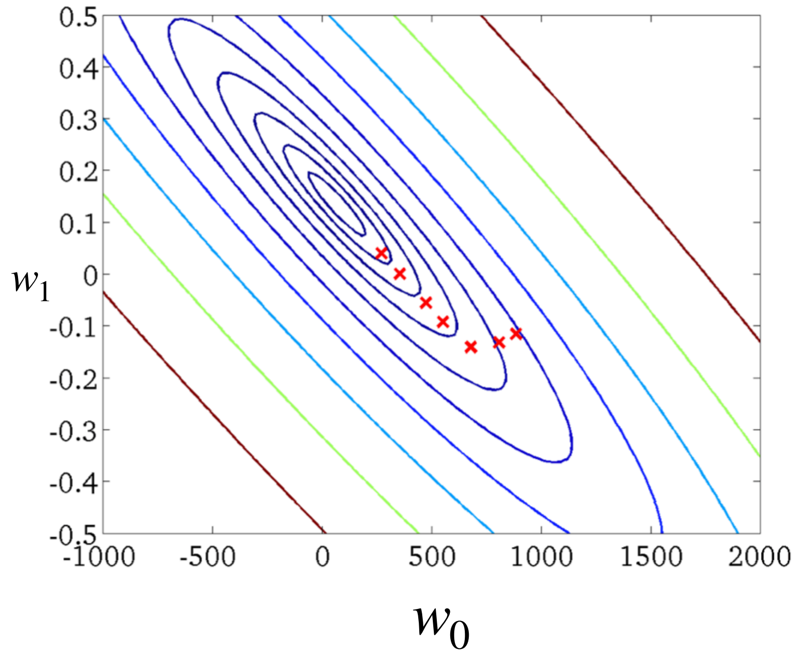


w_1
w_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000

$$y = w_0 + w_1 x$$

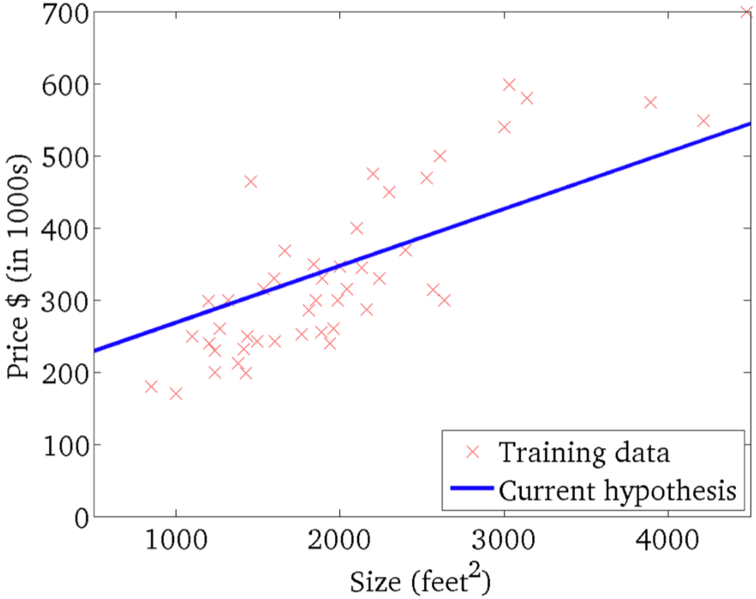


700
600
500
400
300
200
100
0
1000
2000
3000
4000
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis

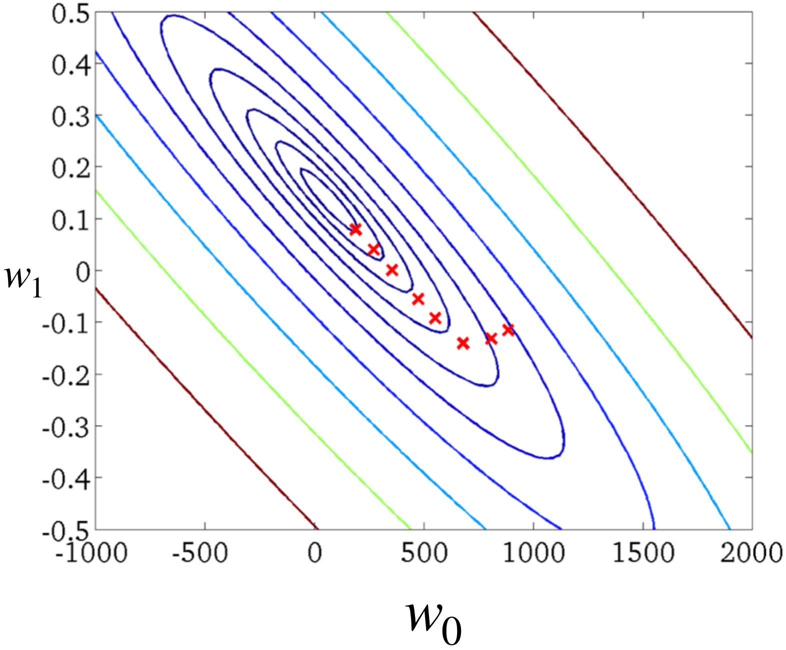


0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000
w_1
w_0

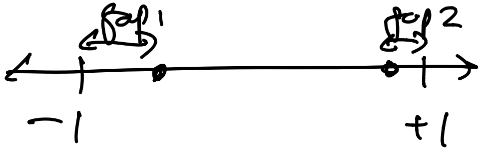
$$y = w_0 + w_1 x$$



Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
700
600
500
400
300
200
100
0
1000
2000
3000
4000



w_1
w_0
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-1000
-500
0
500
1000
1500
2000



