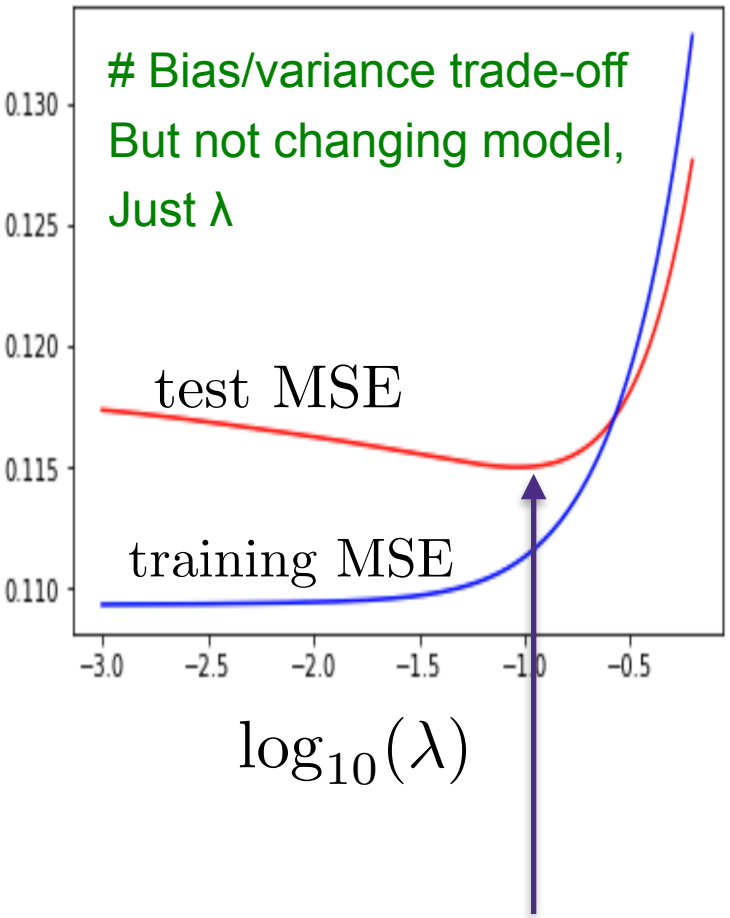
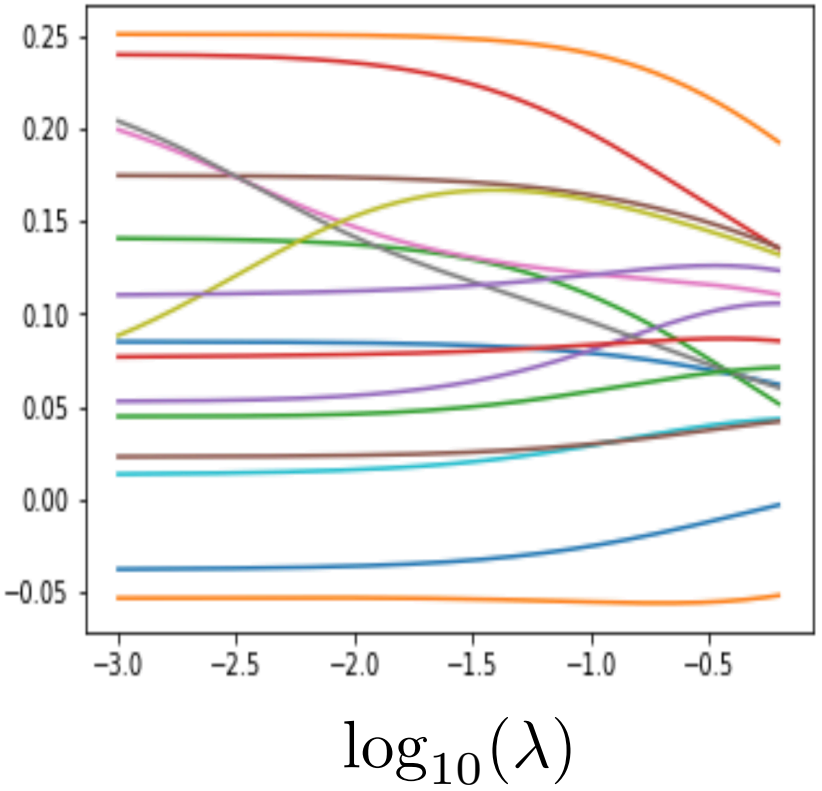
Finishing up bias-variance for Ridge



Ridge regression: $\text{minimize} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$



Illustrative potato delivery: w_i



- this gain in test MSE comes from shrinking w's to get a less sensitive predictor
  (which in turn reduces the variance)

- this is the role of regularizer

A quick trick/lemma for all of our pesky cross terms

Lemma:

$$\mathbb{E}[(A-B)^2] = \mathbb{E}[(A-C)^2] + \mathbb{E}[(B-C)^2]$$

When

(1) $(A-C)$ and $(B-C)$have finite variance,

(2) There exists some information G which

(1) fixes $(A-C)$ and

(2) Makes $\mathbb{E}[B-C\,|\,G] = 0$

Bias-Variance Properties

Plan: work through an example where we can directly put bias/ variance in terms of λ

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X, \ \mathbf{y} = \mathbf{X}w + \epsilon, \ \epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$ for some ground truth model parameter w
- The true error at a sample with feature x is $\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$

x is a test sample, $\hat{w}_{\text{ridge}}$ is fit to a dataset

Bias-Variance Properties

Plan: work through an example where we can directly put bias/ variance in terms of λ

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X, \ \mathbf{y} = \mathbf{X}w + \epsilon, \ \epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$
- The true error at a sample with feature x is

$$\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

x is a test sample, $\hat{w}_{\text{ridge}}$ is fit to a dataset

$$= \underbrace{\mathbb{E}_{y|x}[(y - \mathbb{E}[y\,|\,x])^2 \,|\, x]}_{\text{Irreducible Error}} + \underbrace{\mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}[y\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]}_{\text{Learning Error}}$$

Bias-Variance Properties

Plan: work through an example where we can directly put bias/ variance in terms of λ

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X,\ \ \mathbf{y} = \mathbf{X}w + \epsilon,\ \ \epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$
- The true error at a sample with feature x is

$$\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2\,|\,x]$$ # x is a test sample, $\hat{w}_{\text{ridge}}$ is fit to a dataset

$$= \mathbb{E}_{y|x}[(y - \mathbb{E}[y\,|\,x])^2\,|\,x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}[y\,|\,x] - x^T\hat{w}_{\text{ridge}})^2\,|\,x]$$

$$= \mathbb{E}_{y|x}[(y - x^Tw)^2\,|\,x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(x^Tw - x^T\hat{w}_{\text{ridge}})^2\,|\,x]$$ # true w

Bias-Variance Properties

Plan: work through an example where we can directly put bias/ variance in terms of λ

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X, \;\; \mathbf{y} = \mathbf{X}w + \epsilon, \;\; \epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$
- The true error at a sample with feature x is

$$\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$ # x is a test sample, $\hat{w}_{\text{ridge}}$ is fit to a dataset

$$= \mathbb{E}_{y|x}[(y - \mathbb{E}[y\,|\,x])^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}[y\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

$$= \mathbb{E}_{y|x}[(y - x^Tw)^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(x^Tw - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$ # true w

$$= \sigma^2 + (x^Tw - \mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x])^2 + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

Irreduc. Error — Bias-squared — Variance

Bias-Variance Properties

Plan: work through an example where we can directly put bias/ variance in terms of λ

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X, \;\; \mathbf{y} = \mathbf{X}w + \epsilon, \;\; \epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$
- The true error at a sample with feature x is

$$\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

x is a test sample, $\hat{w}_{\text{ridge}}$ is fit to a dataset

$$= \mathbb{E}_{y|x}[(y - \mathbb{E}[y\,|\,x])^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}[y\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

$$= \mathbb{E}_{y|x}[(y - x^Tw)^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(x^Tw - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

true w

$$= \sigma^2 + (x^Tw - \mathbb{E}_{\mathcal{D}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x])^2 + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

Irreduc. Error — Bias-squared — Variance

nI

Suppose $\mathbf{X}^T\mathbf{X} = n\mathbf{I}$, then $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T(\mathbf{X}w + \epsilon)$

$$= \frac{n}{n+\lambda}w + \frac{1}{n+\lambda}\mathbf{X}^T\epsilon$$

Not realistic but pretend we're sampling independent Gaussians

λ trades off weight on data vs. noise

Bias-Variance Properties

Suppose $\mathbf{X}^T\mathbf{X} = n\mathbf{I}$, then

$$\hat{w}_{\text{ridge}} = \frac{n}{n+\lambda} w + \frac{1}{n+\lambda}\mathbf{X}^T\epsilon$$

- Recall: $\hat{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model: $x_i \sim P_X, \quad \mathbf{y} = \mathbf{X}w + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$
- The true error at a sample with feature x is

$$\begin{aligned}
&\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x] \\
&\quad = \mathbb{E}_{y|x}[(y - \mathbb{E}[y\,|\,x])^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}[y\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x] \\
&\quad = \mathbb{E}_{y|x}[(y - x^Tw)^2 \,|\, x] + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(x^Tw - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]
\end{aligned}$$

$$= \sigma^2 + (x^Tw - \mathbb{E}_{\mathcal{D}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x])^2 + \mathbb{E}_{\mathcal{D}_{\text{train}}}[(\mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}}[x^T\hat{w}_{\text{ridge}}\,|\,x] - x^T\hat{w}_{\text{ridge}})^2 \,|\, x]$$

(verify at home)

$$= \sigma^2 + \frac{\lambda^2}{(n+\lambda)^2}(w^Tx)^2 + \frac{\sigma^2 n}{(n+\lambda)^2}\|x\|_2^2$$

