6.7900 Machine Learning (Fall 2023)

Lecture III:

Optimization and Regularization (supporting slides)

Outline for Today

- Optimization
  - Set up and terminology
  - Convex and strictly convex functions
  - (Stochastic) gradient descent

- Regression and regularized regression
  - Ordinary, ridge, lasso regression
  - Other regularizers and interpretations

- Regularization
  - Mitigating training (optimization) and testing (statistics)
  - Explicit regularization
  - Implicit regularization

References

- Optimization and (S)GD:
  - Convex Optimization [Boyd and Vandenberghe]
  - Introduction to Optimization, [Chong and Zak], especially Chapter 8.
- Ridge/lasso/explicit regularization:
  - Pattern Recognition and Machine Learning, [Bishop]
  - Referenced on slides
- Implicit regularization:
  - Dropout (Srivastava et al., 2014)
  - Label smoothing (Szegedy et al 2016)
  - Early stopping (Caruana et al., 2001)
  - Gradient Descent Only Converges to Minimizers (Lee et al, 2019)
- Some slides edited from: Tamara Broderick, Stephen Boyd, and Suvrit Sr a

What is Empirical Risk Minimization?

Learner does not know $\mathbb{P}(X, Y)$, so true error (Bayes error) is not known to the learner. However,

- **Training Error**: The error that the classifier incurs on the training data

$$L_S(h) := \frac{1}{N} \# \{i \in [N] \mid h(x_i) \neq y_i\} ,$$

aka *empirical risk*

- **ERM principle**: Seek predictor that minimizes $L_S(h)$
- **Pitfall**: Overfitting!

Optimization Terminology

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

(Global) **optimal solution** $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

Accommodates maximization:

$$\begin{array}{ll} \text{maximize} & -f_0(x) \\ \text{subject to} & f_i(x) \leq b_i \end{array}$$

Accommodates unconstrained:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & 0 \leq 0 \end{array}$$

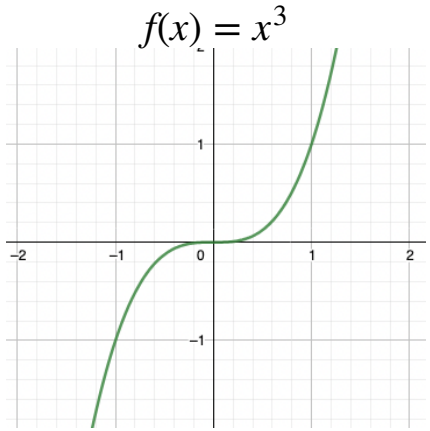
Unconstrained more heavily used in numerical solvers and modern ML

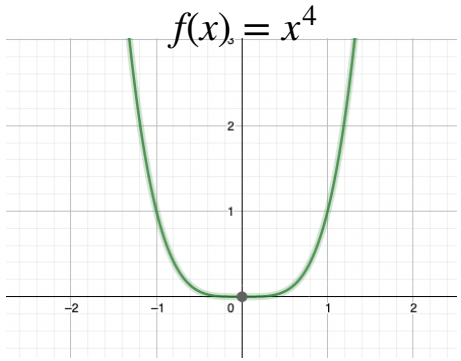
Anatomy

- Feasible solution(s): Any x that satisfies all constraints $f_i(x) \leq b_i$
- Fixed points: Any x where $\nabla f(x) = 0$
- Local optimal solutions: Any x among feasible solutions that's smaller than its neighbors
- (Global) optimal solutions: Any x among feasible solutions that's globally minimum
- Optimal value: the objective function evaluated at an optimal solution $f_0(x^*)$

Unconstrained Local Optimality Condition

FONC (First order necessary condition)	x^* is a local minimizer	$\Longrightarrow$ $\not\Longleftarrow$	$\nabla f(x^*) = 0$ example: $f(x) = x^3$ at 0
SONC (Second order necessary condition	x^* is a local minimizer	$\Longrightarrow$ $\not\Longleftarrow$	$\nabla f(x^*) = 0$ $\& \nabla^2 f(x^*) \succeq 0$ example: $f(x) = x^3$ at 0
SOSC (Second order sufficient condition)	x^* is a (strict) local minimizer	$\Longleftarrow$ $\not\Longrightarrow$	$\nabla f(x^*) = 0$ $\& \nabla^2 f(x^*) \succ 0$ example: $f(x) = x^4$ at 0





Positive Semidefinite Matrices

- Definition: An $n \times n$ symmetric real matrix A is said to be positive semidefinite (i.e., $A \succeq 0$) if $x^T A x \geq 0$ for all x in $\mathbb{R}^n$.
- Or, equivalently:
  - All eigenvalues of A are non-negative.
  - There exists a factorization $A = B^T B$.
  - All $2^n - 1$ principal minors of A are nonnegative
- e.g.

