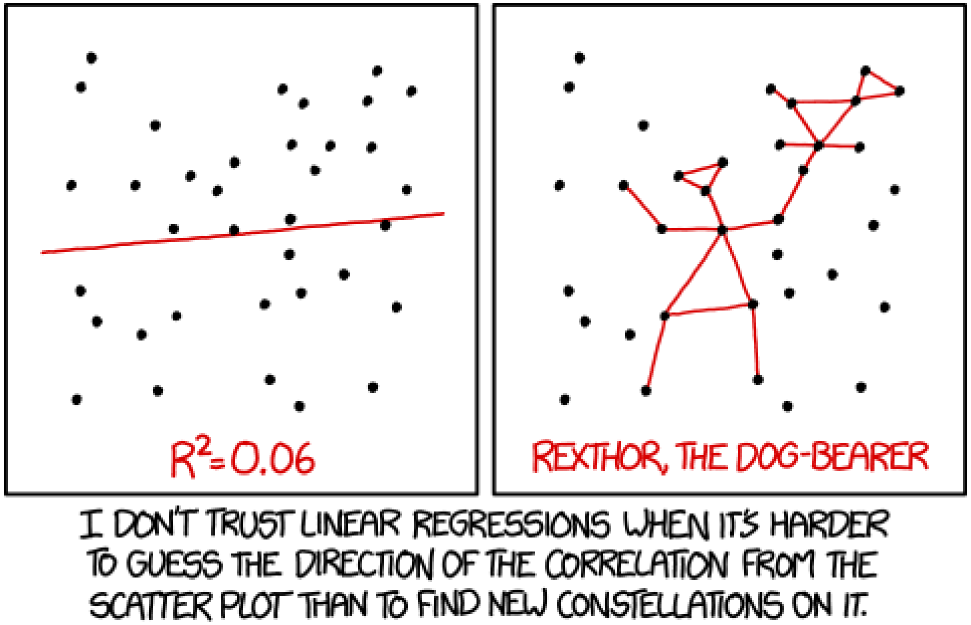
# LINEAR REGRESSION




Itay Evron & Ronen Nir

# Tutorial outline

- Linear regression
- The least squares problem
- Regularization
    - Ridge regression ($\ell 2$ regularization)
    - LASSO ($\ell 1$ regularization)

# Linear regression

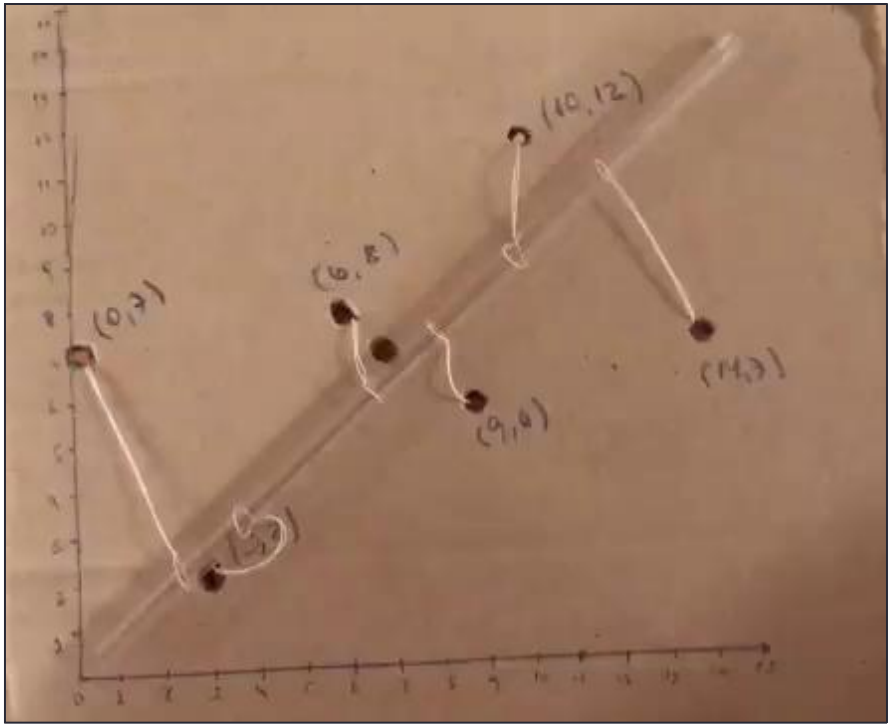
- We assume $\boldsymbol{y} \sim \boldsymbol{x}$ and ask how