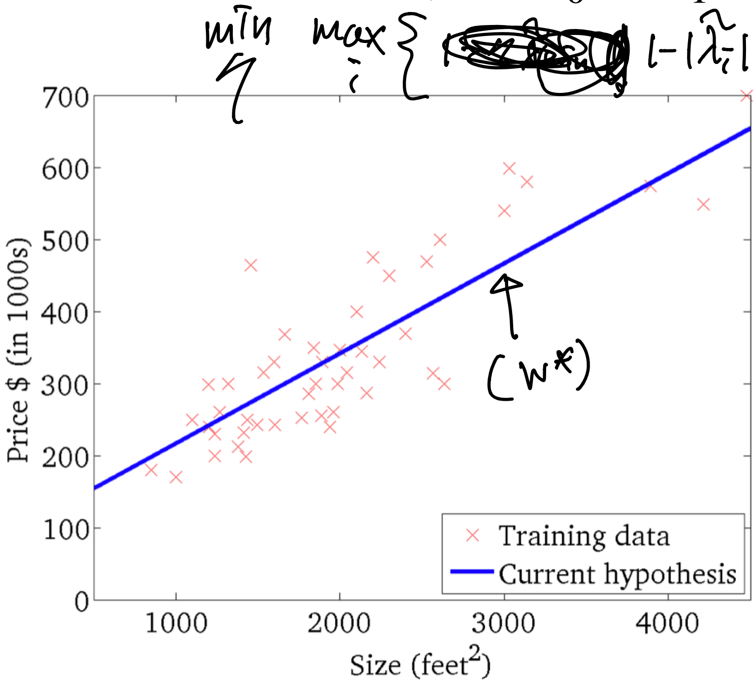
$-1 < 1-\eta\tilde{\lambda}_{max}$, $\quad 1-\eta\cdot\tilde{\lambda}_{min} < 1$

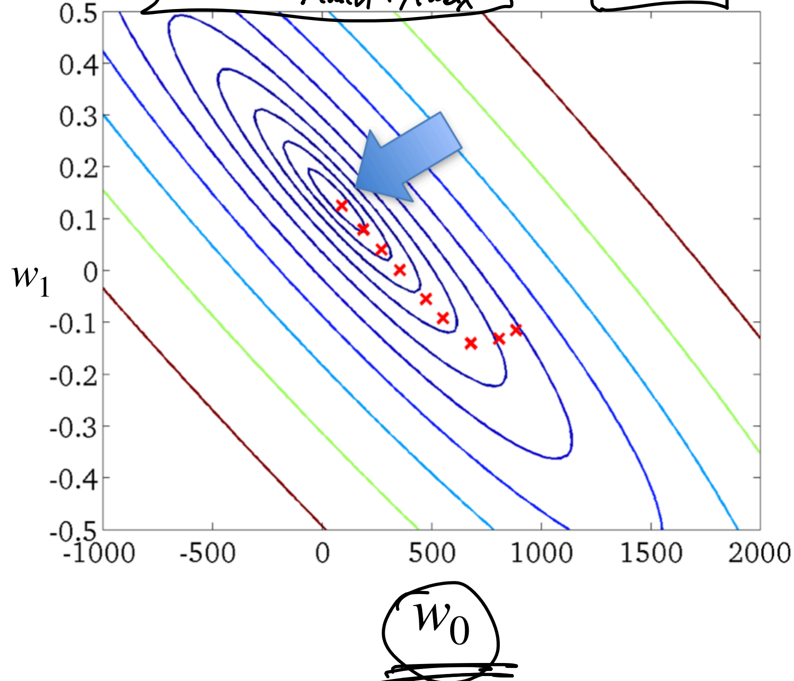
gap 1 $= 2-\eta\tilde{\lambda}_{max} =$ gap 2 $= \eta\tilde{\lambda}_{min}$

$$y = w_0 + w_1 x$$

$$\min_{\eta} \max_{i} \{ |1-\eta\tilde{\lambda}_i| \}$$

$$\boxed{\eta = \frac{2}{\tilde{\lambda}_{min}+\tilde{\lambda}_{max}}} \leftarrow \boxed{X^T X}$$





$$\boxed{\frac{\lambda_{max}}{\lambda_{min}}} = \text{condition number}$$

$$\boxed{\lambda_i(X^T X)}$$

Gradient descent for logistic regression

Loss function: Conditional Likelihood

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i|x_i, w) \qquad P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

$$= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))$$

$$\nabla f(w) = \sum_{i=1}^n \frac{1}{1 + \exp(-y_i x_i^T w)} \exp(-y_i x_i^T w)(-y_i x_i)$$

Gradient descent

- For L smooth functions, with a (small enough) fixed step size η
  - if $f(w)$ is convex,

$$f(w_t) - f(w^*) \leq \frac{\|w_0 - w^*\|_2^2}{2\eta t}$$

$\frac{1}{L}$

  - if $f(w)$ is μ-strongly convex, $\eta < \boxed{\ } < \mu$

$$f(w_t) - f(w^*) \ \leq \ \left(1 - \eta\mu\right)^t \left(f(w_0) - f(w^*)\right)$$

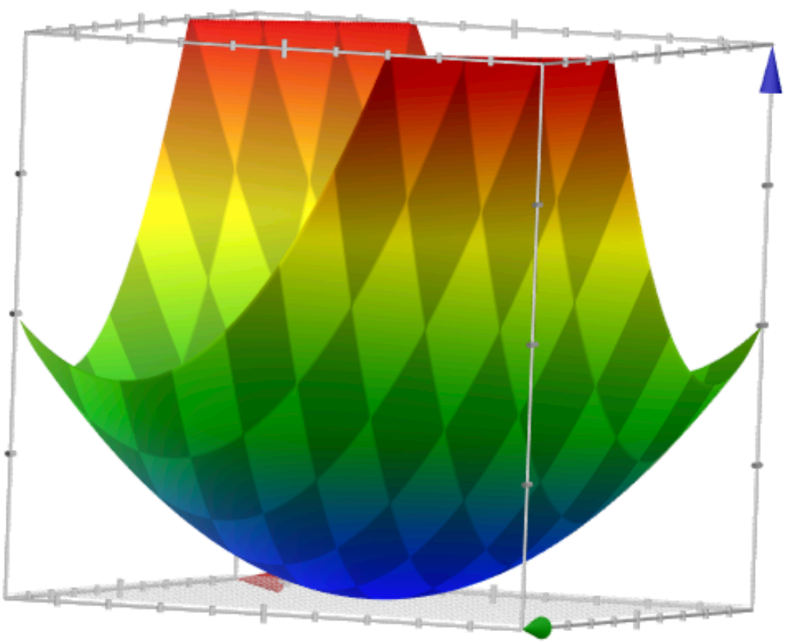
- What can we do for non-smooth function $f(w)$?
  - for example, LASSO