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix}$$

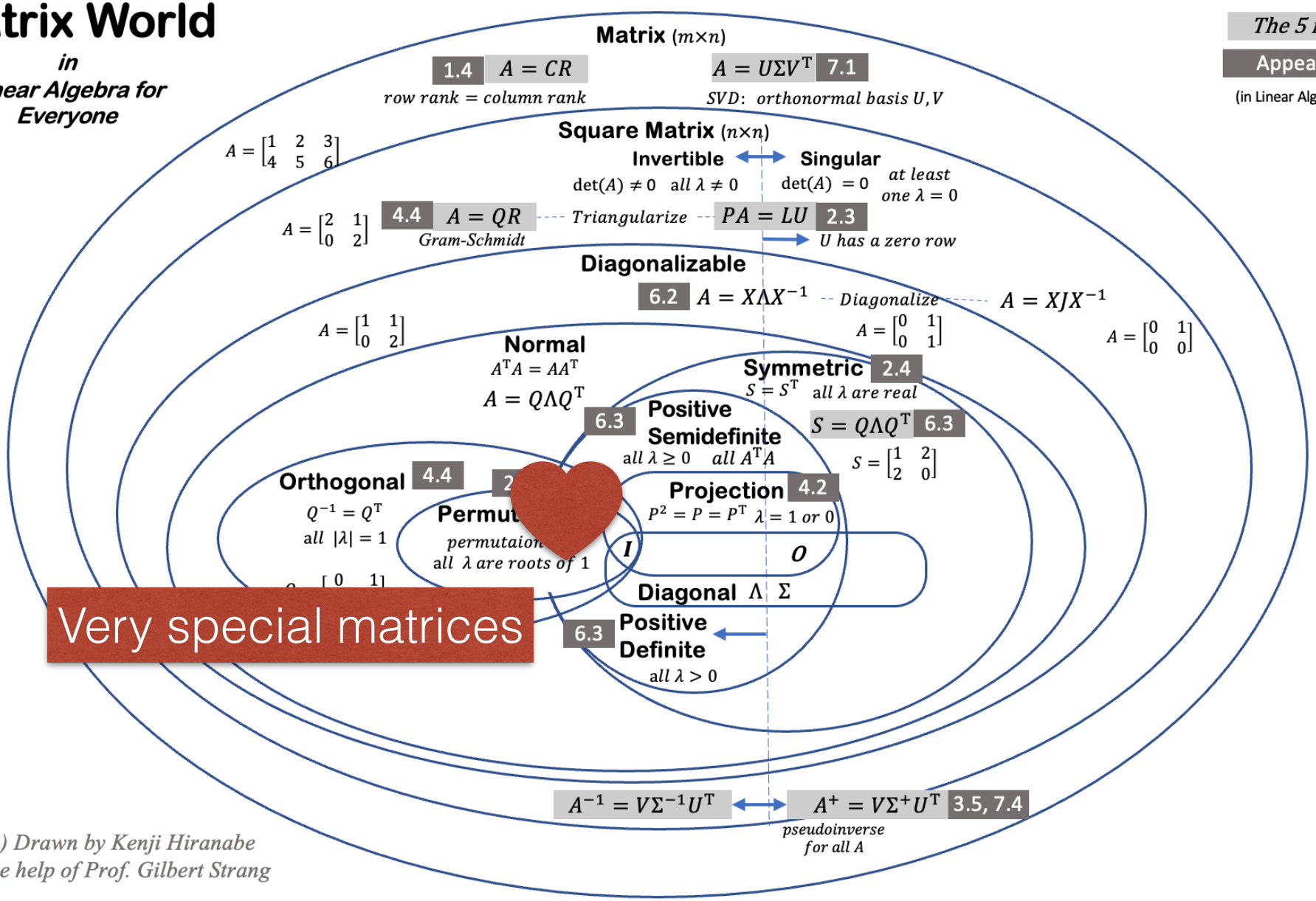
Positive Definite Matrices

- Definition: An $n \times n$ symmetric real matrix A is said to be positive definite (i.e., $A \succ 0$) if $x^T A x > 0$ for all x in $\mathbb{R}^n$ and $x \neq 0$.
- Or, equivalently:
  - All eigenvalues of A are positive.
  - There exists a factorization $A = B^T B$ where B is square and non-singular.
  - All n **leading** principal minors of A are positive.
- e.g.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 7 & 2 \\ 2 & 1 \end{bmatrix}$$

Matrix World

in
Linear Algebra for Everyone



Very special matrices

The 5 factorization

Appearing section

(in Linear Algebra for Everyone)

(v1.4.3) Drawn by Kenji Hiranabe
with the help of Prof. Gilbert Strang

Global Optimality Condition

- If objective/constraints are convex functions, any local min is global min.

- Mainly why convexity is so beloved in optimization
- When is a function convex and how do we check for it?

$$\text{A function } f : \mathbb{R}^n \to \mathbb{R} \text{ is convex if its domain is a convex set and}$$
$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \forall x, y \in \text{domain}(f), \forall \lambda \in [0,1]$$

- Equivalent (sometimes easier to check) condition:

$\nabla^2 f(x) \succeq 0, \quad \forall x \in \text{dom}(f)$ (i.e., the Hessian is psd $\forall x \in \text{dom}(f)$)

- [demo]

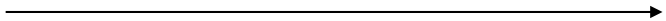
Unconstrained Global Optimality Condition

- Generally, no "easy" way to **check** global optimality (let alone **find** solutions).
- **Convex** functions are a major class of exceptions to the above.

Classical problems
Almost all convex

Deep learning era
Almost none convex

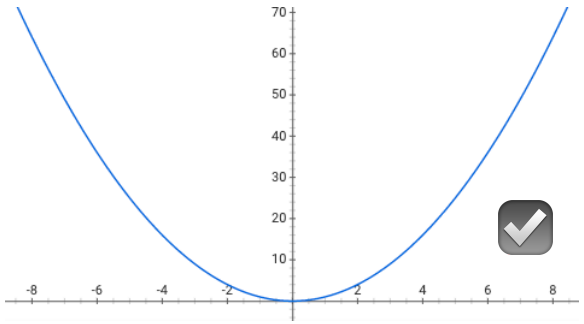
ML timeline

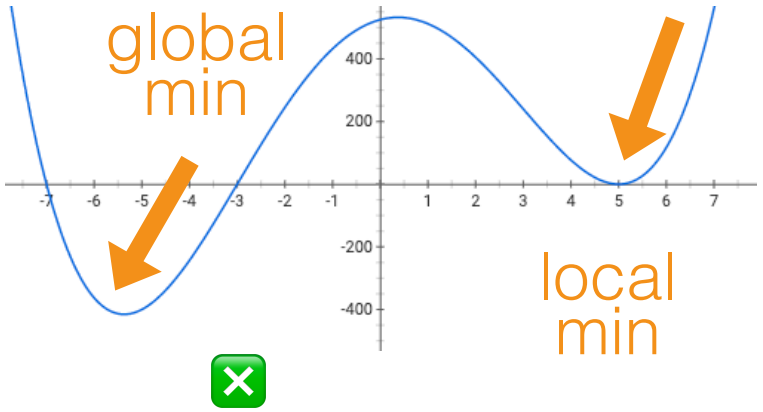


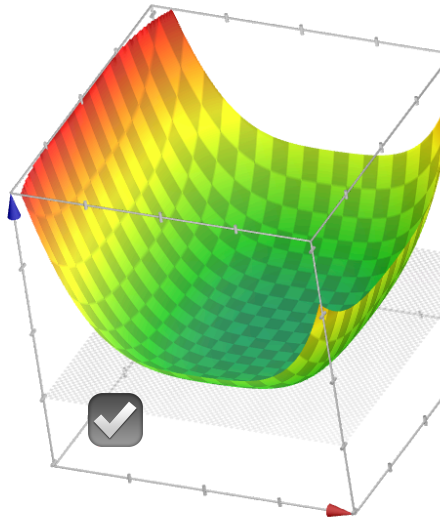
- Ongoing research on over-parameterization, local vs global min, implicit convexity, in deep learning
- (Explicit) regularization is usually done by "injecting more convexity". So let's understand convexity a bit.

Convex Functions

- A function f on $\mathbb{R}^m$ is convex if any line segment connecting two points of the graph of f lies above or on the graph



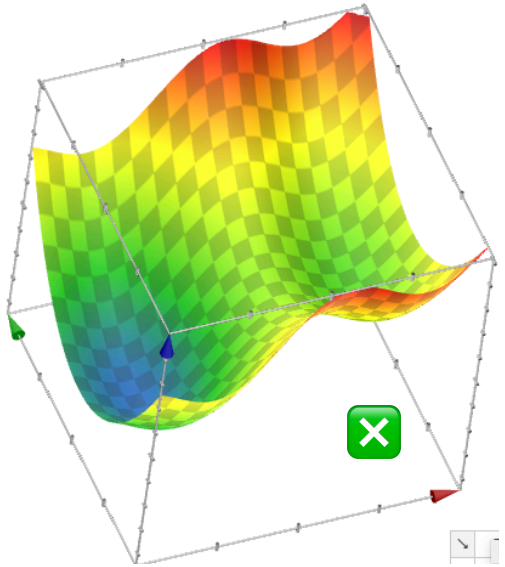




[demo]

Convex functions are important because:

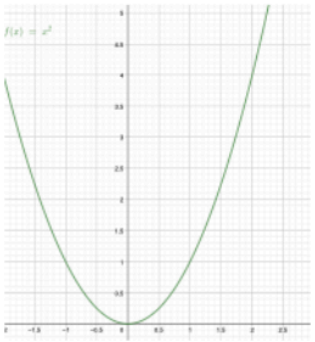
Every local minimizer is a global minimizer.