Irreduc. Error | Bias-squared | Variance

- λ trades off bias + variance
- Larger λ →
  - Bigger bias
  - Smaller variance

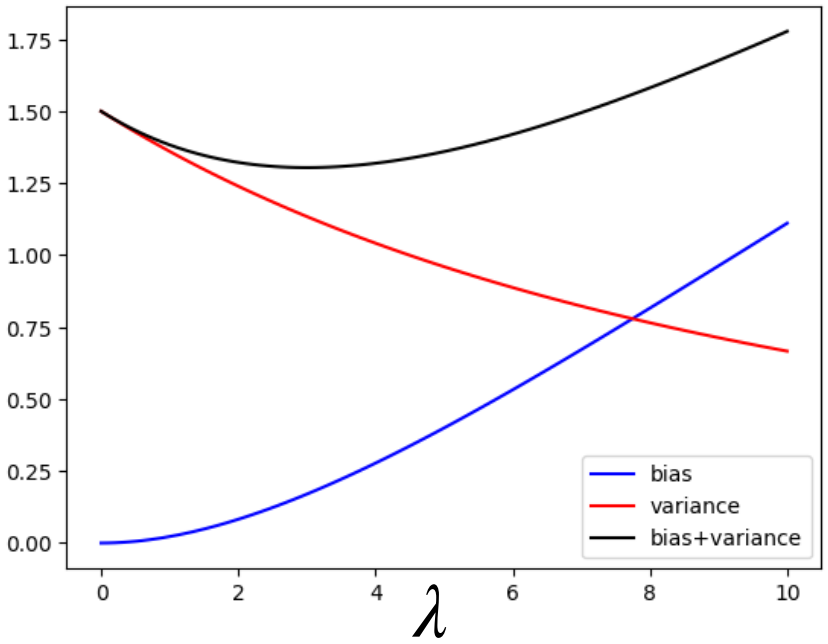
Bias-Variance Properties

Suppose $\mathbf{X}^T\mathbf{X} = n\mathbf{I}$,

- Ridge regressor: $\hat{w}_{ridge} = \arg\min_w \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 + \lambda ||w||_2^2$
- True error

$$\mathbb{E}_{y,\mathcal{D}_{\text{train}}|x}[(y - x^T\hat{w}_{\text{ridge}})^2 \,|\, x] = \sigma^2 + \underbrace{\frac{\lambda^2}{(n+\lambda)^2}(w^T x)^2}_{\text{Bias-squared}} + \underbrace{\frac{\sigma^2 n}{(n+\lambda)^2}\|x\|_2^2}_{\text{Variance}}$$

d=10, n=20, $\sigma^2 = 3.0, \|w\|_2^2 = 10$



as $\lambda \to 0$,
$\hat{w}_{\text{ridge}} \to \hat{w}_{\text{LS}}$

as $\lambda \to \infty$
$\hat{w}_{\text{ridge}} \to 0$

What you need to know…

- Regularization
  - Penalizes complex models towards preferred, simpler models
- Ridge regression
  - L_2 penalized least-squares regression $\lambda \|w\|_2^2$
  - Regularization parameter trades off model complexity with training error
  - Never regularize the offset!

Simple Variable Selection
LASSO: Sparse Regression



Sparsity

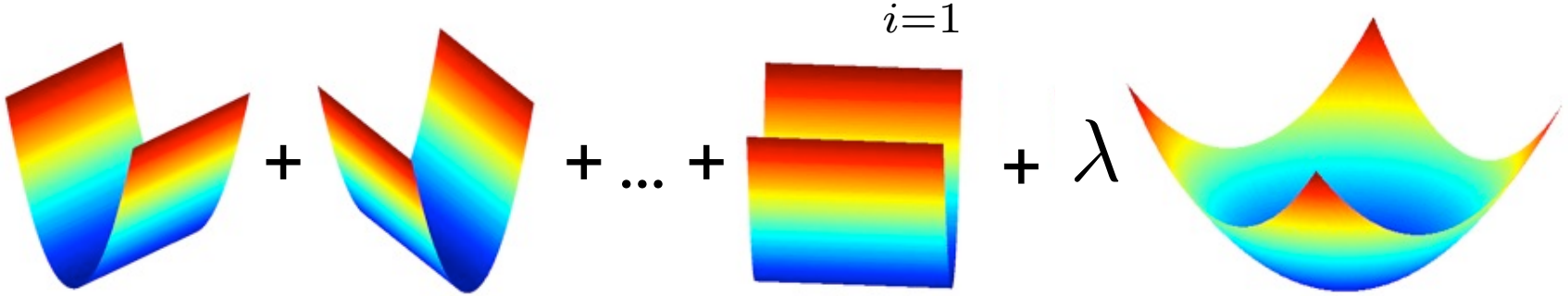
$$\widehat{w}_{LS} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2$$

- Vector w is sparse, if many entries are zero

Ridge vs. Lasso Regression

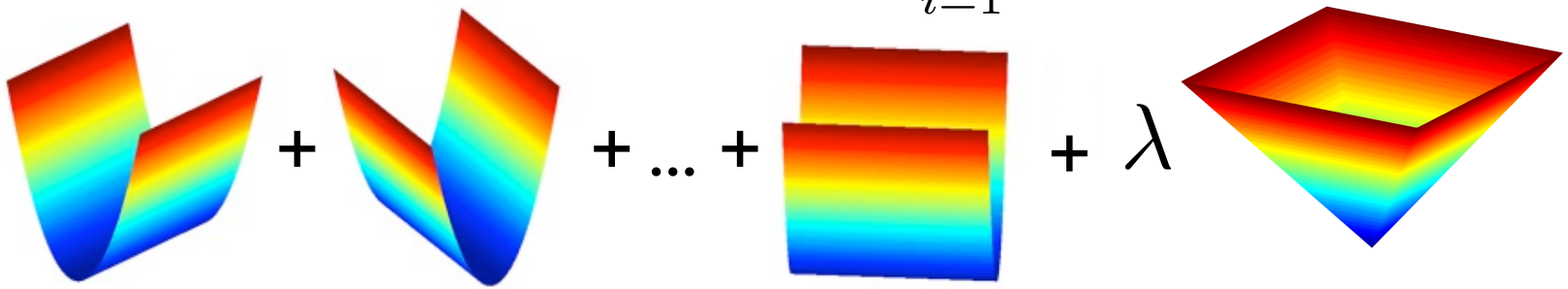
- Ridge Regression objective:

$$\widehat{w}_{ridge} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 + \lambda ||w||_2^2$$



- Lasso objective:

$$\widehat{w}_{lasso} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 + \lambda ||w||_1$$



Regularized/Penalized Least Squares

$$\text{Ridge} : r(w) = ||w||_2^2 \qquad \text{Lasso} : r(w) = ||w||_1$$

$$\widehat{w}_r = \arg\min_w \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 + \lambda r(w)$$

For any $\lambda \geq 0$ for which $\widehat{w}_r$ achieves the minimum, there exists a $\nu \geq 0$ such that

$$\widehat{w}_r = \arg\min_w \sum_{i=1}^{n} \left(y_i - x_i^T w\right)^2 \qquad \text{subject to } r(w) \leq \nu$$

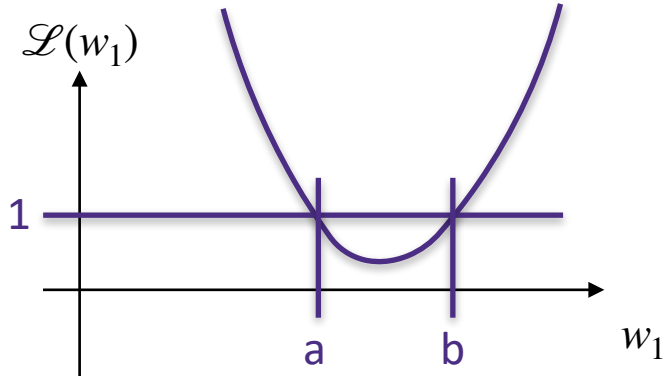
How does Lasso penalization affect solutions?

$$\text{minimize}_w \quad \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

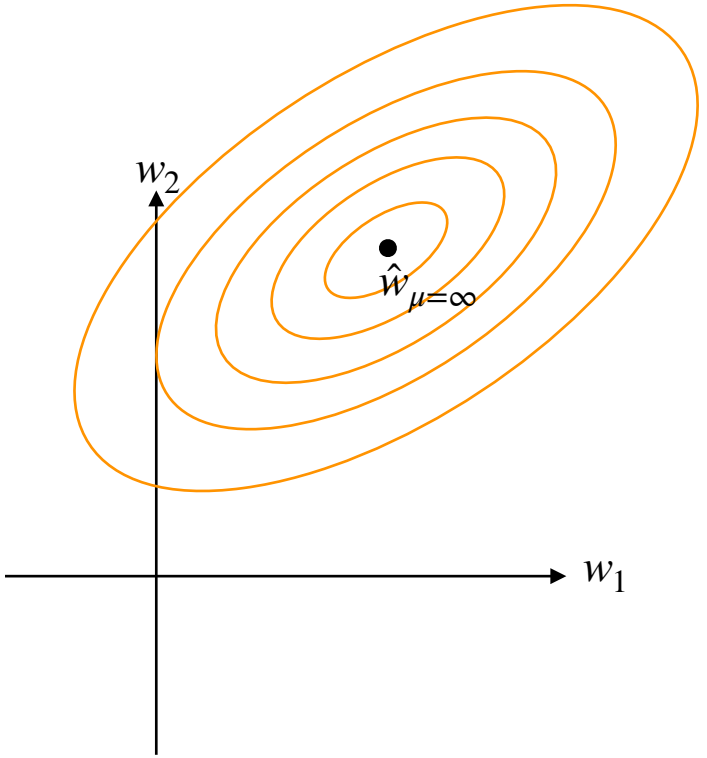
$$\text{subject to} \quad \|w\|_1 \leq \mu$$

- the ***level set*** of a function $\mathscr{L}(w_1, w_2)$ is defined as the set of points (w_1, w_2) that have the same function value (objective value)
- the level set of a quadratic function is an oval
- the center of the oval is the least squares solution $\hat{w}_{\mu=\infty} = \hat{w}_{\text{LS}}$

1-D example with quadratic loss



Level set of $\mathscr{L}(w_1)$ at value 1 is {a,b}



Why does Lasso give sparse solutions?

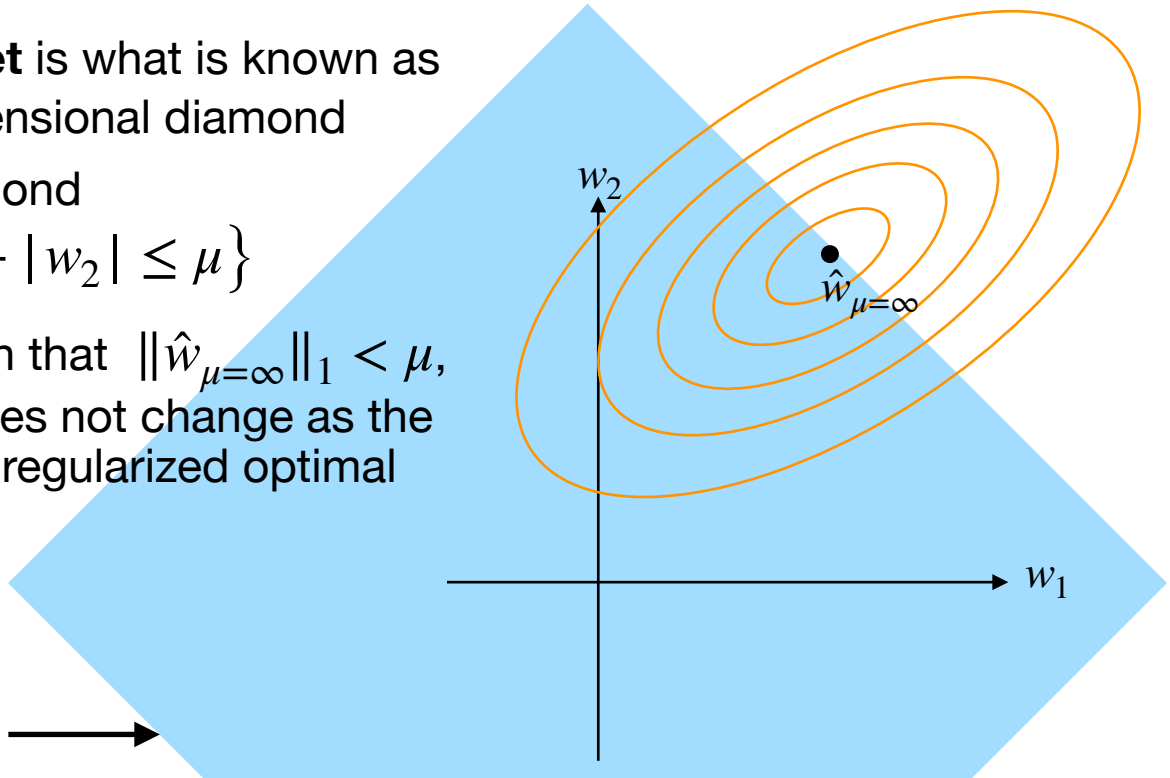
$$\text{minimize}_w \quad \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\text{subject to} \quad \|w\|_1 \leq \mu$$