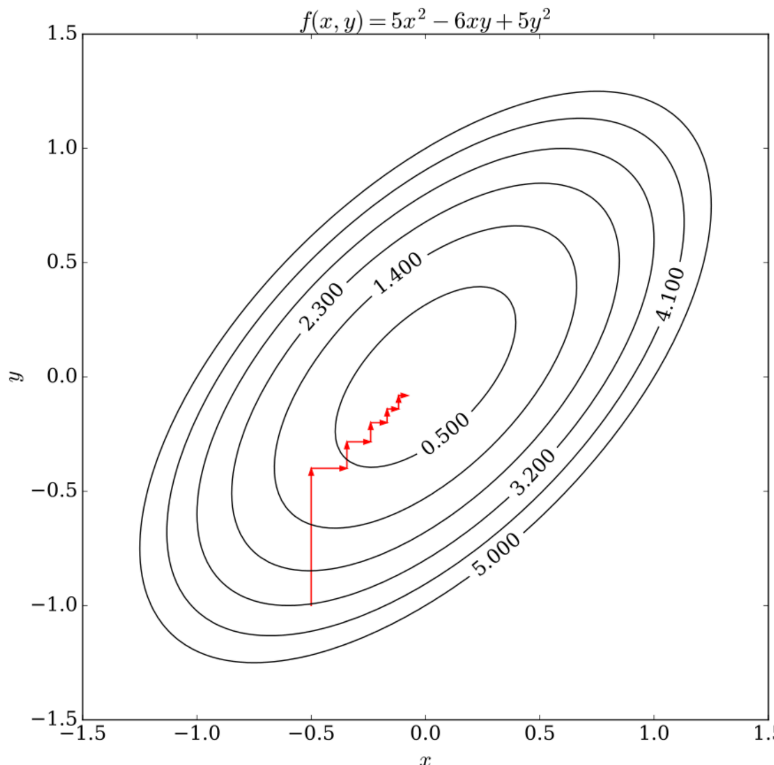
$$\hat{w}_{\text{Lasso}} = \arg\min_{w \in \mathbb{R}^d} \underbrace{\|\mathbf{y} - \mathbf{X}w\|_2^2 + \lambda\|w\|_1}_{f(w)}$$

Coordinate Descent

Optimization: how do we solve Lasso?

- among many methods to find the solution, we will learn **coordinate descent method**
- as an illustrating example, we show coordinate descent updates on finding the minimum of $f(x, y) = 5x^2 - 6xy + 5y^2$





Optimization: how do we solve Lasso?

- **Coordinate descent**
  - **input: training data S_{train}, max # of iterations T**
  - **initialize: $w^{(0)} = \mathbf{0} \in \mathbb{R}^d$**
  - **for $t = 1,\ldots,T$**
    - **for $j = 1,\ldots,d$**
      - **fix $w_1^{(t)},\ldots,w_{j-1}^{(t)}$ and $w_{j+1}^{(t-1)},\ldots,w_d^{(t-1)}$, and**

$$w_j^{(t)} \leftarrow \arg\min_{w_j\in\mathbb{R}} \mathcal{L}\left(\begin{bmatrix} w_1^{(t)} \\ \vdots \\ w_{j-1}^{(t)} \\ w_j \\ w_{j+1}^{(t-1)} \\ \vdots \\ w_d^{(t-1)} \end{bmatrix}\right) + \lambda \left\| \begin{bmatrix} w_1^{(t)} \\ \vdots \\ w_{j-1}^{(t)} \\ w_j \\ w_{j+1}^{(t-1)} \\ \vdots \\ w_d^{(t-1)} \end{bmatrix} \right\|_1$$

this is a one-dimensional optimization, which is much easier to solve

Coordinate descent for (un-regularized) linear regression

- let us understand what coordinate descent does on a simpler problem of linear least squares, which minimizes
$$\text{minimize}_w \mathcal{L}(w) = \|\mathbf{X}w - \mathbf{y}\|_2^2$$

- note that we know that the optimal solution is
$$\hat{w}_{\text{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$
so we do not need to run any optimization algorithm

- we are solving this problem with coordinate descent for illustration purpose

- the main challenge we want to address is, how do we update $w_j^{(t)}$?

- let us derive an **analytical rule** for updating $w_j^{(t)}$

Coordinate descent for (un-regularized) linear regression

Coordinate descent for (un-regularized) linear regression

- we will study the case $j = 1$, for now (other cases are almost identical)
- when updating $w_1^{(t)}$, recall that

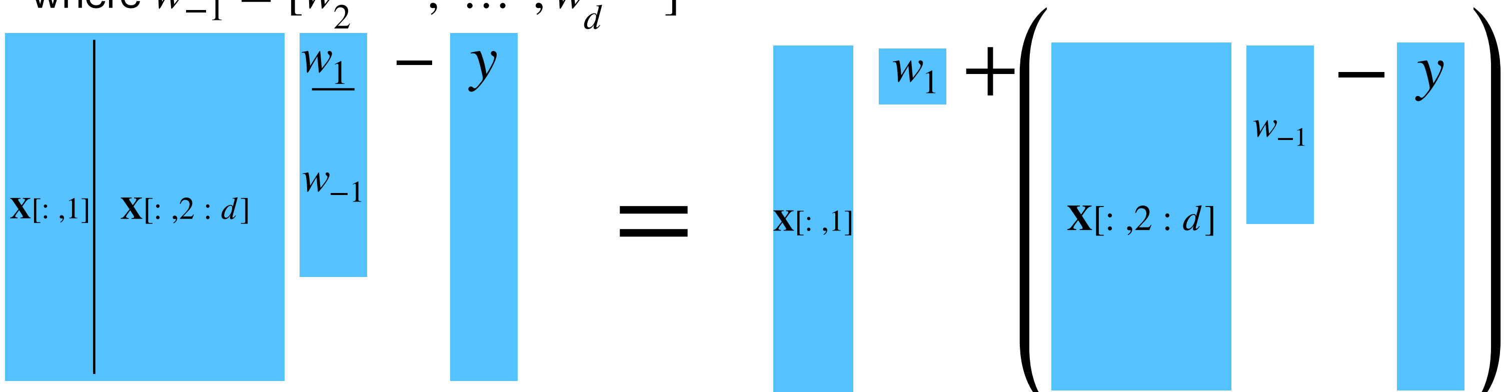
$$w_1^{(t)} \leftarrow \arg\min_{w_1} \|\mathbf{X}w - \mathbf{y}\|_2^2$$