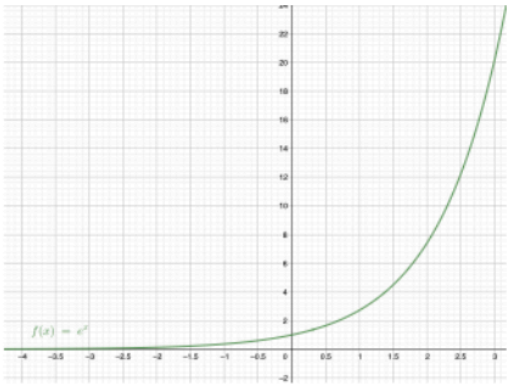


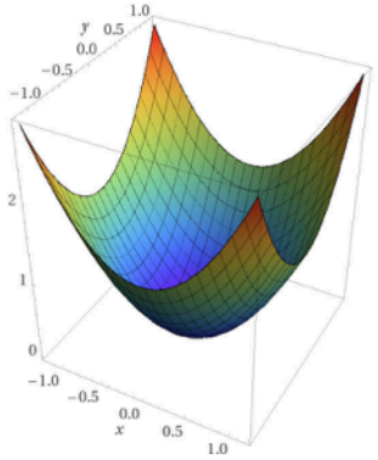
Convex Functions

Simple examples

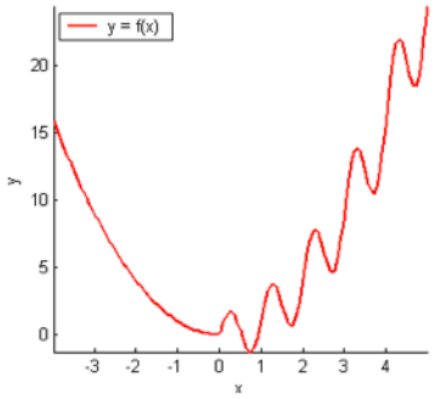
Convex functions

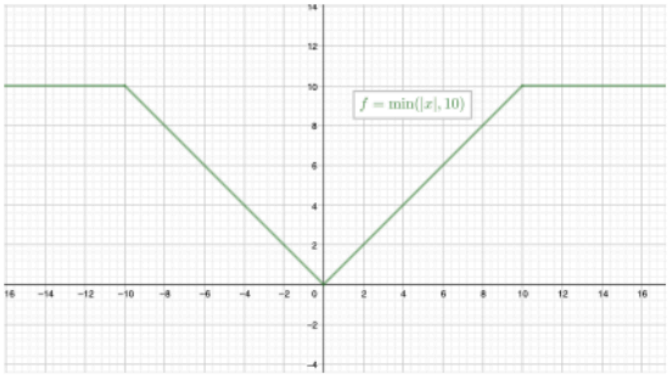


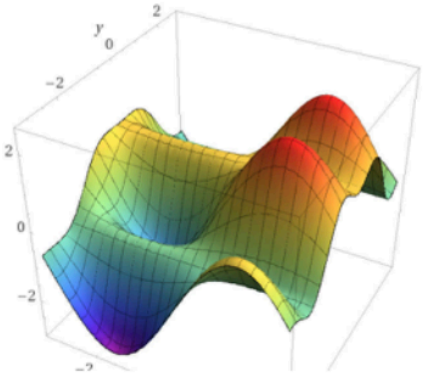




Non-convex functions



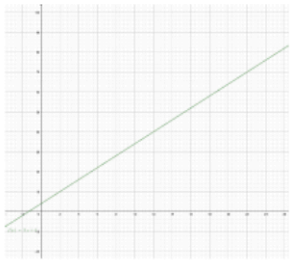


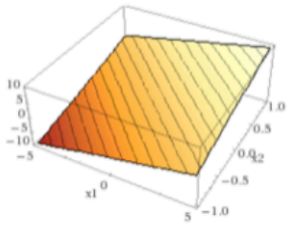


f is called a concave function if $-f$ is convex

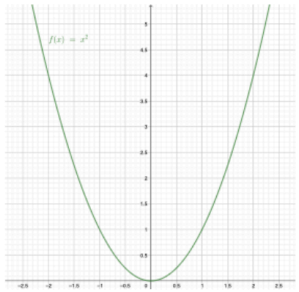
Common Convex Functions

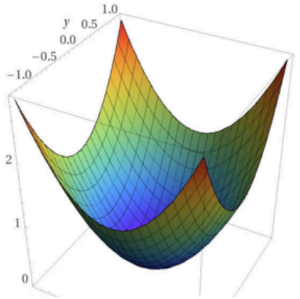
- All linear(affine) functions $f(x) = a^T x + b$ (for any $a \in \mathbb{R}^n, b \in \mathbb{R}$)





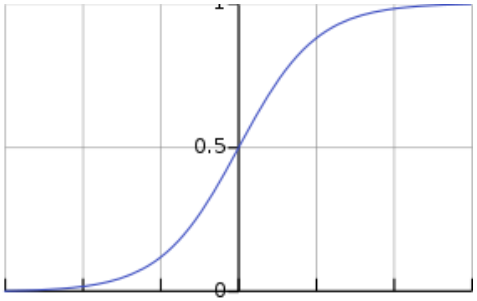
- **Some** quadratic functions





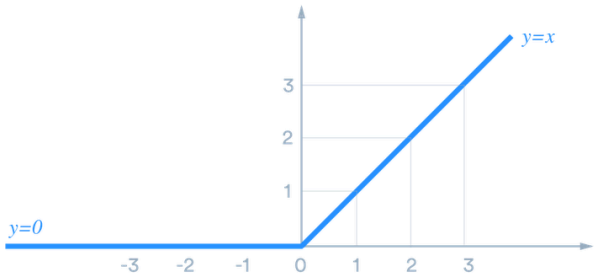
- All norms

- Is a Sigmoid $f(x) = \frac{1}{1 + e^{-x}}$ convex?



❌

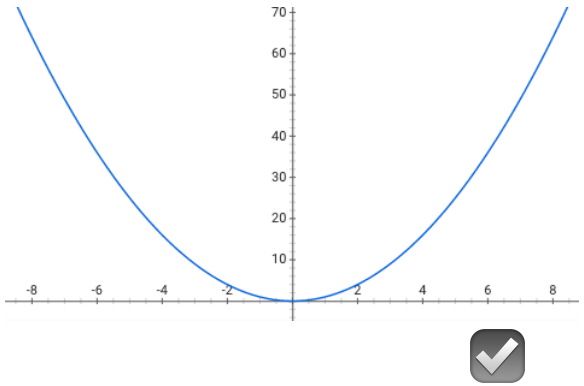
- Is ReLU $f(x) = \max(0,x)$ convex?