- as we decrease μ from infinity, the feasible set becomes smaller
- the shape of the **feasible set** is what is known as L_1 ball, which is a high dimensional diamond
- In 2-dimensions, it is a diamond

$$\{(w_1, w_2) \mid |w_1| + |w_2| \leq \mu\}$$

- when μ is large enough such that $\|\hat{w}_{\mu=\infty}\|_1 < \mu$, then the optimal solution does not change as the feasible set includes the un-regularized optimal solution



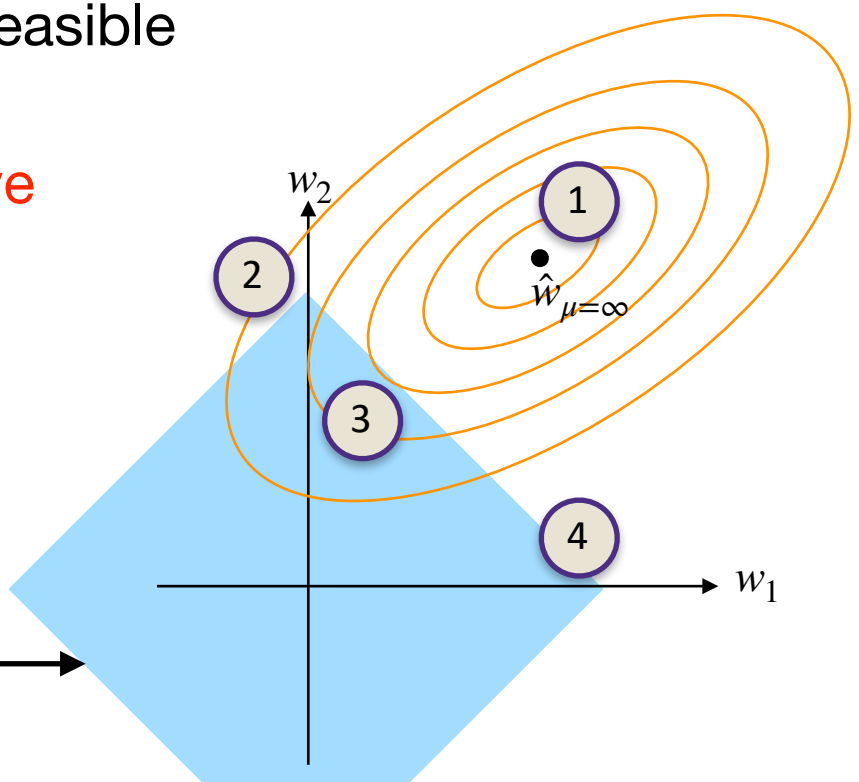
feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$ ⟶

Why does Lasso give sparse solutions?

$$\text{minimize}_w \quad \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\text{subject to} \quad \|w\|_1 \leq \mu$$

- As μ decreases (which is equivalent to increasing regularization λ) the feasible set (blue diamond) shrinks
- The optimal solution of the above optimization is ?



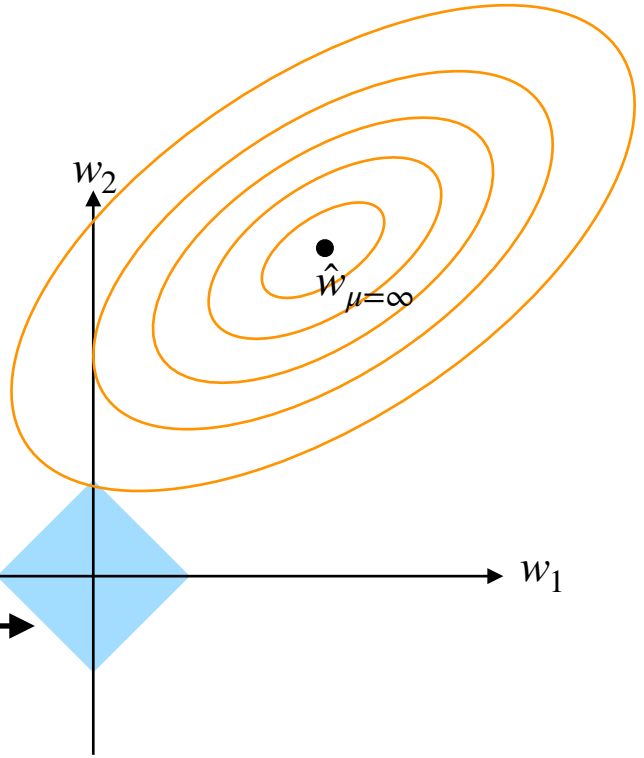
feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$

Why does Lasso give sparse solutions?

$$\text{minimize}_w \quad \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\text{subject to} \quad \|w\|_1 \leq \mu$$

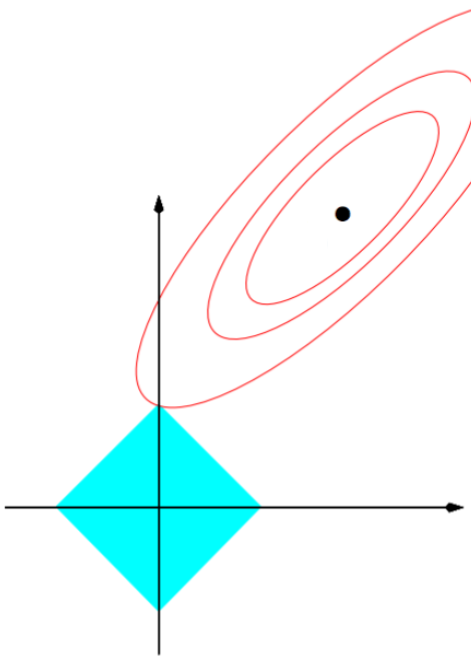
- For small enough μ, the optimal solution becomes **sparse**
- This is because the L_1-ball is "pointy",i.e., has sharp edges aligned with the axes

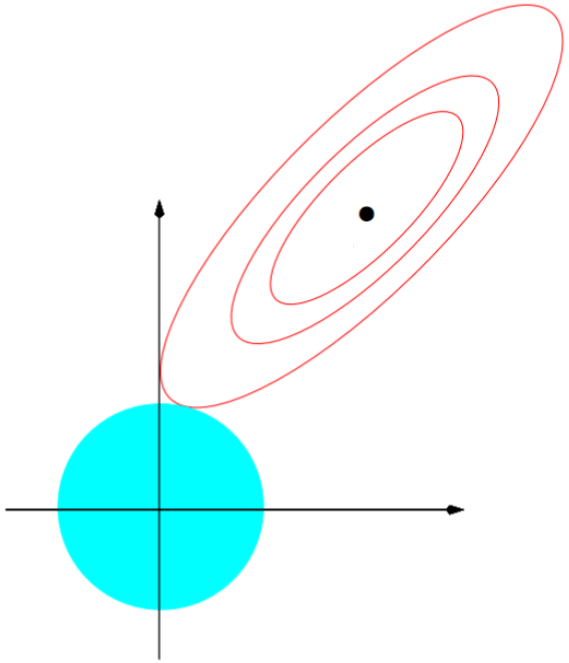


feasible set: $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$

Constrained Least Squares

- LASSO regression finds sparse solutions, as L_1-ball is "pointy"
- Ridge regression finds dense solutions, as L_2-ball is "smooth"



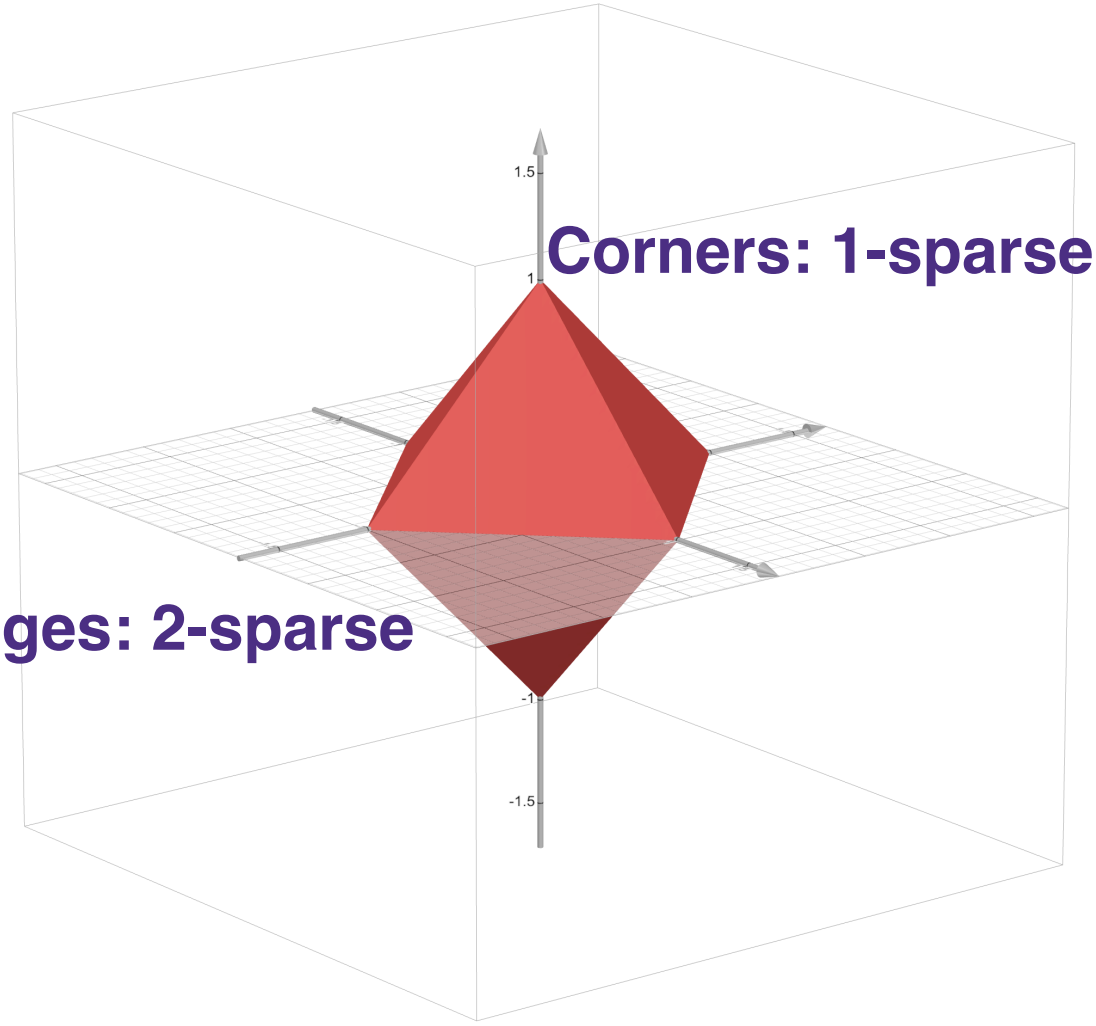


$$\text{minimize}_w \sum_{i=1}^{n} (w^T x_i - y_i)^2 \quad \text{subject to} \quad \|w\|_1 \leq \mu$$

$$\text{minimize}_w \sum_{i=1}^{n} (w^T x_i - y_i)^2 \quad \text{subject to} \quad \|w\|_2^2 \leq \mu$$

L1 Ball in Higher Dimensions

> L1 ball 3 dimensions



CSE 546
Gradient Descent

UNIVERSITY of WASHINGTON



How do we find LASSO weights?