where $w = [w_1,\ w_2^{(t-1)},\ \dots\ , w_d^{(t-1)}]^T$

- first step is to write the objective function in terms of the variable we are optimizing over, that is w_1:

$$\mathscr{L}(w) = \left\| \mathbf{X}[:,1]w_1 + \mathbf{X}[:,2:d]w_{-1} - \mathbf{y} \right\|_2^2$$

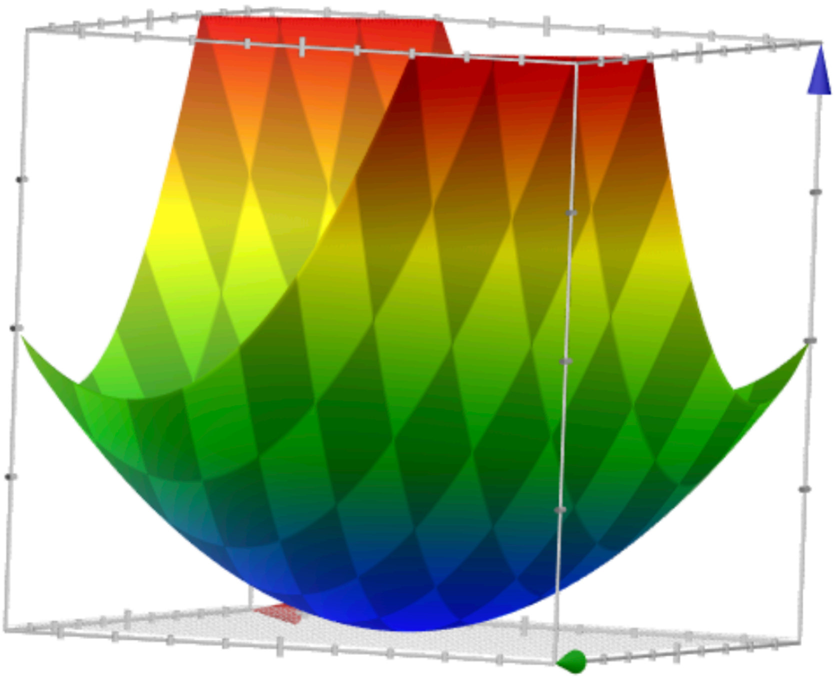
where $w_{-1} = [w_2^{(t-1)},\ \dots\ , w_d^{(t-1)}]^T$

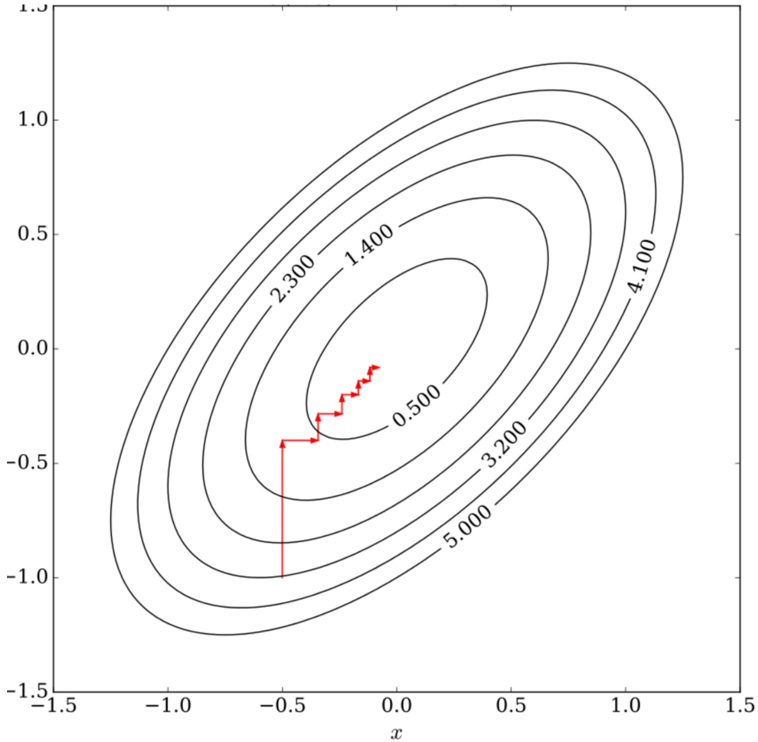


- we know from linear least squares that the minimizer is

$$w_1^{(t)} \leftarrow (\mathbf{X}[:,1]^T\mathbf{X}[:,1])^{-1}\mathbf{X}[:,1]^T\left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right)$$

- Coordinate descent applied to a quadratic loss





Coordinate descent for Lasso

- let us apply coordinate descent on Lasso, which minimizes
$$\text{minimize}_w \mathscr{L}(w) + \lambda \|w\|_1 = \|\mathbf{X}w - \mathbf{y}\|_2^2 + \lambda \|w\|_1$$

- the goal is to derive an **analytical rule** for updating $w_j^{(t)}$'s

- let us first write the update rule explicitly for $w_1^{(t)}$
  - first step is to write the loss in terms of w_1