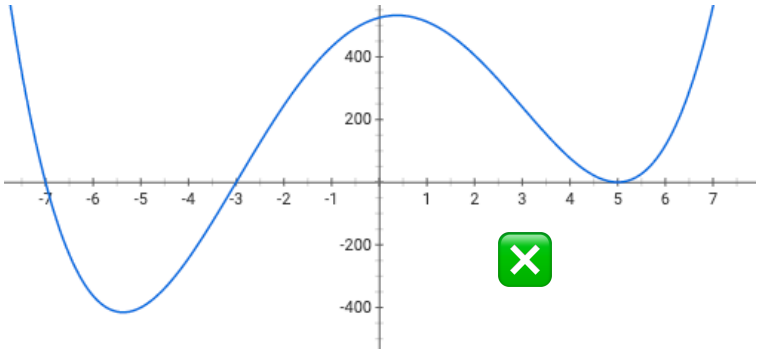


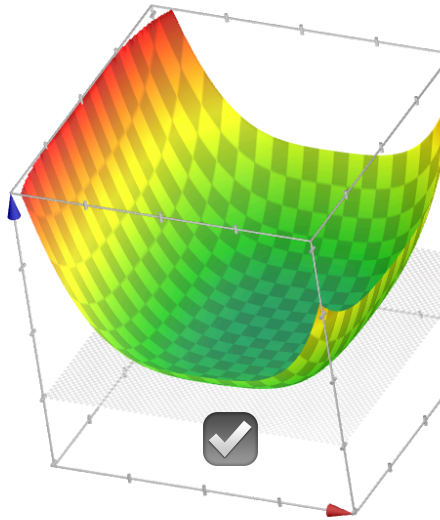
✔️

Strictly Convex Functions

- A function f on $\mathbb{R}^m$ is convex if any line segment connecting two points of the graph of f lies above ~~or on~~ the graph

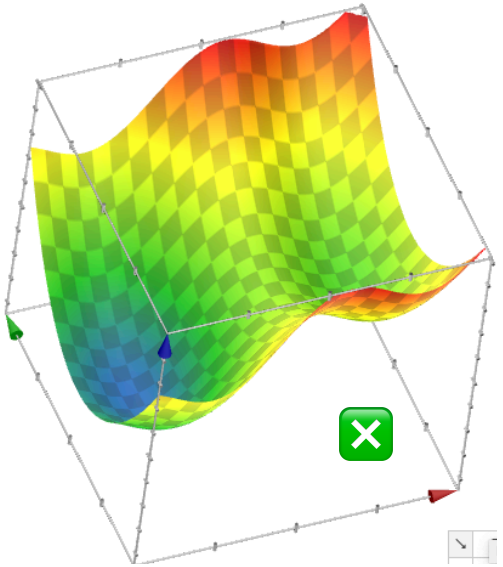






Strictly convex functions are important because:

- Every local minimizer is a global minimizer.
- There can be only one unique local/global min.
- Better theoretical properties (e.g., convergence rate).



[Quadratic function demo]

(Stochastic) Gradient Descent

Iteratively applies "gradient vector points to the direction where the function value increases the fastest"

And hoping to get

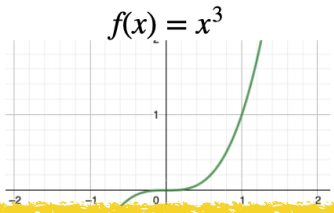
Unconstrained (Local) Optimality

FONC (First order necessary condition)

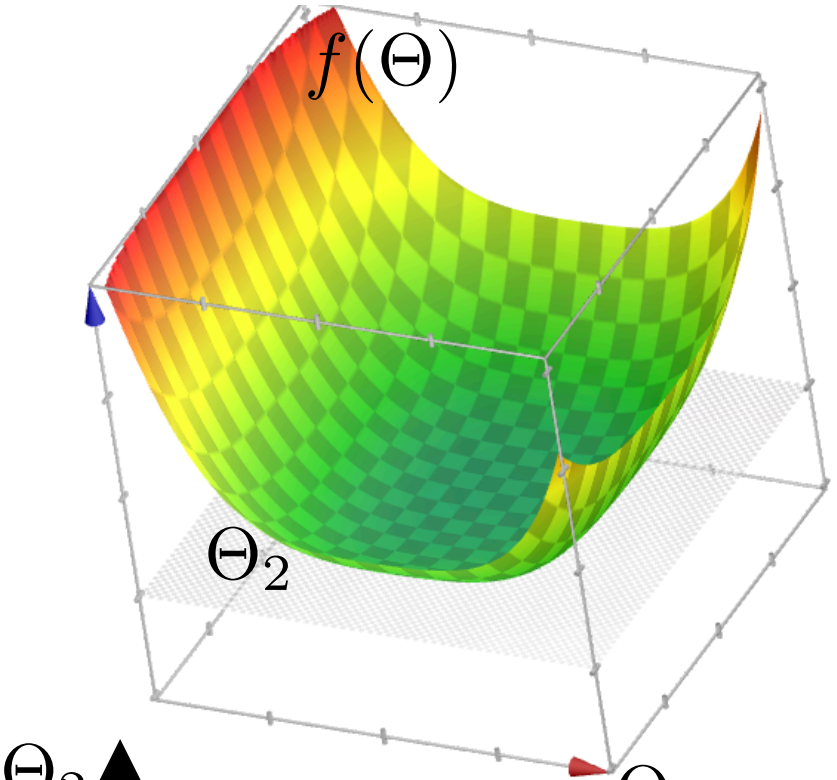
x^* is a local minimizer $\Longrightarrow$ $\nabla f(x^*) = 0$