  $\boldsymbol{x}$ explains $\boldsymbol{y}$

- Often, we assume a linear approximation $\boldsymbol{y} \approx (\boldsymbol{w}^*)^\top \boldsymbol{x} + \epsilon$ for an unknown "ground truth" $\boldsymbol{w}^* \in \mathbb{R}^d$

- Assuming a linear connection limits the search space

- We wish to find a good coefficient vector $\boldsymbol{w}$




Source: @jorge_pacheco

# Optimality criterion (loss)

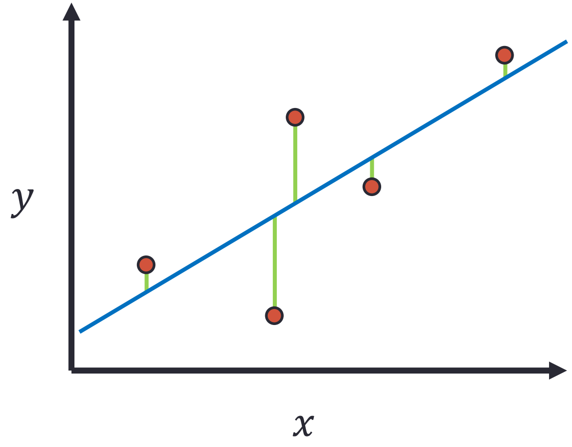
- For many reasons, we choose the squared error

$$\big(y_i - h(\boldsymbol{x}_i)\big)^2 = (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2$$

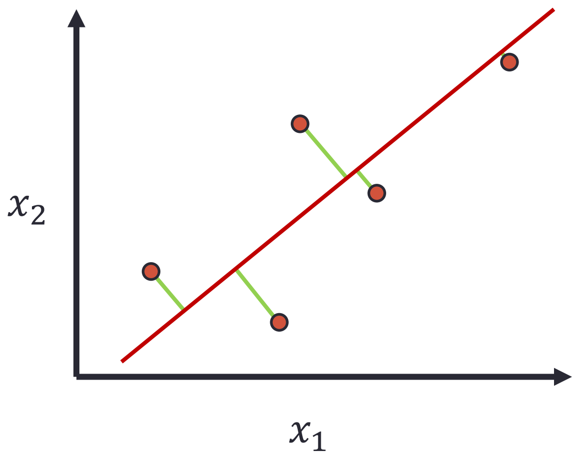
- The minimized loss:

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - h(\boldsymbol{x}_i)\big)^2$$

(Ordinary) Least Squares




Extra – another criterion:
Total Least Squares




Further reading: OLS vs. TLS

# The least squares problem

- Define the squared loss over the residuals:

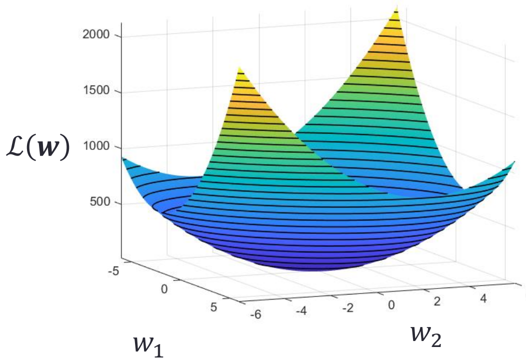
$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \underbrace{\boldsymbol{w}^\top \boldsymbol{x}_i}_{h(\boldsymbol{x}_i)})^2 = \tfrac{1}{m}\|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$$

where $\mathbf{X} = \begin{bmatrix} -\boldsymbol{x}_1^\top - \\ \vdots \\ -\boldsymbol{x}_m^\top - \end{bmatrix} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$.