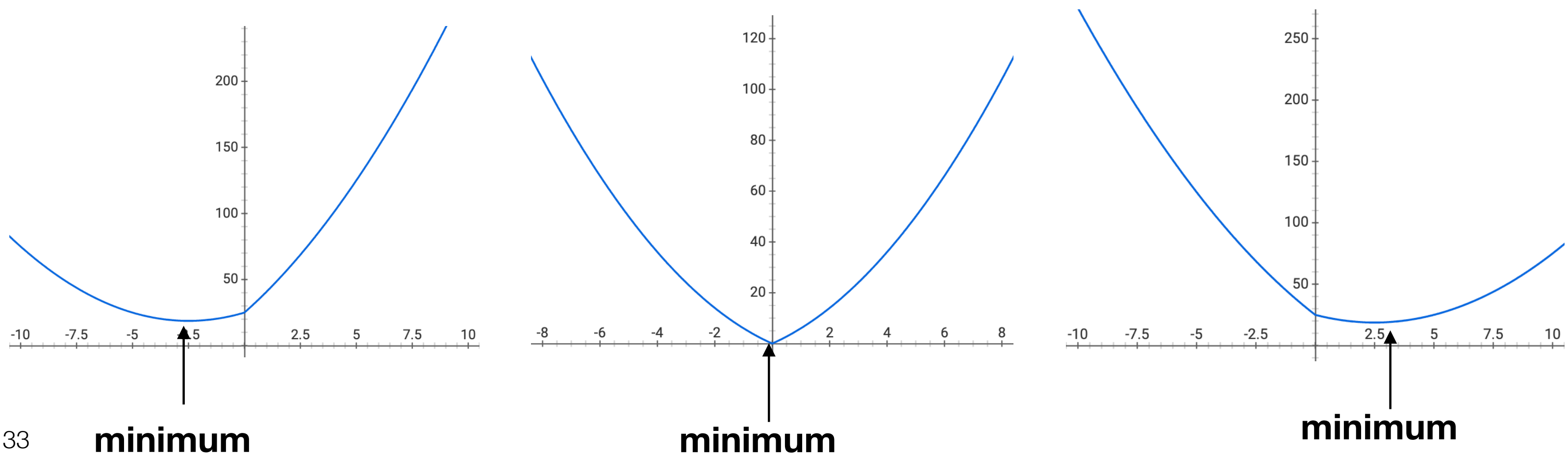
$$\left\| \mathbf{X}[:,1]w_1 - \left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right) \right\|_2^2 + \lambda\left(|w_1| + \underbrace{\|w_{-1}\|_1}_{\text{constant}} \right)$$

  - hence, the coordinate descent update boils down to

$$w_1^{(t)} \leftarrow \arg\min_{w_1} \underbrace{\left\| \mathbf{X}[:,1]w_1 - \left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right) \right\|_2^2 + \lambda |w_1|}_{f(w_1)}$$

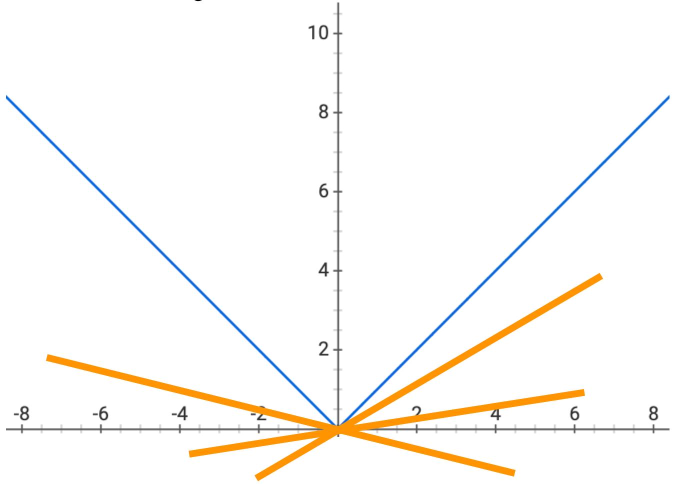
Convexity

- to find the minimizer of $f(w_1)$, let's study some properties
- for simplicity, we represent the objective function as
$$f(w_1) = (aw_1 - b)^2 + \lambda \,|\, w_1 \,|$$
- this function is
  - **convex**, and
  - **non-differentiable**
- depending on the values of a and b, the function looks like one of the three below



Convexity

$$f(x) = |x|$$



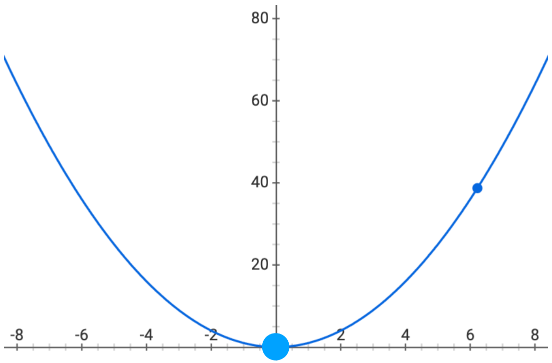
- for a **non-differentiable** function, gradient is not defined at some points, for example at $x = 0$ for $f(x) = |x|$
- at such points, **sub-gradient** plays the role of gradient
  - sub-gradient at a differentiable point is the same as the gradient
  - sub-gradient at a non-differentiable point is a set of vector satisfying

$$\partial f(x) \; = \; \left\{ g \in \mathbb{R}^d \,|\, f(y) \geq f(x) + g^T(y - x), \text{ for all } y \in \mathbb{R}^d \right\}$$

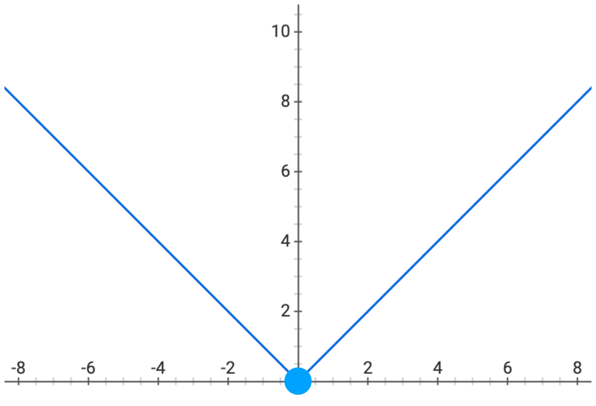
  - for example, $\partial |x| \; = \begin{cases} +1 & \text{for } x > 0 \\ [-1, 1] & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases}$