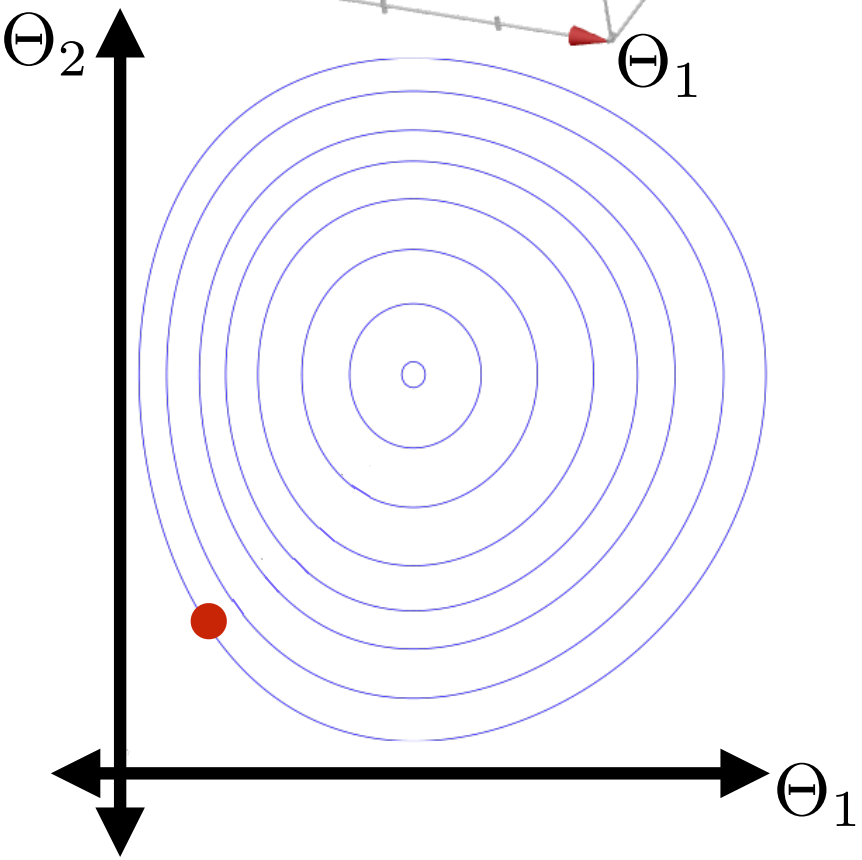
$\nLeftarrow$

example: $f(x) = x^3$ at 0



Gradient descent





Gradient descent

- Gradient $\nabla_\Theta f = \left[\frac{\partial f}{\partial \Theta_1}, \ldots, \frac{\partial f}{\partial \Theta_m}\right]$
  - with $\Theta \in \mathbb{R}^m$

`Gradient-Descent` $(\Theta_{\text{init}}, \eta, f, \nabla_\Theta f, \epsilon$

`Initialize` $\Theta^{(0)} = \Theta_{\text{init}}$

`Initialize t = 0`

`repeat`

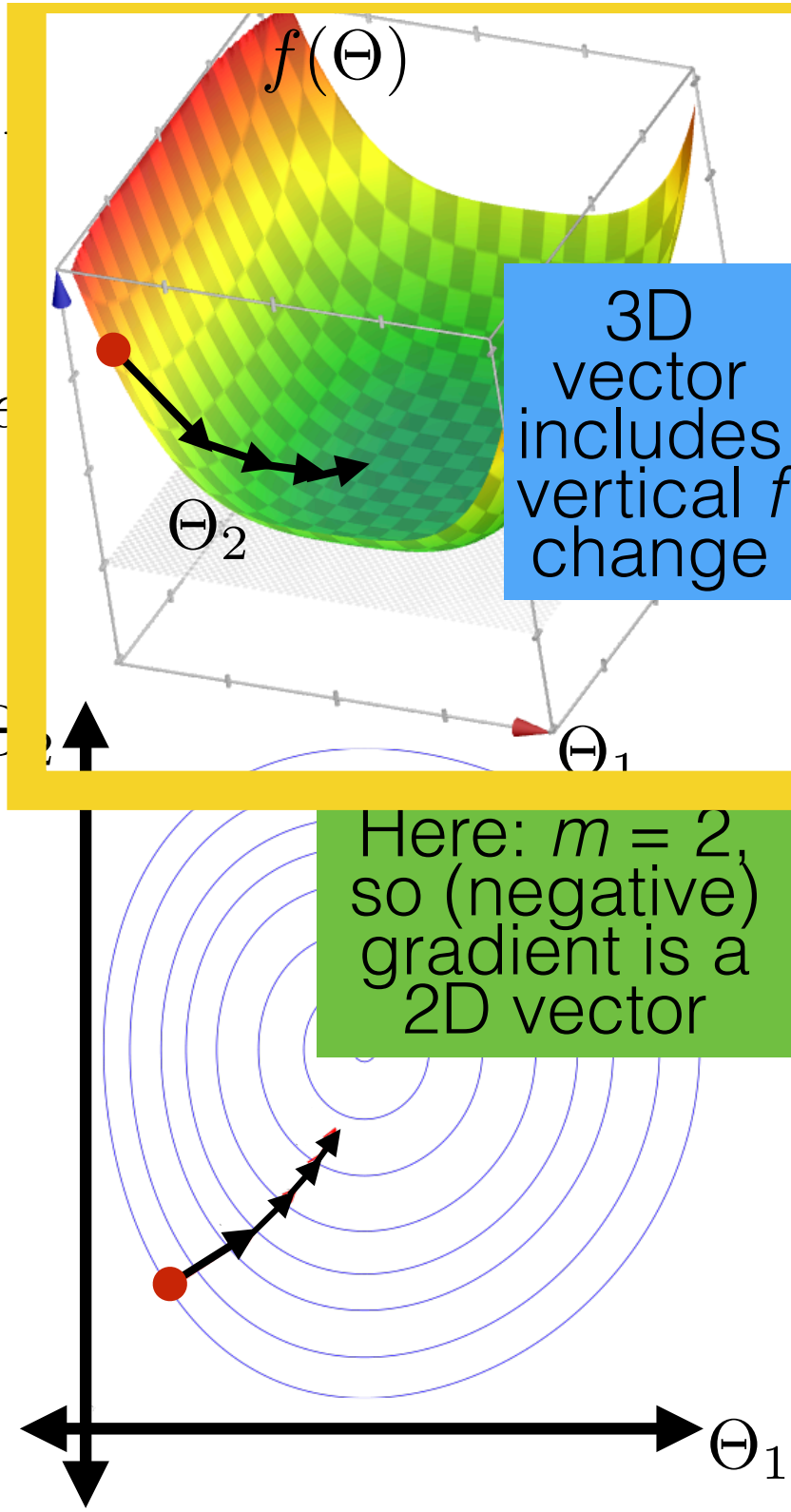
`t = t + 1`

$\Theta^{(t)} = \Theta^{(t-1)} - \eta \nabla_\Theta f(\Theta^{(t-1)})$

`until` $\left| f(\Theta^{(t)}) - f(\Theta^{(t-1)}) \right| < \epsilon$

`Return` $\Theta^{(t)}$

- Other possible stopping criteria:
  - Max number of iterations *T*
  - $\|\Theta^{(t)} - \Theta^{(t-1)}\| < \epsilon$
  - $\|\nabla_\Theta f(\Theta^{(t)})\| < \epsilon$



Stochastic gradient descent

- ERM or training error typically can be written as:

$$f(\Theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\Theta)$$

Stochastic-Gradient-Descent($\Theta_{\text{init}}, \eta, T$)

  Initialize $\Theta^{(0)} = \Theta_{\text{init}}$

  for t = 1 **to** T

    randomly select i from {1,…,n} (with equal probability)

    $\Theta^{(t)} = \Theta^{(t-1)} - \eta(t)\nabla_\Theta f_i(\Theta^{(t-1)})$

  Return $\Theta^{(t)}$

Compare to gradient descent update:

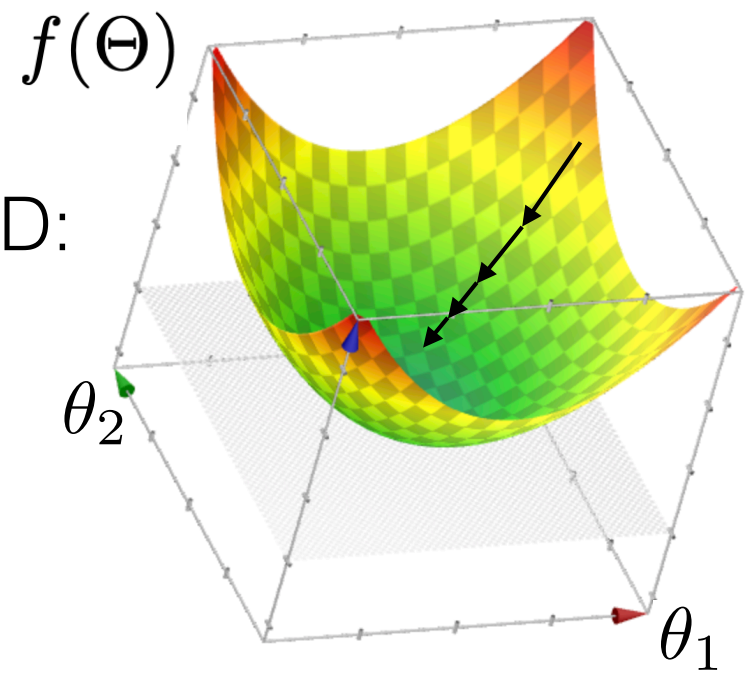
$$\Theta^{(t)} = \Theta^{(t-1)} - \eta\nabla_\Theta f(\Theta^{(t-1)})$$

GD vs SGD

Compare to gradient descent update:

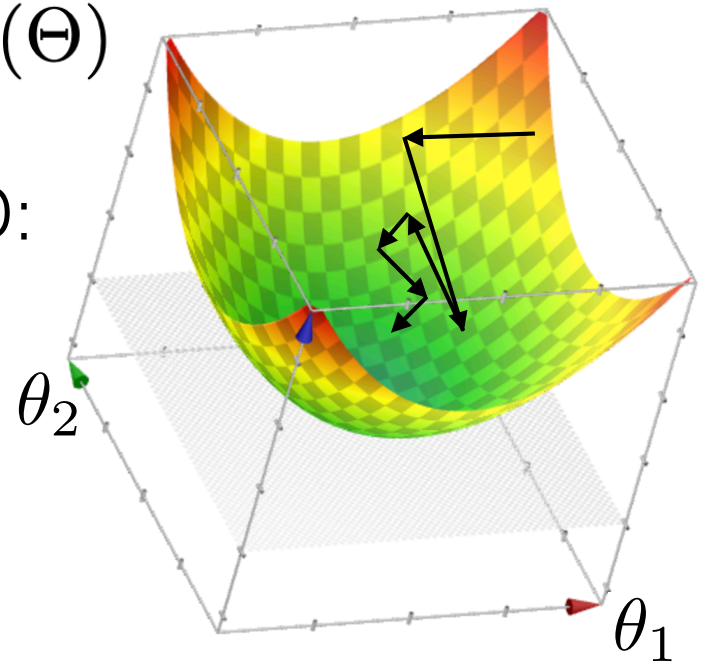
$$\Theta^{(t)} = \Theta^{(t-1)} - \eta \nabla_\Theta f(\Theta^{(t-1)})$$

- GD:



$$\Theta^{(t)} = \Theta^{(t-1)} - \eta(t) \nabla_\Theta f_i(\Theta^{(t-1)})$$

- SGD:



Quick Summary

- Optimality conditions
- Convexity and strong convexity
- GD and SGD
- Quick statements:
  - SGD on general functions: wild wild world; no guarantee whatsoever.
  - SGD on convex functions: with step-sizing annealing, can be shown to converge to local/global min.
  - GD on convex functions: can converge to local/glocal min with appropriately chosen fixed step size.
  - GD on strongly convex functions: same as above; additionally, easier step-size calculation, faster convergence, and converges to unique global min.

Regression and Regularized Regression

Ordinary, ridge, lasso, and interpretations

Ordinary Linear Least Squares (OLS)

Given training data $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where $x \in \mathbb{R}^d, y \in \mathbb{R}$

$$\min_w \quad L(w) := \sum_i (y_i - w^T x_i)^2 = \|Xw - y\|^2$$

$$X \in \mathbb{R}^{N \times d}, \; y \in \mathbb{R}^N, w \in \mathbb{R}^d$$

$$L(w) = w^T X^T X w - 2w^T X^T y + y^T y$$
$$\nabla L(w) = 2X^T X w - 2X^T y$$

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

Exercise: Observe that if using nonlinear features $\phi(x)$, we obtain $(\Phi^T \Phi)^{-1}$

Question: What if $d > N$?

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

Rank deficiency; no longer invertible

Exercise: If $d \leq N$, are there any situations under which we still would lose invertibility?

Yes, if there's so-called colinearity among features, we still lose invertibility

What about a linear algebra trick?

Trick: Replace $X^T X \mapsto X^T X + \lambda I$

(since $X^T X$ is positive semidefinite, adding λI with $\lambda > 0$ guarantees invertibility, refer to recitation 1)

$$w = (X^T X + \lambda I)^{-1} X^T y$$

"Nudge" to makes $X^T X$ non-singular – this was the original motivation for ridge regression (Hoerl and Kennard, 1970)

Ridge Regression: regularized least squares

Given training data $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where $x \in \mathbb{R}^d, y \in \mathbb{R}$

$$\min_w \quad L(w) := \|Xw - y\|^2 + \lambda\|w\|^2$$

$$X \in \mathbb{R}^{N \times d},\ y \in \mathbb{R}^N, w \in \mathbb{R}^d, \lambda > 0$$

$$w = (X^T X + \lambda I)^{-1} X^T y$$

Q: This regularization also called “weight-decay”. Why?

Importantly, adding a

$\lambda\|w\|^2$

$\lambda > 0$

makes the objective function having a unique solution. (How?)

Other Forms/Norms of Regularization

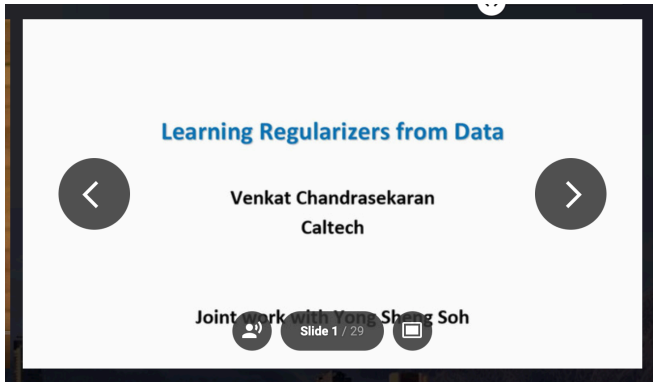
$$\min_w \quad \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2 + \lambda \|w\|_p^p$$

p=2: Ridge regression; **p=1: LASSO**

$$\min_w \quad \frac{1}{N}\sum_{i=1}^{N}(y_i - w^T x_i)^2 + \lambda \Omega(w)$$

Ω: norm, nuclear norm, atomic norm, and many others!