- This is related to some questions you might have so far in this course
  - Why do we use quadratic loss, $\sum_{i=1}^{n} (y_i - w^T x_i)^2$?
  - Why is Gaussian noise so popular?

$$f(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z-\mu)^2}{\sigma^2}}$$

Convex!

- The local minima is the global minimum

  - Why was Ridge Regression with L_2 regularizer, $\|w\|_2^2$, the first to be used?

Math is easier, but also….

  - When we want sparsity, why do we use L_1 regularizer, $\|w\|_1$, and not $L_{0.5}$ regularizer, $\|w\|_{0.5}$?

Easier to optimize! Because…?

Our method for finding weights so far:

- Standard ML paradigm: Define loss, then optimize

$$\hat{w}_{LS} = \arg\min_{w} ||\mathbf{y} - \mathbf{X}w||_2^2 \qquad +\lambda\|w\|_1$$

$$2X^T(Xw - y) = 0 \qquad \text{\# } ||w||_1 \text{ is not differentiable at } w_i = 0$$

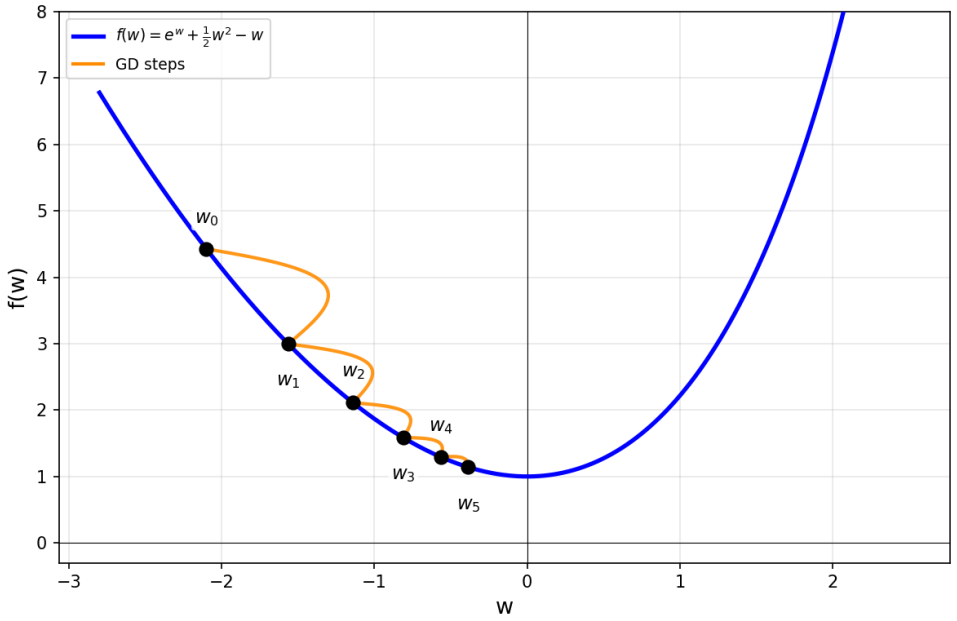
$$\hat{w} = (X^T X)^{-1} X^T y \qquad \text{\# No “closed form” solution!}$$

- But, no closed-form solutions for most losses we use in practice.
- Key idea: Iterative methods # Start with a guess for w
- Used everywhere! # Iteratively refine to reduce loss

Gradient Descent: THE iterative method

$$\widehat{w}_{LS} = \arg\min_{w} ||\mathbf{y} - \mathbf{X}w||_2^2 + \lambda \|w\|_1$$

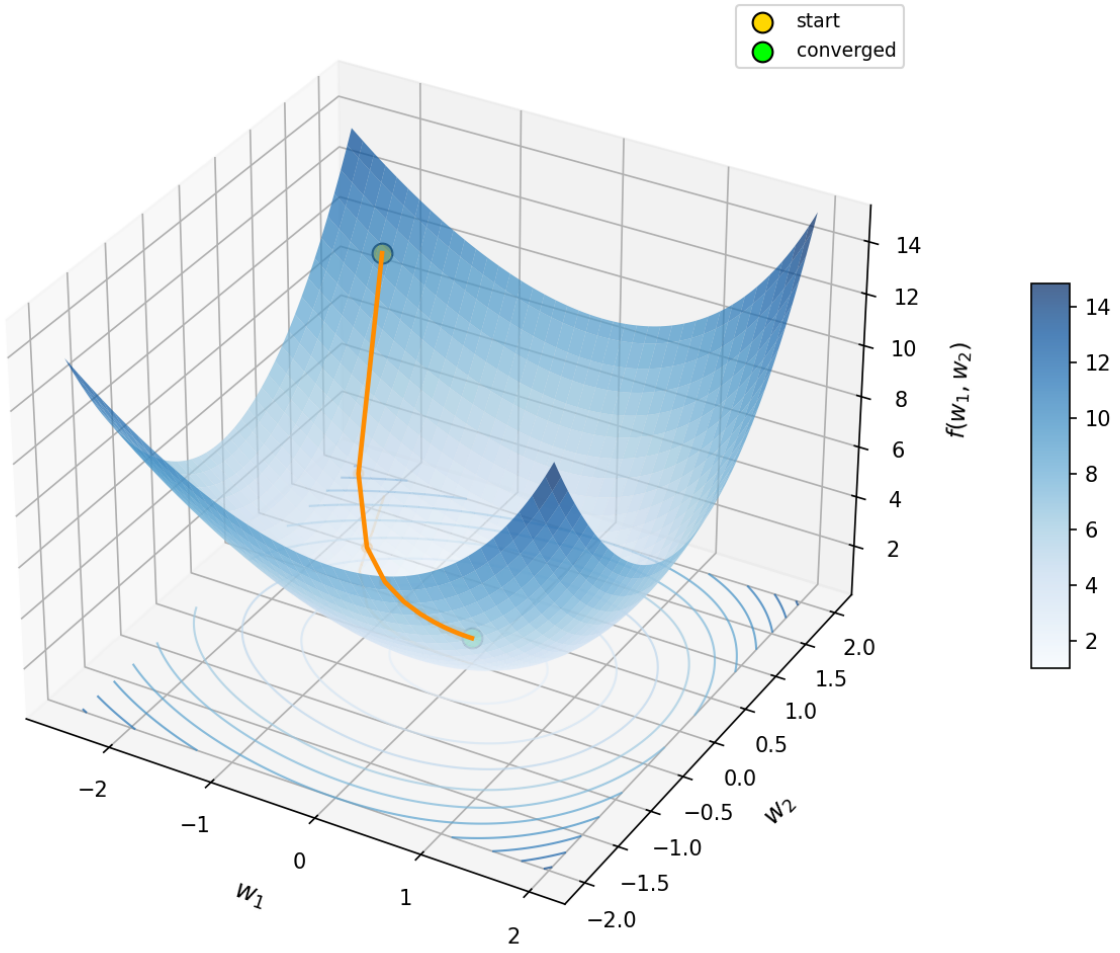
- What information do we need to minimize a function f?
  - It helps to know f(w)
  - And ideally $\nabla_w f(w)$



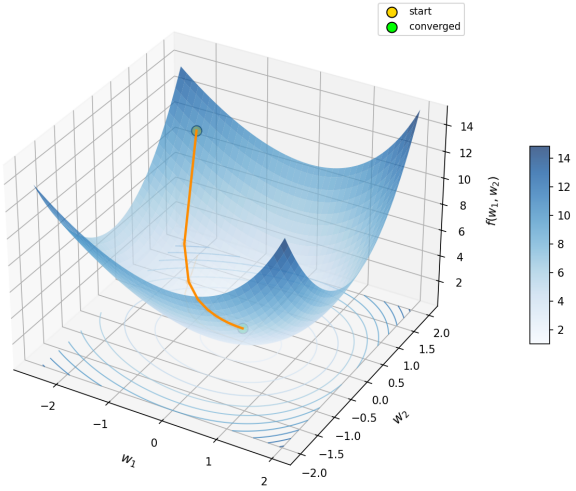
Step direction: $-\nabla_w f(w)$

Step size: $\eta \, | \nabla_w f(w) |$

Gradient Descent in higher dimensions



Gradient Descent pseudocode



hyperparameters

GradientDescent(f, w_0, η, T) :

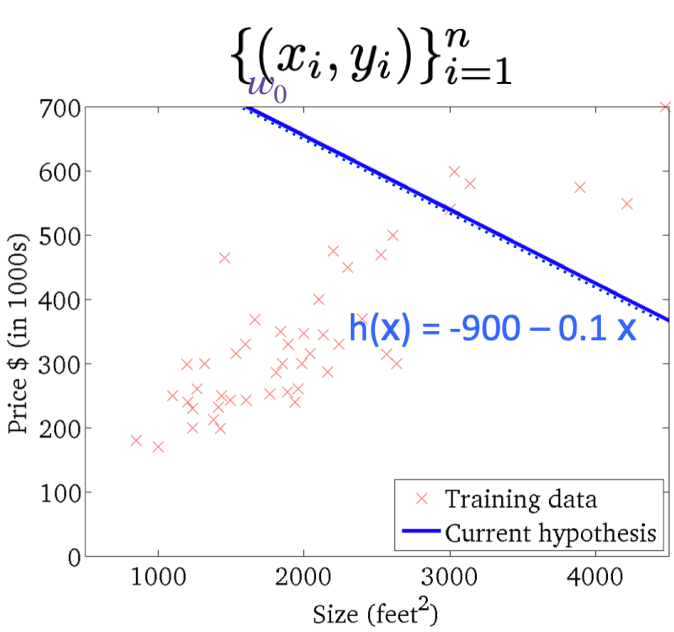
For (t = 1 to T):