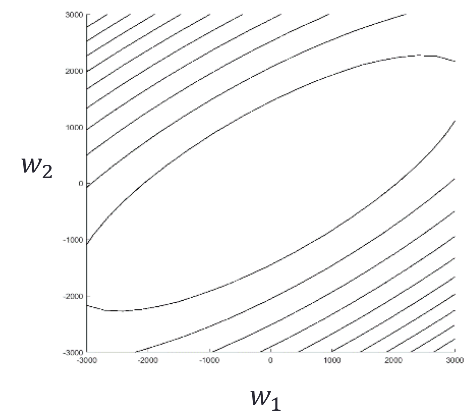
- The optimization problem: $\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \cancel{\tfrac{1}{m}} \|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$

- The gradient of the loss: $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}) = 2\mathbf{X}^\top(\mathbf{X}\boldsymbol{w} - \boldsymbol{y})$
- The Hessian of the loss: $\nabla_{\boldsymbol{w}}^2\mathcal{L}(\boldsymbol{w}) = 2\mathbf{X}^\top\mathbf{X} \succcurlyeq \mathbf{0}$
- $\Longrightarrow$ The objective loss is convex in $\boldsymbol{w}$!

Loss landscape
of a 2d LS problem




Loss level sets



# Solving least squares problems

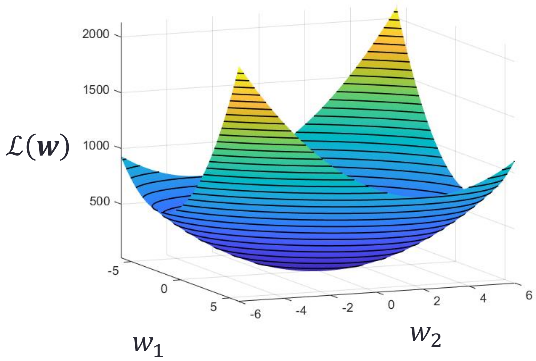
- Derive the normal equation:

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}) = 2\mathbf{X}^\top(\mathbf{X}\boldsymbol{w} - \boldsymbol{y}) = 0 \implies \mathbf{X}^\top\mathbf{X}\widehat{\boldsymbol{w}} = \mathbf{X}^\top\boldsymbol{y}$$

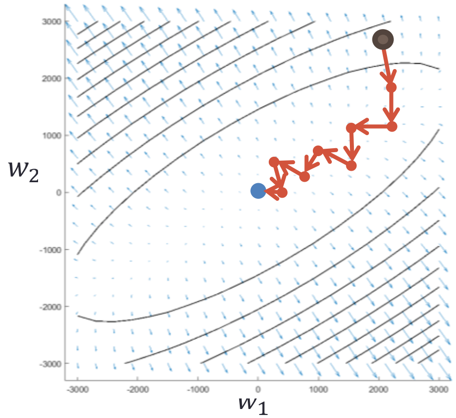
- Closed-form solution:
  - If $\mathbf{X}^\top\mathbf{X} \succ \mathbf{0}$, the unique solution is $\widehat{\boldsymbol{w}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}$
  - More generally $\widehat{\boldsymbol{w}} = (\mathbf{X}^\top\mathbf{X})^{+}\mathbf{X}^\top\boldsymbol{y} = \mathbf{V}\boldsymbol{\Sigma}^{+}\mathbf{U}^\top\boldsymbol{y}$

- Complexity:
  - Often, inversion is too expensive
  - Can use gradient methods $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \eta\mathbf{X}^\top(\mathbf{X}\boldsymbol{w} - \boldsymbol{y})$