Convexity

- for convex differentiable functions, the minimum is achieved at points where gradient is zero



- for convex non-differentiable functions, the minimum is achieved at points where sub-gradient includes zero



Computing the sub-gradient

$$w_1^{(t)} = \arg\min_{w_1} \underbrace{\left\| \mathbf{X}[:,1]w_1 - \left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right)\right\|_2^2 + \lambda |w_1|}_{f(w_1)}$$

Coordinate descent update on Lasso

$$w_1^{(t)} = \arg\min_{w_1} \underbrace{\left\| \mathbf{X}[:,1]w_1 - \left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right) \right\|_2^2 + \lambda |w_1|}_{f(w_1)}$$

- this is $f(w_1) = (aw_1 - b)^2 + \lambda |w_1|$ + constants, with
  - $a = \sqrt{\mathbf{X}[:,1]^T \mathbf{X}[:,1]}$, and
  - $b = \dfrac{\mathbf{X}[:,1]^T \left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right)}{\sqrt{\mathbf{X}[:,1]^T \mathbf{X}[:,1]}}$
- $f(w_1)$ is non-differentiable, and its sub-gradient is

$$\partial f(w_1) = (2a(aw_1 - b) + \lambda \, \partial |w_1|$$

$$= \begin{cases} 2a(aw_1 - b) + \lambda & \text{for } w_1 > 0 \\ [-2ab - \lambda, -2ab + \lambda] & \text{for } w_1 = 0 \\ 2a(aw_1 - b) - \lambda & \text{for } w_1 < 0 \end{cases}$$

Computing the minimizer

- the minimizer $w_1^{(t)}$ is when zero is included in the sub-gradient

$$\partial f(w_1) = \begin{cases} 2a(aw_1 - b) + \lambda & \text{for } w_1 > 0 \\ [-2ab - \lambda, -2ab + \lambda] & \text{for } w_1 = 0 \\ 2a(aw_1 - b) - \lambda & \text{for } w_1 < 0 \end{cases}$$

Computing the minimizer

- the minimizer $w_1^{(t)}$ is when zero is included in the sub-gradient

$$\partial f(w_1) = \begin{cases} 2a(aw_1 - b) + \lambda & \text{for } w_1 > 0 \\ [-2ab - \lambda, -2ab + \lambda] & \text{for } w_1 = 0 \\ 2a(aw_1 - b) - \lambda & \text{for } w_1 < 0 \end{cases}$$

Computing the minimizer

- the minimizer $w_1^{(t)}$ is when zero is included in the sub-gradient

$$\partial f(w_1) = \begin{cases} 2a(aw_1 - b) + \lambda & \text{for } w_1 > 0 \\ [-2ab - \lambda, -2ab + \lambda] & \text{for } w_1 = 0 \\ 2a(aw_1 - b) - \lambda & \text{for } w_1 < 0 \end{cases}$$

How do we find the minimizer?

- the minimizer $w_1^{(t)}$ is when zero is included in the sub-gradient

$$\partial f(w_1) = \begin{cases} 2a(aw_1 - b) + \lambda & \text{for } w_1 > 0 \\ [-2ab - \lambda, -2ab + \lambda] & \text{for } w_1 = 0 \\ 2a(aw_1 - b) - \lambda & \text{for } w_1 < 0 \end{cases}$$

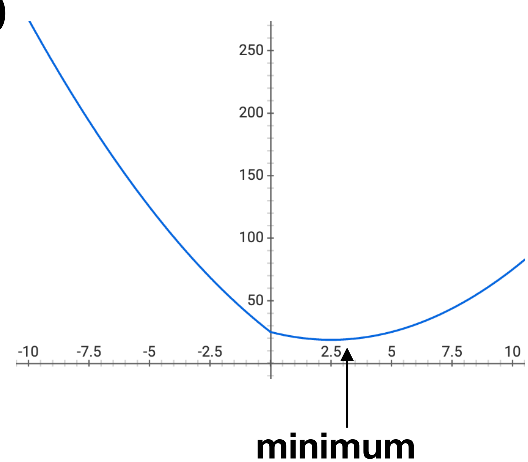
- case 1:
  - $2a(aw_1 - b) + \lambda = 0$ for some $w_1 > 0$
  - this happens when

    $$w_1 = \frac{-\lambda + 2ab}{2a^2} > 0$$

  - hence,

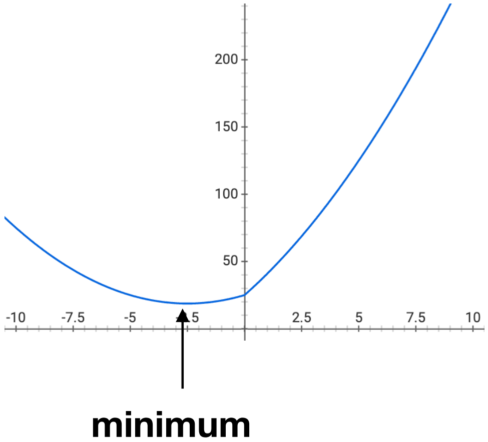
    $$w_1^{(t)} \leftarrow \frac{b}{a} - \frac{\lambda}{2a^2},$$