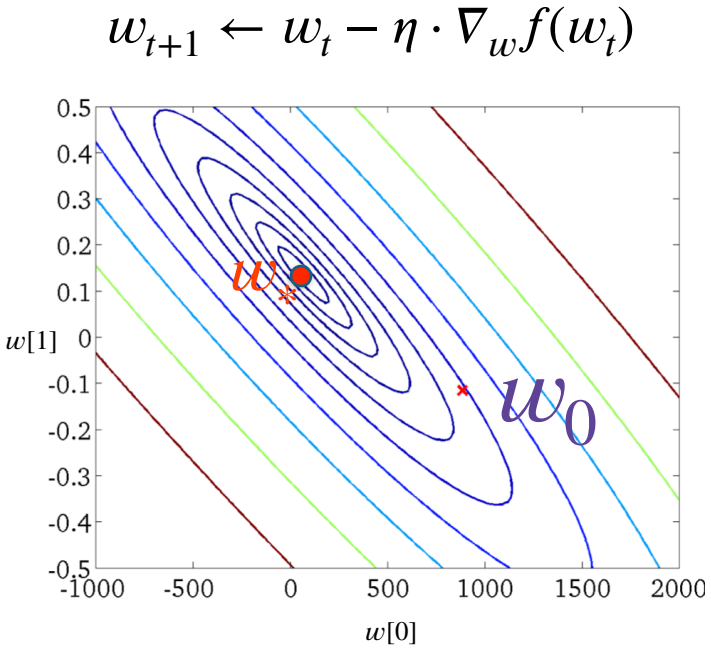
$$w_t = w_{t-1} - \eta \, | \, \nabla_{w_{t-1}} f(w_{t-1}) \, |$$

To pick w_0? Meh… randomly?

To pick η? Trial and error…

GD for LS, offset + 1 feature, initialized w_0



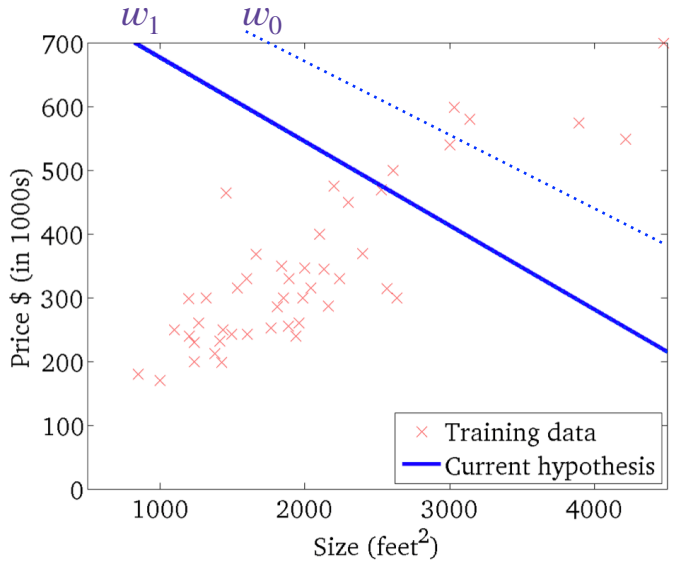


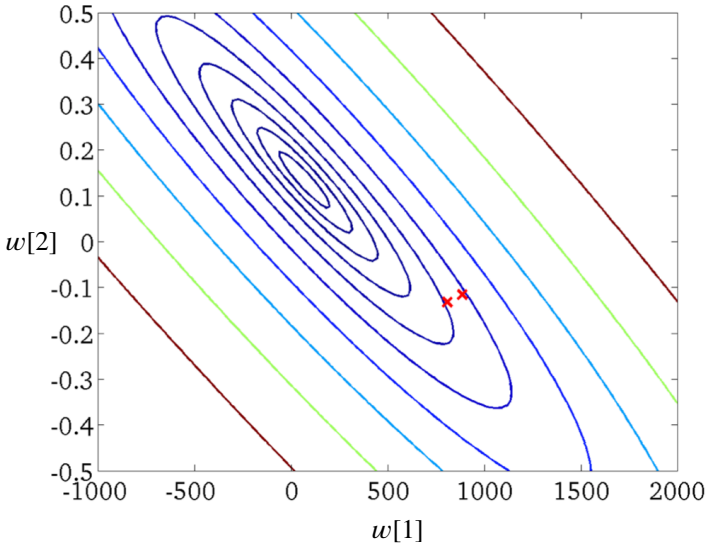
Evolution of the predictor y = w[0] + w[1]x

Gradient descent dynamics in the parameter space w
Ovals show the **level set** of the objective function

GD for LS, offset + 1 feature, w_1

current model



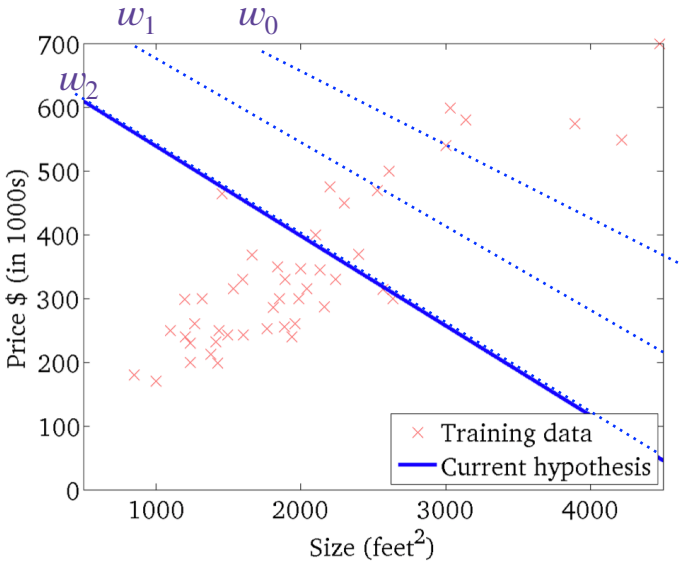


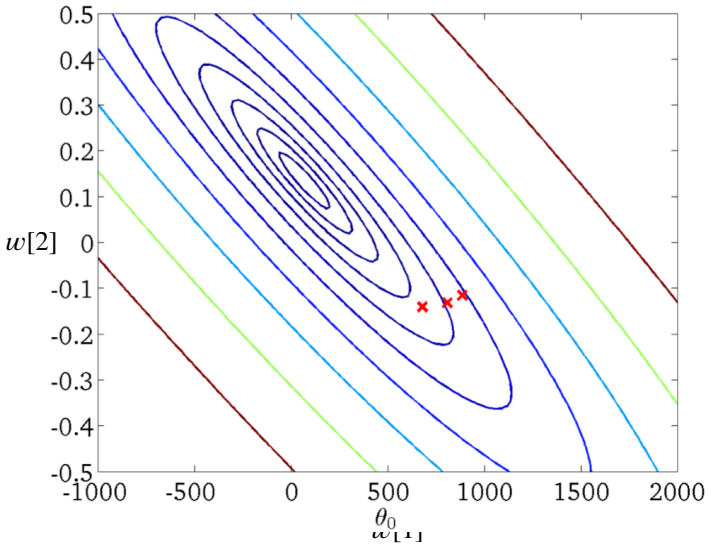
Evolution of the predictor y = w[0] + w[1]x

Gradient descent dynamics in the parameter space w
Ovals show the **level set** of the objective function

GD for LS, offset + 1 feature, w_2

current model



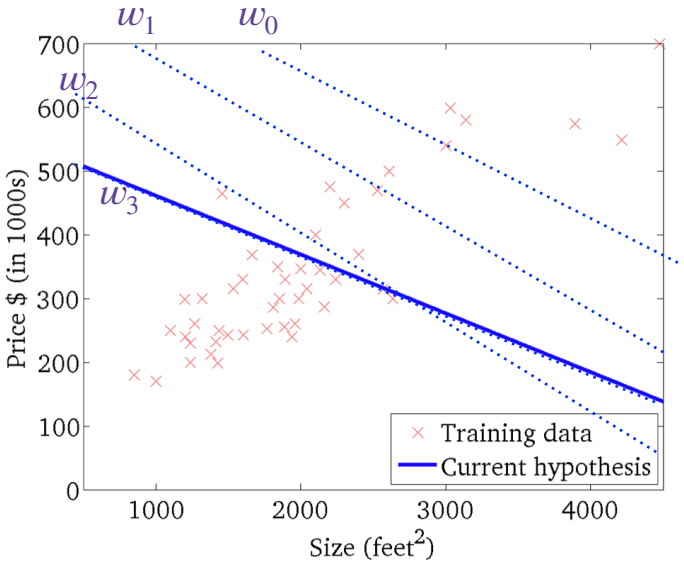


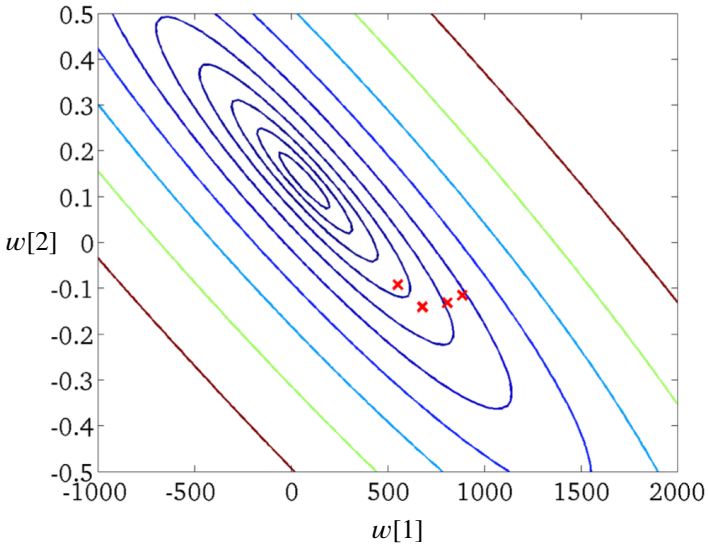
Evolution of the predictor y = w[0] + w[1]x

Gradient descent dynamics in the parameter space w
Ovals show the **level set** of the objective function