Food for thought: Which regularizer should we use? when? why?



1-dimensional for insight

Ridge leads to "shrinkage"

$$\text{minimize} \quad (y-w)^2 + \lambda w^2 \quad \Rightarrow \quad w = \frac{y}{1+\lambda}$$

L1-reg causes "thresholding"

$$\text{minimize} \quad (y-w)^2 + \lambda |w|$$

$$w = \begin{cases} y - \frac{\lambda}{2} & \text{if } y > \frac{\lambda}{2} \\ y + \frac{\lambda}{2} & \text{if } y < -\frac{\lambda}{2} \\ 0 & \text{if } y \in [-\frac{\lambda}{2}, \frac{\lambda}{2}] \end{cases}$$

Thus, small values are pushed to 0. Because of this property, it is widely used for obtaining "sparse solutions"

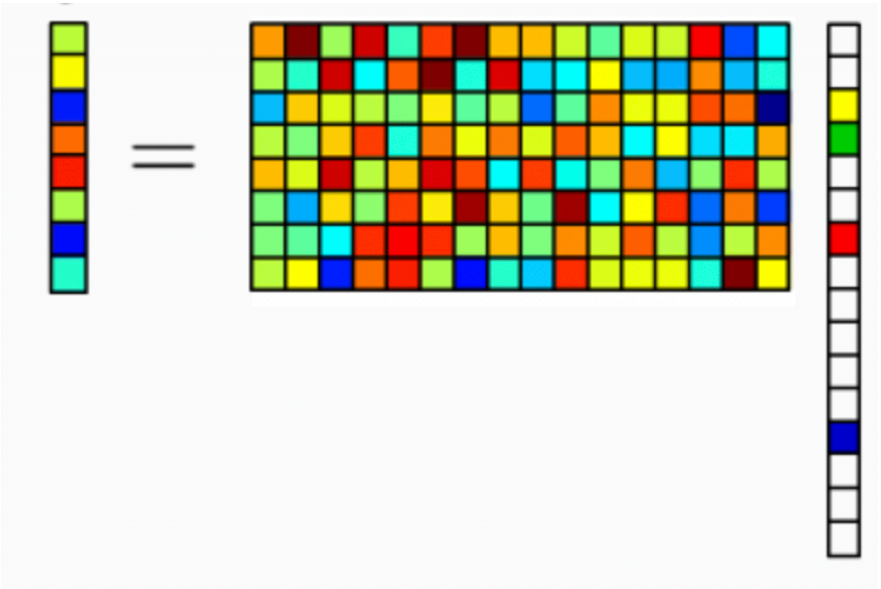
L1-norm regularization: sparsity

- LASSO = Least Absolute Shrinkage and Selection Operator
- Automated selection of “relevant features”
- A large number of features is useful to capture complex models, e.g.
  - variety of representations for capturing structure of image
  - or, higher order polynomials
- But limited data does not allow meaningful selection
- Regularization like Ridge Regression tends to select everything
- LASSO, on the other hand, tries to choose **sparsest** model parameter

L1-regularization: optimization interpretation

Let $\mathbf{w}$ be a vector in $\mathbb{R}^n$. We define the ℓ_0 pseudo-norm by:

$$\|\mathbf{w}\|_0 = \#\{i : \mathbf{w}_i \neq 0\}$$



L1-regularization
Can be thought of as a convex relaxation to
L0 pseudo-norm

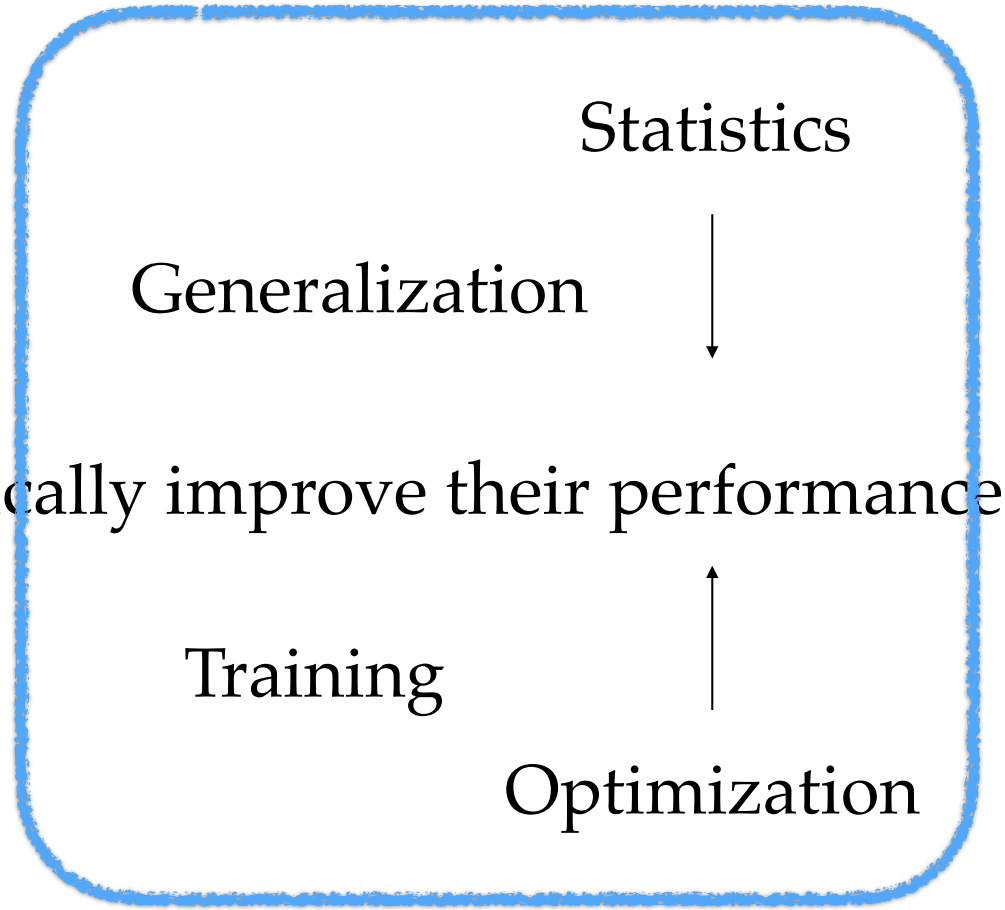
Similar dea generalizes to matrix world too:

- Matrix l_0 pseudo-norm: rank(A)
- Matrix $l_1$norm (nuclear norm): $\|A\|_* = \text{trace}\left(\sqrt{A^*A}\right) = \sum \sigma_i(A)$

https://en.wikipedia.org/wiki/Compressed_sensing

Regularization: the big picture role

"ML is concerned with computer programs that



regularization (mitigator)

automatically improve their performance through experience."