Loss landscape
of a 2d LS problem

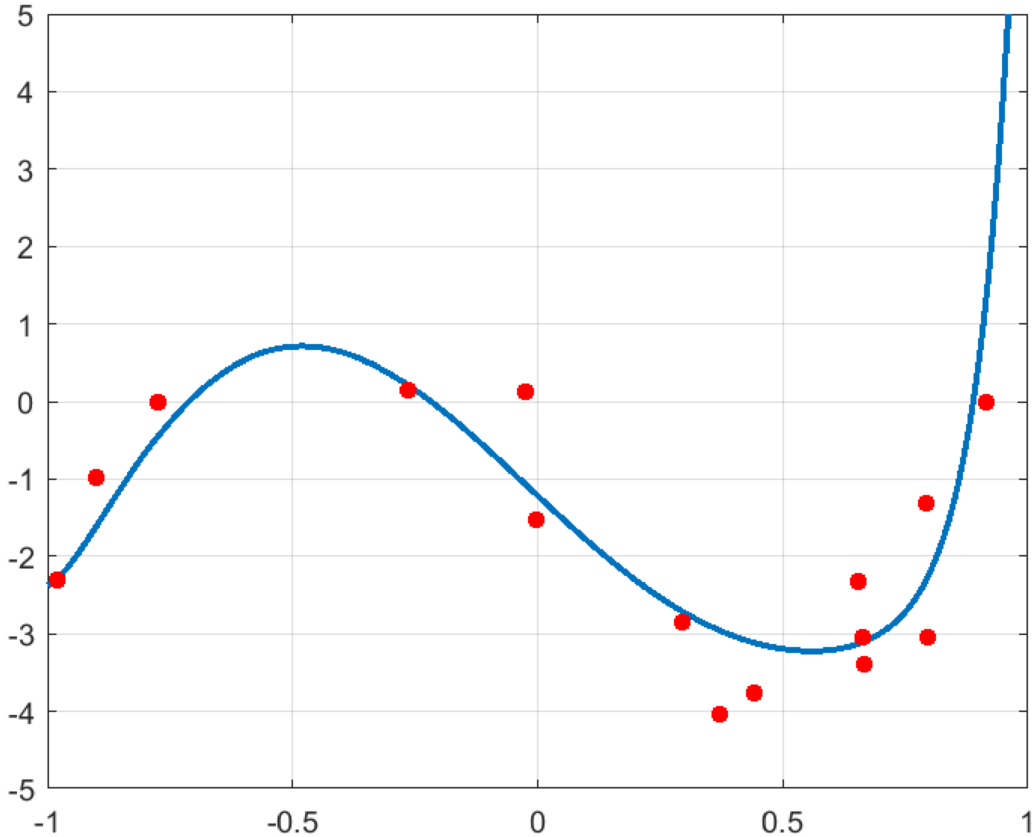



Loss level sets



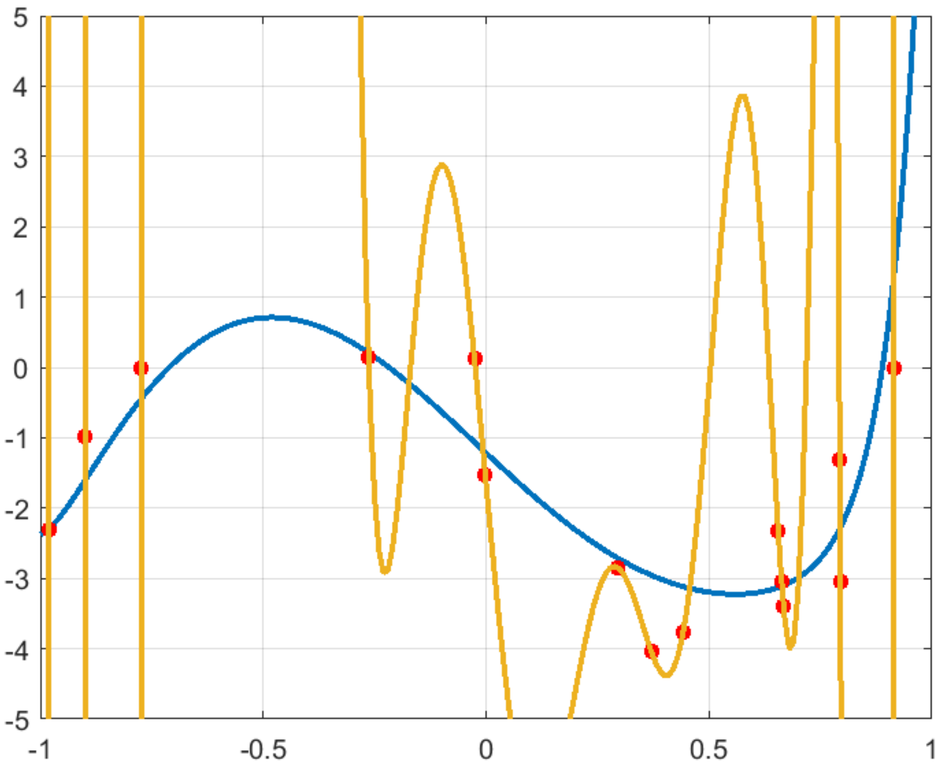
# Demo: Ridge regression for polynomial fitting