GD for LS, offset + 1 feature, w_3

current model



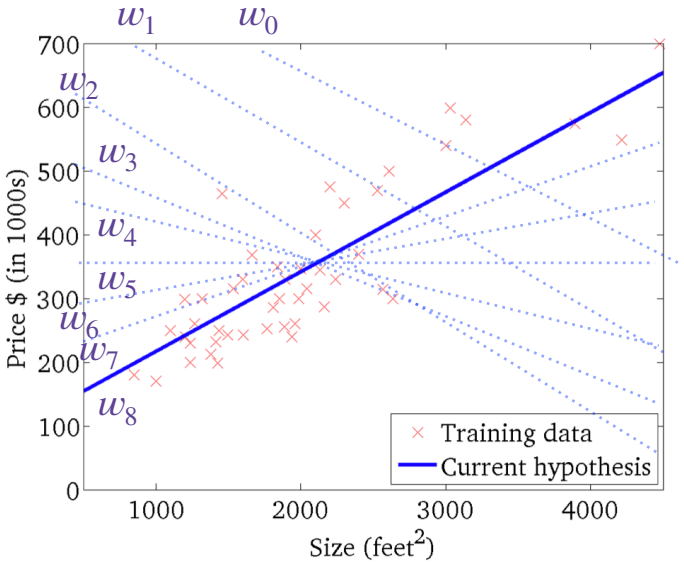


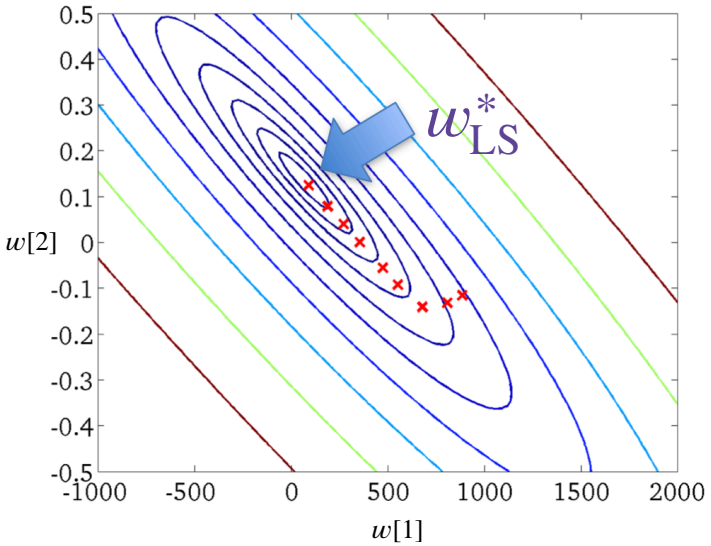
Evolution of the predictor y = w[0] + w[1]x

Gradient descent dynamics in the parameter space w
Ovals show the **level set** of the objective function

GD for LS, offset + 1 feature, w_4…

current model





Evolution of the predictor y = w[0] + w[1]x

Gradient descent dynamics in the parameter space w
Ovals show the **level set** of the objective function

GD for LS, analytically

$$\hat{w} = \underset{w}{\operatorname{argmin}} \frac{1}{2} \sum_i \left(y_i - x_i^\top w\right)^2$$

f(w) or loss function

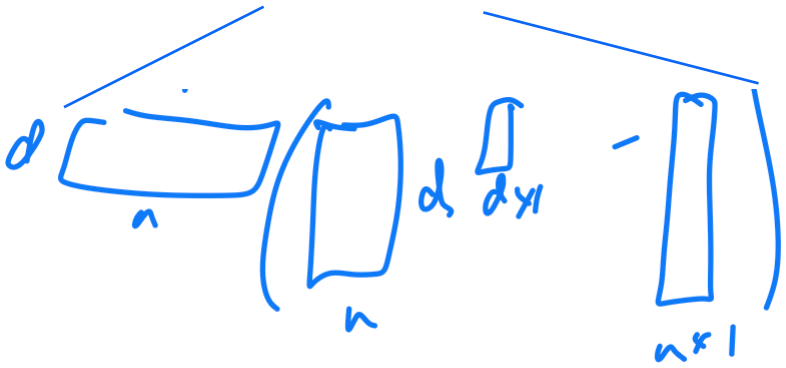
$$w_{t+1} = w_t - \eta \frac{df(w)}{dw}\Big|_{w=w_t}$$

$$\nabla_w f(w_0) = X^T(Xw - y)$$

from previous lectures

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

how can we check if this is right?



GD for LASSO, analytically

f(w) or loss function

$$\hat{w} = \underset{w}{\operatorname{argmin}} \frac{1}{2}\|y - Xw\|_2^2 + \lambda\|w\|_1$$

$$w_{t+1} = w_t - \eta \frac{df(w)}{dw}\Big|_{w=w_t}$$

LASSO regularizer is convex. So?

$$\nabla_w f = X^T(Xw - y) + \lambda \sum_{i=1}^{d} \text{sign}(w_i)$$

Local minima = global minimum

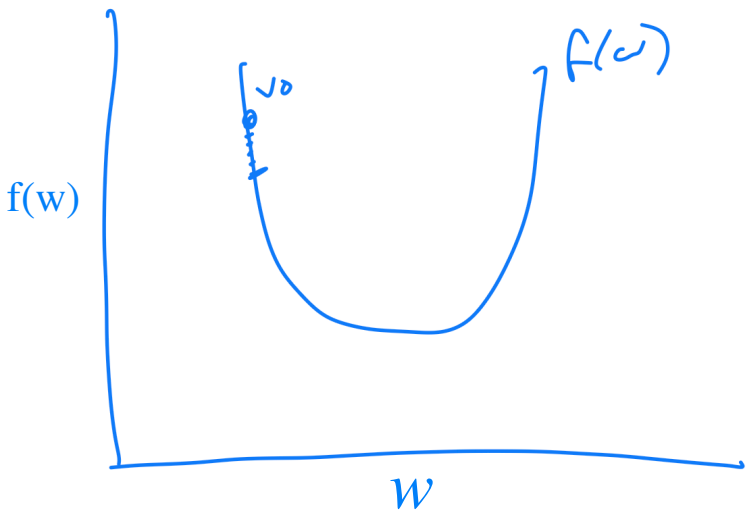
$$w_{t+1} \leftarrow w_t - \eta \cdot \nabla_w f_{w=w_t}$$

$$\frac{d|w_i|}{dw_i} = \begin{cases} +1 & w_i > 0 \\ [-1, 1] & w_i = 0 \\ -1 & w_i < 0 \end{cases}$$

Gradient of absolute value is undefined at w=0, so define a sub gradient

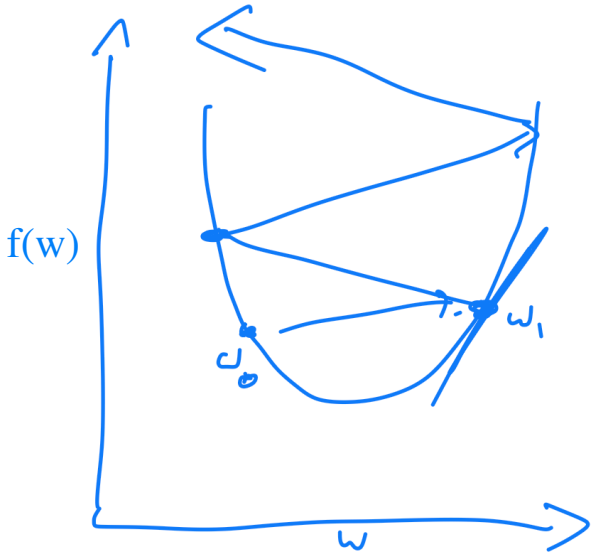
How do you choose a step size?

What can go wrong if η is too small?



Slow convergence

What can go wrong if η is too large?



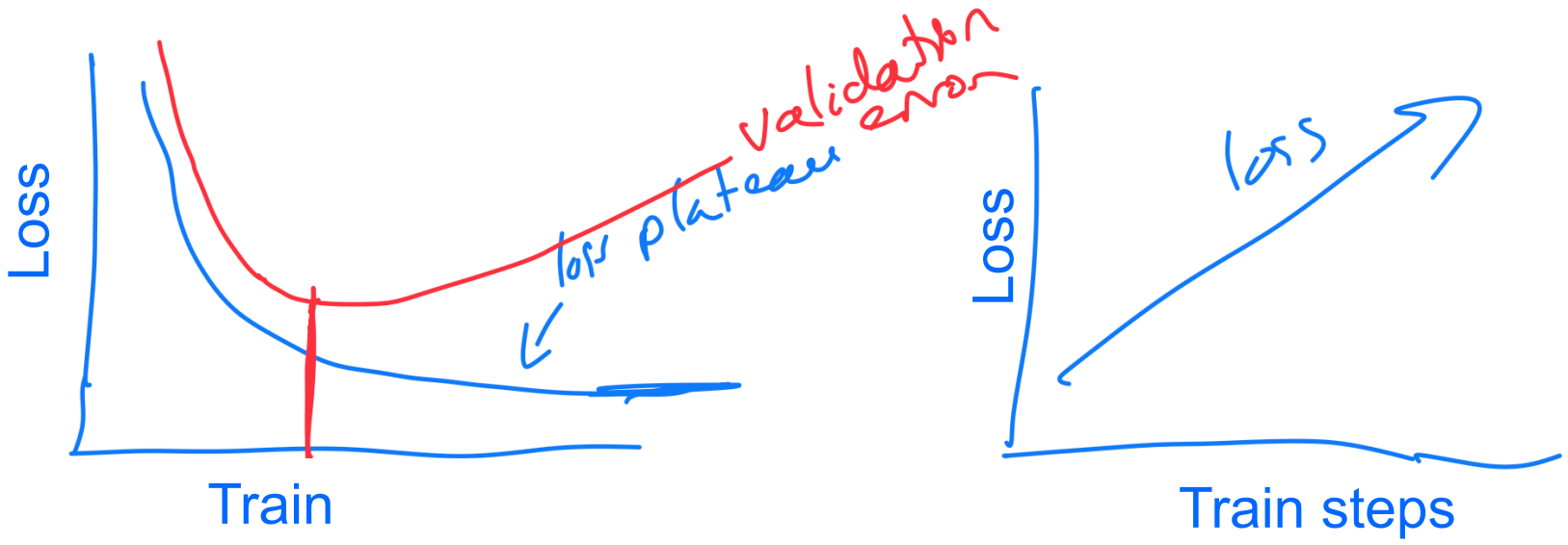
Divergence!

How do you choose a step size?

- In practice: guess and check # MAKE PLOTS

How do you choose a step size?

- In practice: guess and check # MAKE PLOTS



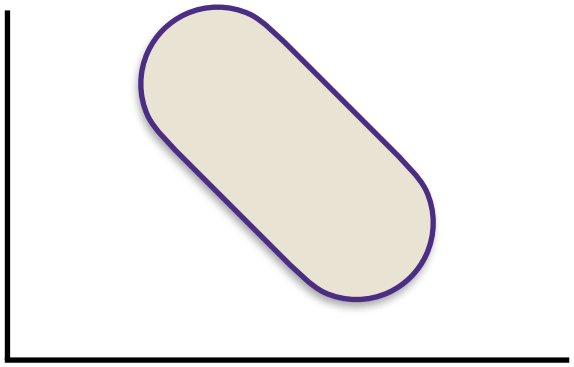
If I pick a learning rate η and observe the plot on the left, did I pick wrong?
What about the plot on the right?