    if $\lambda < 2ab$



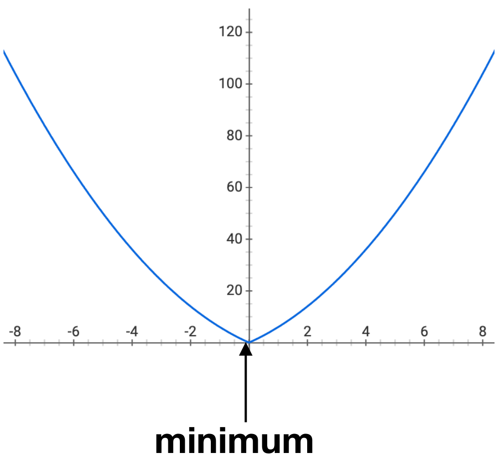
- case 2:
  - $2a(aw_1 - b) - \lambda = 0$ for some $w_1 < 0$
  - this happens when
    $$w_1 = \frac{\lambda + 2ab}{2a^2} < 0$$
  - hence,
    $$w_1^{(t)} \leftarrow \frac{b}{a} + \frac{\lambda}{2a^2},$$

    if $\lambda < -2ab$



- case 3:
  - $0 \in [-2ab - \lambda, -2ab + \lambda]$
  - and $w_1 = 0$
  - hence,
    $$w_1^{(t)} \leftarrow 0,$$

    if $-\lambda \leq 2ab \leq \lambda$

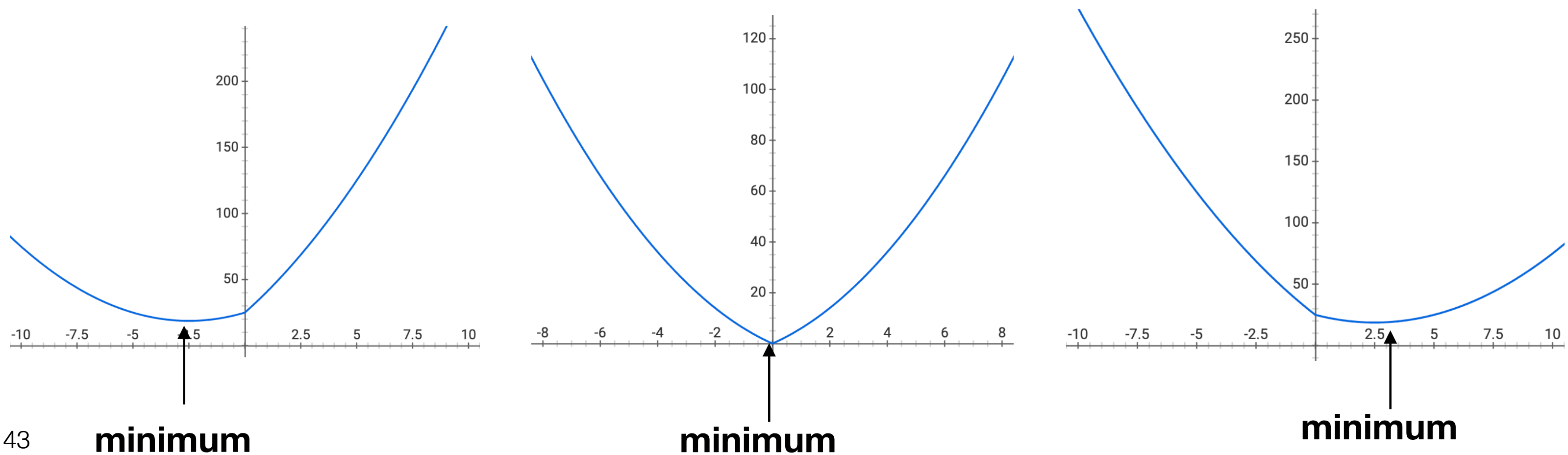


Coordinate descent on Lasso

- considering all three cases, we get the following update rule by setting the sub-gradient to zero

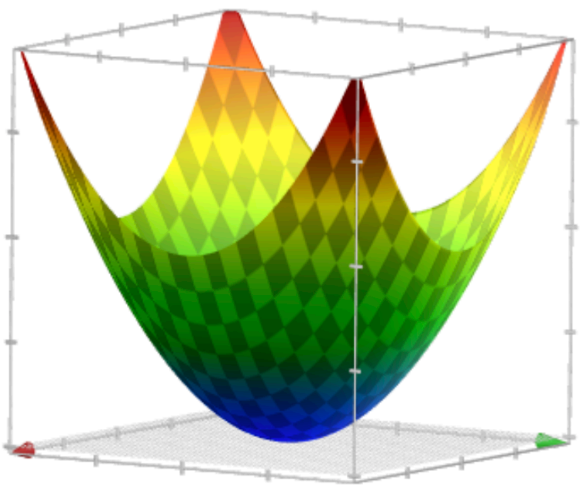
$$w_1^{(t)} \leftarrow \begin{cases} \frac{b}{a} - \frac{\lambda}{2a^2} & \text{for } 2ab > \lambda \\ 0 & \text{for } -\lambda \leq 2ab \leq \lambda \\ \frac{b}{a} + \frac{\lambda}{2a^2} & \text{for } \lambda < -2ab \end{cases}$$

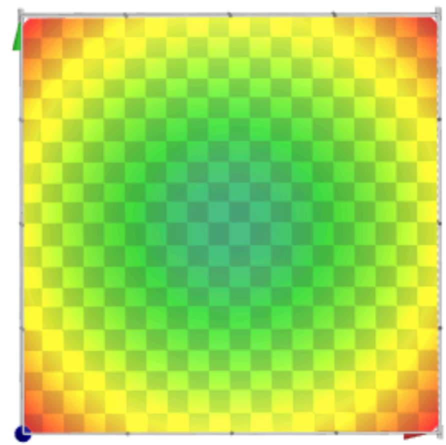
- where $a = \sqrt{\mathbf{X}[:,1]^T \mathbf{X}[:,1]}$, and $b = \dfrac{\mathbf{X}[:,1]^T\left(\mathbf{y} - \mathbf{X}[:,2:d]w_{-1}\right)}{\sqrt{\mathbf{X}[:,1]^T \mathbf{X}[:,1]}}$



When does coordinate descent work?