Probability (Data)

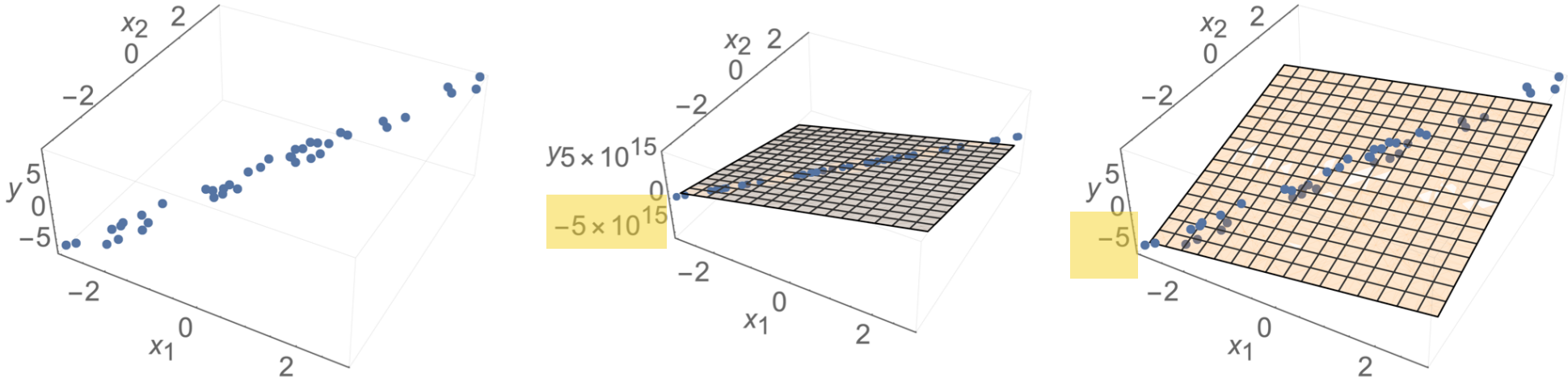
Regularization: Curb your complexity

Here we seek to minimize the *regularized empirical risk*

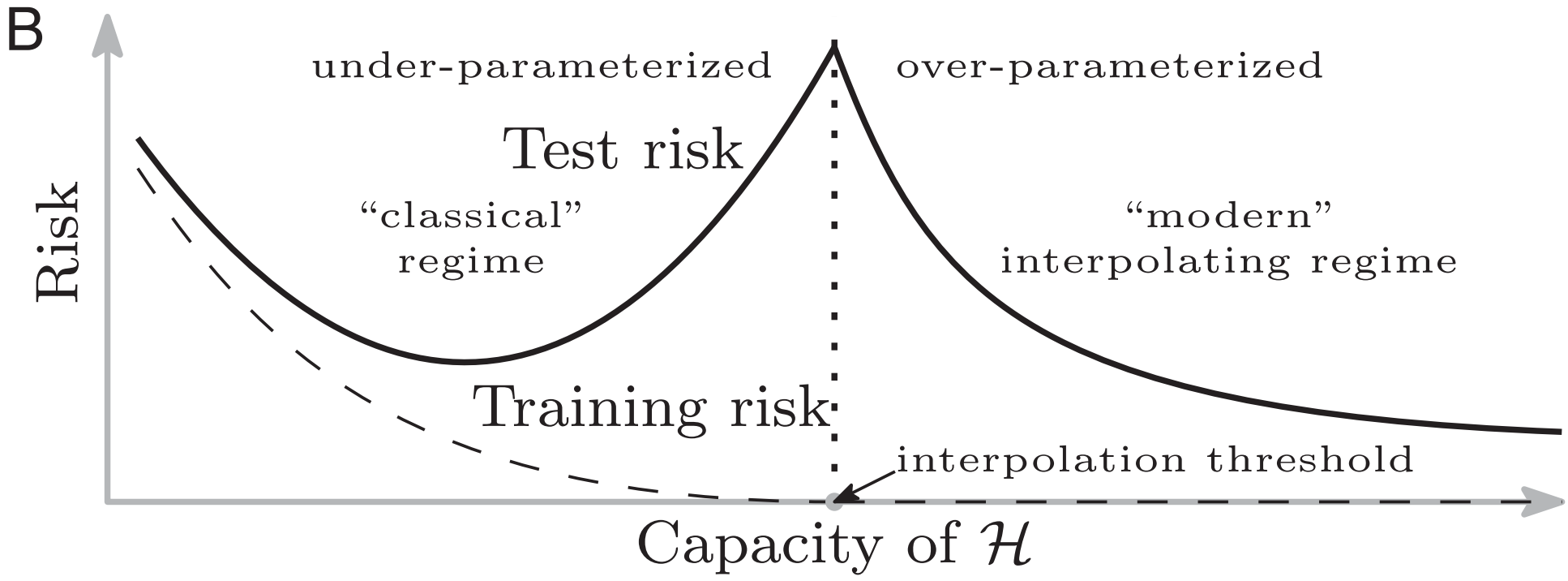
$$\min_{h \in \mathcal{H}} \quad L_S(h) + \lambda R(h),$$

where $\lambda \geq 0$ is a hyper-paramter that regulates the bias-complexity tradeoff.

How?



ERM: Bias-Complexity Tradeoff



"Modern" viewpoint on generalization: the double-descent curve

Reconciling modern machine-learning practice and the classical bias–variance trade-off

Mikhail Belkin[a,b,1], Daniel Hsu[c], Siyuan Ma[a], and Soumik Mandal[a]

Implicit regularization of GD/SGD

Assume linear model $y = Xw$ and consider ERM

SGD update

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

Here g_t is the gradient of the loss at the current prediction

Simple but important observation

If we initialize $w_0 = 0$, then

w_t always lies in span of data!

Exercise: verify above claim

Even though general weights are high-dimensional, SGD searches over space of at most dimension *n,* the number of data points.

Suppose we have nonnegative loss with $\frac{\partial \ell(z, y)}{\partial z} = 0$ iff $y{=}z$ (square-loss satisfies this)

Implicit regularization of GD/SGD

Thus, at optimality we have:

1. $Xw{=}y$, because total loss is zero ($\|Xw - y\|^2$)
2. $w = X^T v$, for some vector v, because w is in the span of data

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

Thus, when we run (S)GD we converge to a very specific solution. This special *w* turns out to be the *minimum Euclidean norm solution* to $Xw{=}y$!

Exercise: Prove that this soln. has minimum Euclidean norm

Suppose $\hat{w} = X^T\alpha + v, \quad v \perp x_i$

Then, $X\hat{w} = XX^T\alpha + Xv = XX^T\alpha$

Thus, $\hat{w} = X^T(XX^T)^{-1}y + v$

whereby, $\|\hat{w}\|^2 = \|X^T(XX^T)^{-1}y\|^2 + \|v\|^2$

Thanks!

Questions?