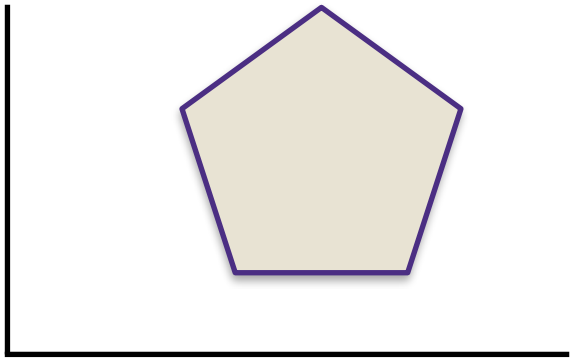
Convexity

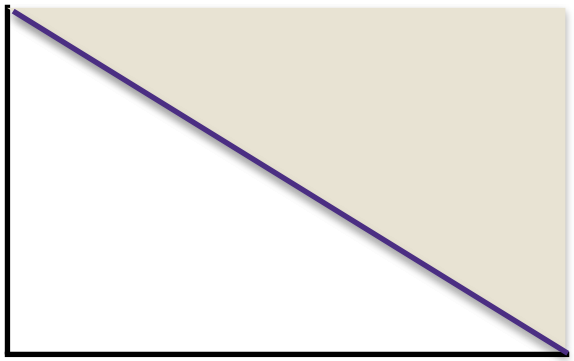
What is a convex set?

A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$

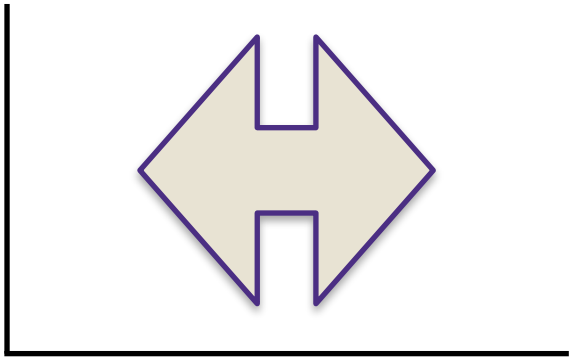
Examples of convex sets







Examples of non-convex functions: anything else



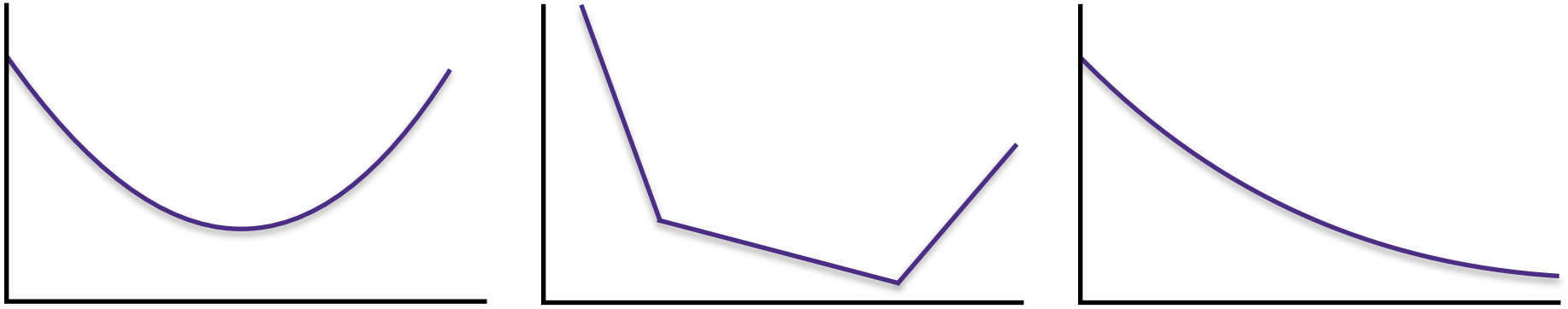




What is a convex function?

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in K$ and $\lambda \in [0, 1]$

Examples of convex functions: "look like bowls"



Examples of non-convex functions: anything else

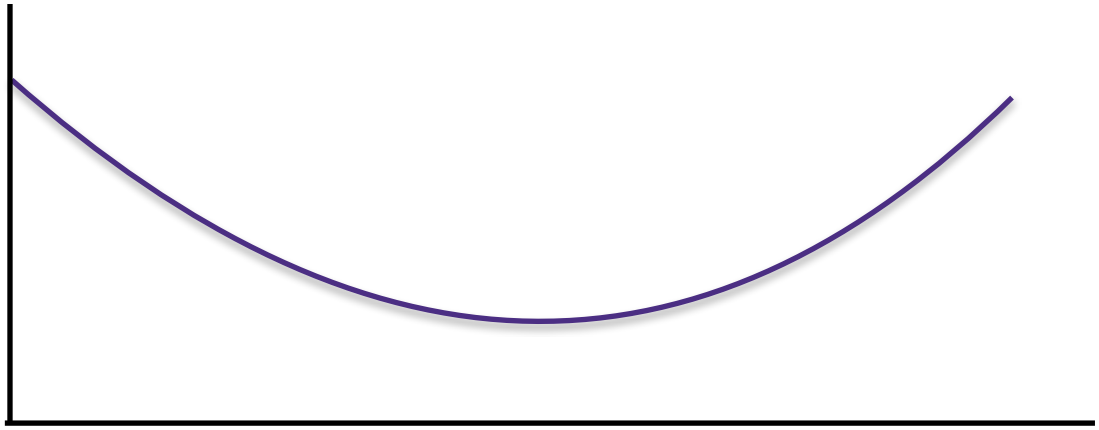


Convex functions and convex sets?

A set $K \subset \mathbb{R}^d$ is convex if $(1-\lambda)x + \lambda y \in K$ for all $x, y \in K$ and $\lambda \in [0,1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in K$ and $\lambda \in [0,1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if the set $\{(x,t) \in \mathbb{R}^{d+1} : f(x) \leq t\}$ is convex

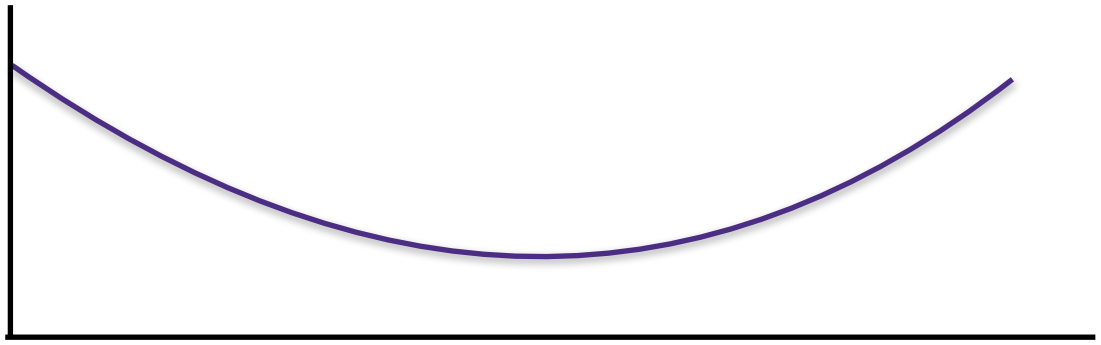


More definitions of convexity

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in K$ and $\lambda \in [0, 1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if the set $\{(x, t) \in \mathbb{R}^{d+1} : f(x) \leq t\}$ is convex

A function $f : \mathbb{R}^d \to \mathbb{R}$ that is differentiable everywhere is convex if $f(y) \geq f(x) + \nabla f(x)^\top (y - x)$ for all $x, y \in dom(f)$



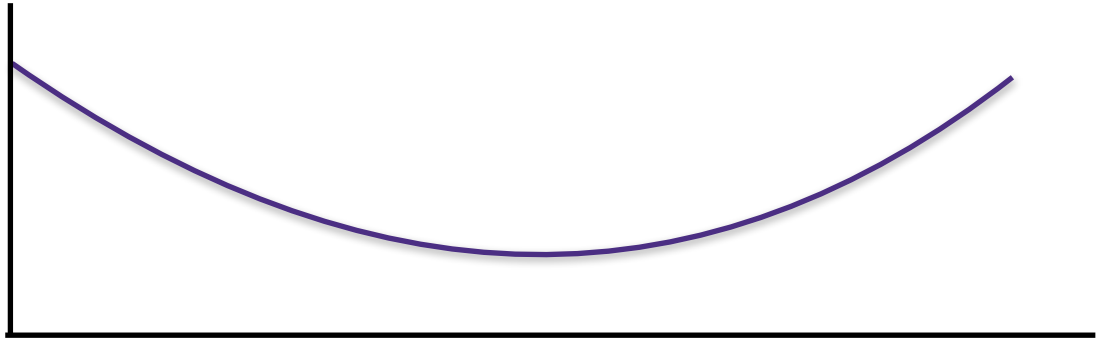
More definitions of convexity

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in K$ and $\lambda \in [0, 1]$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if the set $\{(x, t) \in \mathbb{R}^{d+1} : f(x) \leq t\}$ is convex

A function $f : \mathbb{R}^d \to \mathbb{R}$ that is differentiable everywhere is convex if $f(y) \geq f(x) + \nabla f(x)^\top (y - x)$ for all $x, y \in dom(f)$

A function $f : \mathbb{R}^d \to \mathbb{R}$ that is twice-differentiable everywhere is convex if $\nabla^2 f(x) \succeq 0$ for all $x \in dom(f)$

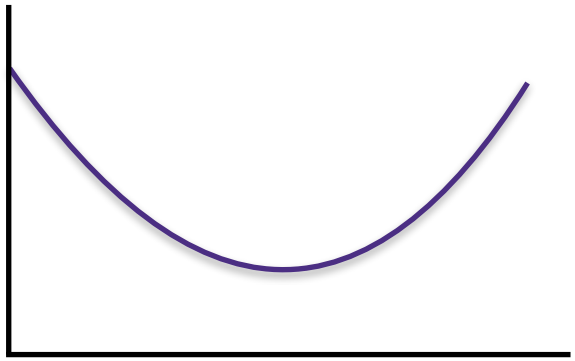


Why do we care about convexity?