- Consider minimizing a **differentiable convex** function $f(x)$, then coordinate descent converges to the global minima

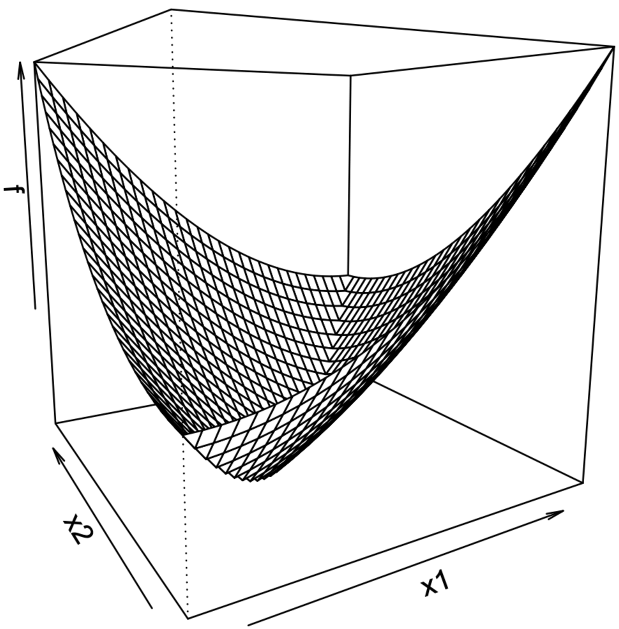


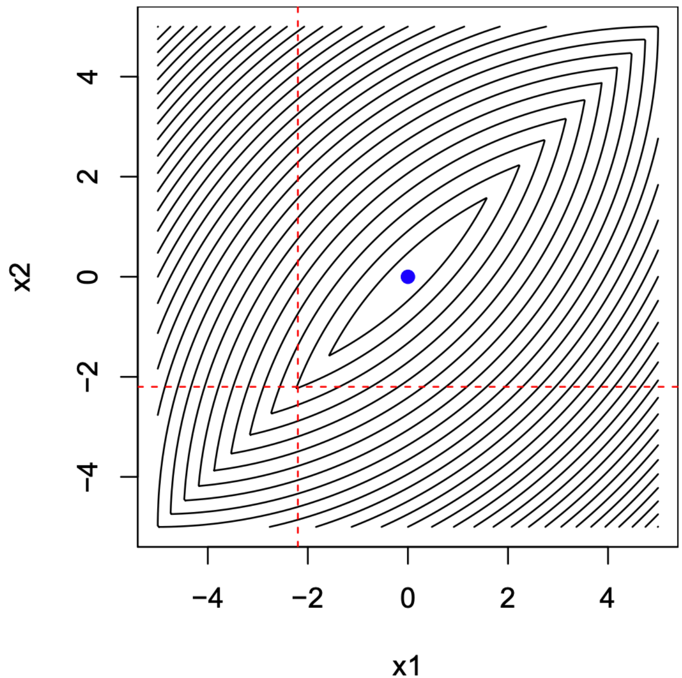


- when coordinate descent has stopped, that means $\dfrac{\partial f(x)}{\partial x_j} = 0$ for all $j \in \{1, \ldots, d\}$
- this implies that the gradient $\nabla_x f(x) = 0$, which happens only at minimum

When does coordinate descent work?

- Consider minimizing a **non-differentiable convex** function $f(x)$, then coordinate descent can get stuck

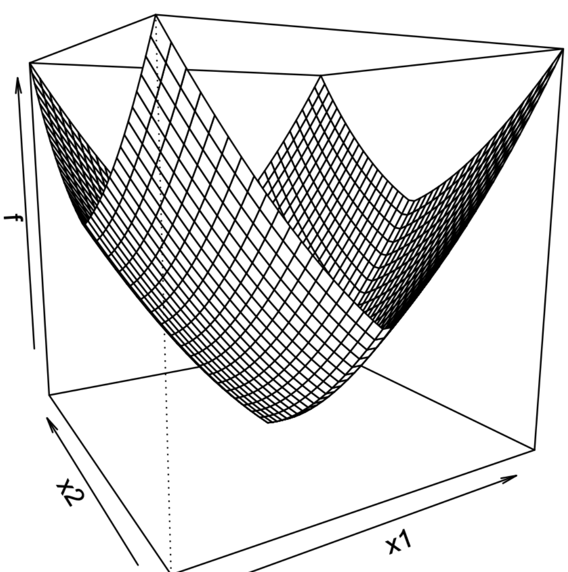


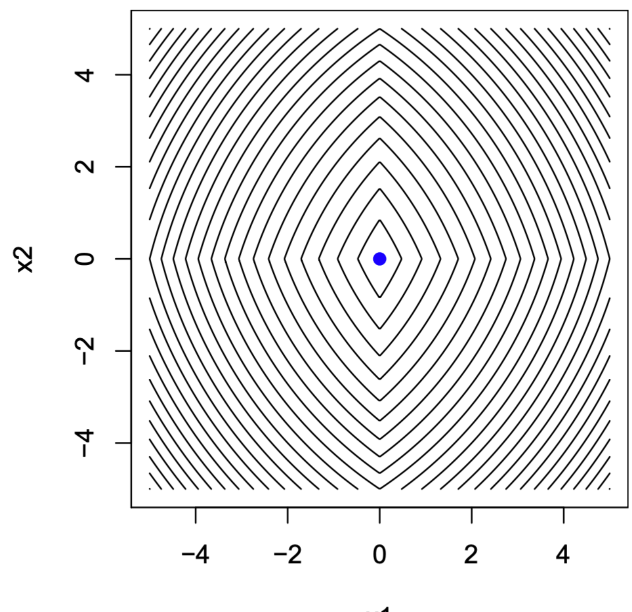


•

When does coordinate descent work?

- then how can coordinate descent find optimal solution for Lasso?
- consider minimizing a **non-differentiable convex** function but has a structure of $f(x) = g(x) + \sum_{j=1}^{d} h_j(x_j)$, with differentiable convex function $g(x)$ and coordinate-wise non-differentiable convex functions $h_j(x_j)$'s, then coordinate descent converges to the global minima





-