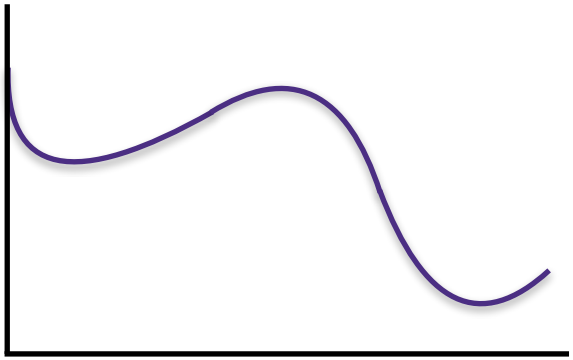
Convex functions

- All local minima are global minima
- Efficient to optimize (e.g., gradient descent)

Convex Function



Non-convex Function



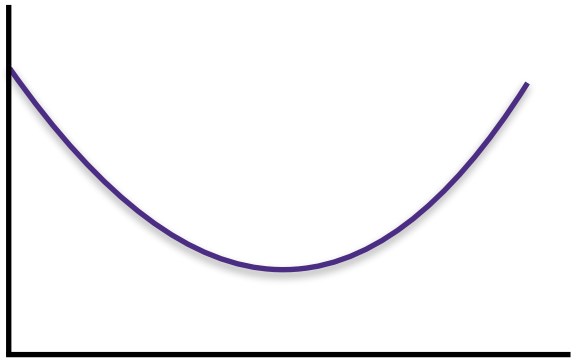
Gradient Descent

Initialize: $w_0 = 0$

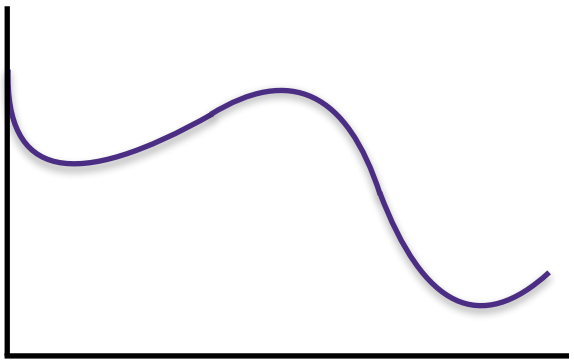
for $t = 1, 2, \ldots$

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

Convex Function



Non-convex Function



Sub-Gradient Descent

Initialize: $w_0 = 0$

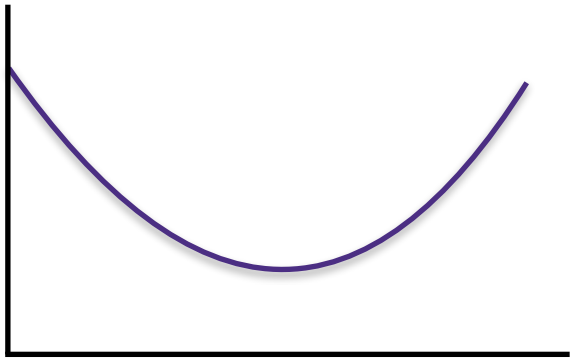
for $t = 1, 2, \ldots$

Find any g_t such that $f(y) \geq f(w_t) + g_t^\top (y - w_t)$

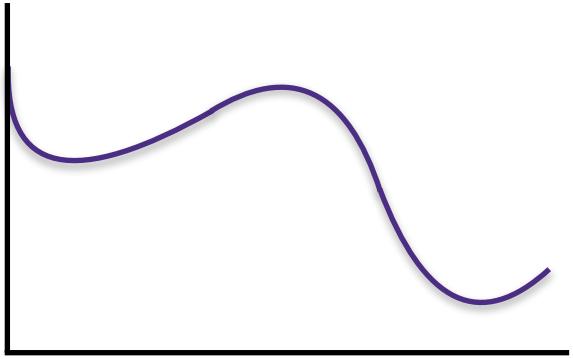
$w_{t+1} = w_t - \eta g_t$

g is a subgradient at x if $f(y) \geq f(x) + g^T (y - x)$

Convex Function



Non-convex Function



Coordinate descent

Initialize: $w_0 = 0$

for $t = 1, 2, \ldots$

Let $i_t = t \ \% \ n$

$$w_{t+1}^{(i_t)} = w_t^{(i_t)} - \eta_t \frac{\partial f(w)}{\partial w^{(i_t)}}\bigg|_{w=w_t}$$

Machine Learning Problems

- **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:** $\sum_{i=1}^{n} \ell_i(w)$

$$\text{Logistic Loss: } \ell_i(w) = \log(1 + \exp(-y_i\, x_i^T w))$$

$$\text{Squared error Loss: } \ell_i(w) = (y_i - x_i^T w)^2$$

Gradient Descent:

$$w_{t+1} = w_t - \eta \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} \ell_i(w) \right) \Big|_{w=w_t}$$

Optimization summary

- You can always run gradient descent whether f is convex or not. But you only have guarantees if f is convex
- Many bells and whistles can be added onto gradient descent such as momentum and dimension-specific step-sizes (Nesterov, Adagrad, ADAM, etc.)

Stochastic Gradient Descent

Gradient descent…

- **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:** $\sum_{i=1}^{n} \ell_i(w)$

Gradient Descent:

$$w_{t+1} = w_t - \eta \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} \ell_i(w) \right) \Big|_{w=w_t}$$

Gradient descent… meet stochastic GD

- **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:** $\sum_{i=1}^{n} \ell_i(w)$

Gradient Descent:

$$w_{t+1} = w_t - \eta \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} \ell_i(w) \right) \Big|_{w=w_t}$$

Stochastic Gradient Descent:

$$w_{t+1} = w_t - \eta \nabla_w \ell_{I_t}(w) \Big|_{w=w_t}$$

I_t drawn uniform at random from $\{1, \ldots, n\}$

$$\mathbb{E}[\nabla \ell_{I_t}(w)] = \frac{1}{n} \sum_{i=1}^{n} \ell_i(w) = \nabla \ell(w)$$

E[Stochastic Gradient descent] = GD

- **Learning a model's parameters:** $\sum_{i=1}^{n} \ell_i(w)$

Stochastic Gradient Descent:

$$w_{t+1} = w_t - \eta \nabla_w \ell_{I_t}(w)\Big|_{w=w_t}$$

I_t drawn uniform at random from $\{1, \ldots, n\}$

$$\mathbb{E}[\nabla \ell_{I_t}(w)] = \nabla \ell(w)$$

- **How do we argue about GD/SGD's convergence?**

Stochastic Gradient Descent Convergence

Theorem: For Convex loss functions,

Let $w_{t+1} = w_t - \eta \nabla_w \ell_{I_t}(w)\Big|_{w=w_t}$ I_t drawn uniform at random from $\{1, \ldots, n\}$ so that

$$\mathbb{E}\left[\nabla \ell_{I_t}(w)\right] = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_i(w) =: \nabla \ell(w)$$

If $\|w_1 - w_0\|_2^2 \le R$ and $\sup_w \max_i \|\nabla \ell_i(w)\|_2 \le G$ then

$$\mathbb{E}[\ell(\bar{w}) - \ell(w_*)] \le \frac{R}{2T\eta} + \frac{\eta G}{2} \le \sqrt{\frac{RG}{T}} \qquad \eta = \sqrt{\frac{R}{GT}}$$

$$\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$$

(In practice use last iterate)

SGD convergence outline

- Proof outline:
  - We'll argue that the gradient of the loss at time t is big if w_t's loss is big
  - Show that the weights at time t+1 are getting close to the optimum weights
    - By arguing about the update from time t to time t_+1

SGD Convergence: argue about update/gradients

$\mathbb{E}[\nabla_{I_t}(w)t)^T(w_t - w_*) \geq \mathbb{E}[\ell(w_t) - \ell(w_*)]$ (Our Lemma)

It'll help to argue our update moves us towards OPT quickly if our loss is far from OPT

$$\mathbb{E}[\nabla \ell_{I_t}(w_t)^T(w_t - w_*)] = \mathbb{E}\big[\mathbb{E}[\nabla \ell_{I_t}(w_t)^T(w_t - w_*)|I_1, w_1, \ldots, I_{t-1}, w_{t-1}]\big]$$

$$= \mathbb{E}\big[\nabla \ell(w_t)^T(w_t - w_*)\big]$$

E1 wrt I_t, E2 wrt I_1, … I_t-1

$$\geq \mathbb{E}\big[\ell(w_t) - \ell(w_*)\big]$$

We argued before

Convexity of ℓ

SGD Convergence: Argue weights converge

$\mathbb{E}[\nabla_{I_t}(w)t)^T(w_t - w_*) \geq \mathbb{E}[\ell(w_t) - \ell(w_*)]$ (Our Lemma)

Now, we argue that after t steps, our weights are pretty close to OPT:

$$\mathbb{E}[||w_{t+1} - w_*||_2^2] = \mathbb{E}[||w_t - \eta\nabla\ell_{I_t}(w_t) - w_*||_2^2]$$ # Def of SGD update

$$= \mathbb{E}[||w_t - w_*||_2^2] - 2\eta\mathbb{E}[\nabla\ell_{I_t}(w_t)^T(w_t - w_*)] + \eta^2\mathbb{E}[||\nabla\ell_{I_t}(w_t)||_2^2]$$ # expand ^2

$$\leq \mathbb{E}[||w_t - w_*||_2^2 - 2\eta\mathbb{E}[\ell(w_t) - \ell(w_*)|| + \eta^2\mathbb{E}[||\nabla\ell_{I_t}(w_t)||_2^2]$$ # Our Lemma

$$\leq \mathbb{E}[||w_t - w_*||_2^2] - 2\eta\mathbb{E}[\ell(w_t) - \ell(w_*)] + \eta^2 G$$ # Upper bound we assumed on our gradients

SGD Convergence: Argue about average weights

$\mathbb{E}[\nabla_{I_t}(w)t)^T(w_t - w_*) \geq \mathbb{E}[\ell(w_t) - \ell(w_*)]$ (Our Lemma)

$\mathbb{E}[||w_{t+1} - w_*||_2^2] \leq \mathbb{E}[||w_t - w_*||_2^2] - 2\eta\mathbb{E}[\ell(w_t) - \ell(w_*)] + \eta^2 G$ (Our Weights Lemma)

$2\eta\mathbb{E}[\ell(w_t) - \ell(w_*)] \leq \mathbb{E}[||w_t - w_*||_2^2] + \eta^2 G - \mathbb{E}[||w_{t+1} - w_*||_2^2]$ (Our Weights Lemma, rear

$$\sum_{t=1}^{T} \mathbb{E}[\ell(w_t) - \ell(w_*)] \leq \frac{1}{2\eta}\left(\mathbb{E}[||w_1 - w_*||_2^2] - \mathbb{E}[||w_{T+1} - w_*||_2^2] + T\eta^2 G\right)$$

$$\leq \frac{R}{2\eta} + \frac{T\eta G}{2}$$

Ignoring the second term and using our assmpt on how close we were to begin with

Stochastic Gradient Descent

Proof

Jensen's inequality:
For any random $Z \in \mathbb{R}^d$ and convex function $\phi : \mathbb{R}^d \to \mathbb{R}$, $\phi(\mathbb{E}[Z]) \leq \mathbb{E}[\phi(Z)]$

$$\mathbb{E}[\ell(\bar{w}) - \ell(w_*)] \leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\ell(w_t) - \ell(w_*)] \qquad \bar{w} = \frac{1}{T} \sum_{t=1}^{T} w_t$$

Stochastic Gradient Descent

Proof

Jensen's inequality:
For any random $Z \in \mathbb{R}^d$ and convex function $\phi : \mathbb{R}^d \to \mathbb{R}$, $\phi(\mathbb{E}[Z]) \leq \mathbb{E}[\phi(Z)]$

$$\mathbb{E}[\ell(\bar{w}) - \ell(w_*)] \leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\ell(w_t) - \ell(w_*)] \qquad \bar{w} = \frac{1}{T} \sum_{t=1}^{T} w_t$$

$$\mathbb{E}[\ell(\bar{w}) - \ell(w_*)] \leq \frac{R}{2T\eta} + \frac{\eta G}{2} \leq \sqrt{\frac{RG}{T}} \qquad \eta = \sqrt{\frac{R}{GT}}$$

Mini-batch SGD

Instead of one iterate, average B stochastic gradient together

Advantages:

- de-noises gradient
- Matrix computations
- Parallelization