- Consider the illustrated polynomial function $f(\boldsymbol{x})$
- We sample points $(\boldsymbol{x}_i, y_i)$ where $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ for some i.i.d noise



# Demo: Ridge regression for polynomial fitting

- We will try to fit a polynomial function of degree 25




Vandermonde matrix
as a polynomial mapping:

$$X = \begin{bmatrix} — x_1^T — \\ \vdots \\ — x_m^T — \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

$$\Downarrow$$

$$X' = \begin{bmatrix} x_1 & x_1^2 & \cdots & x_1^{25} \\ \vdots & & & \vdots \\ x_m & & \cdots & x_m^{25} \end{bmatrix} = \begin{bmatrix} —\varphi(x_1)— \\ —\varphi(x_m)— \end{bmatrix}$$

$$\min \|X'w - y\|_2^2$$

- This is the solution that minimizes $\|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$
- Is this a good solution?

# Ridge regression ($\ell 2$ regularization)

- Regularize solutions with the $\ell 2$ norm:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left( \frac{1}{m} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_2^2 \right) = \underset{\boldsymbol{w}}{\operatorname{argmin}} \underbrace{\left( \frac{1}{m} \|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2 \right)}_{\mathcal{L}_\lambda(\boldsymbol{w})}$$

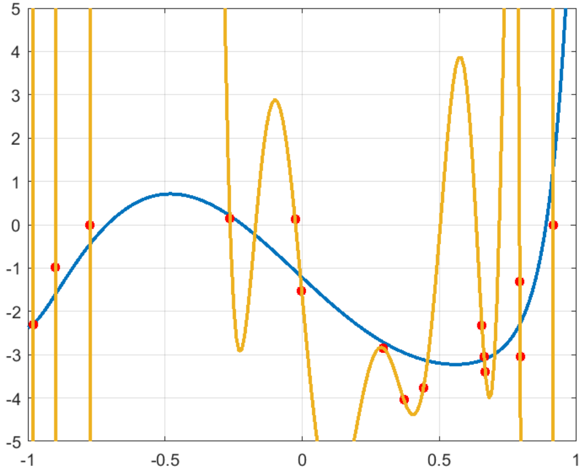
- Also called Tikhonov regularization or weight decay (esp. in deep learning).
- The updated gradient and normal equation:

$$\nabla_{\boldsymbol{w}} \mathcal{L}_\lambda(\boldsymbol{w}) = \frac{2}{m} \mathbf{X}^\top (\mathbf{X}\boldsymbol{w} - \boldsymbol{y}) + 2\lambda \boldsymbol{w} \Longrightarrow \underbrace{\left( \mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I}_{d \times d} \right)}_{\succ \mathbf{0}} \hat{\boldsymbol{w}} = \mathbf{X}^\top \boldsymbol{y}$$

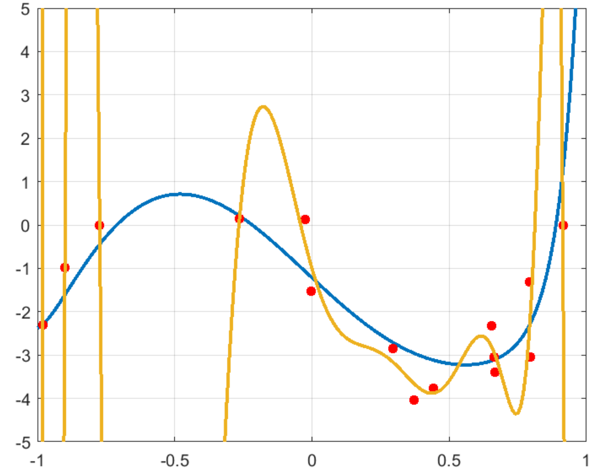
- The shrinkage effect
  - We should understand what happens in the limits $\lambda \to 0$ and $\lambda \to \infty$
- Optimization
  - We can compute the closed form solution $\hat{\boldsymbol{w}} = (\mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I}_{d \times d})^{-1} \mathbf{X}^\top \boldsymbol{y}$
  - Loss remains convex <u>and</u> differentiable, so we can still run GD

# Demo: Ridge regression for polynomial fitting

$\lambda = 0$

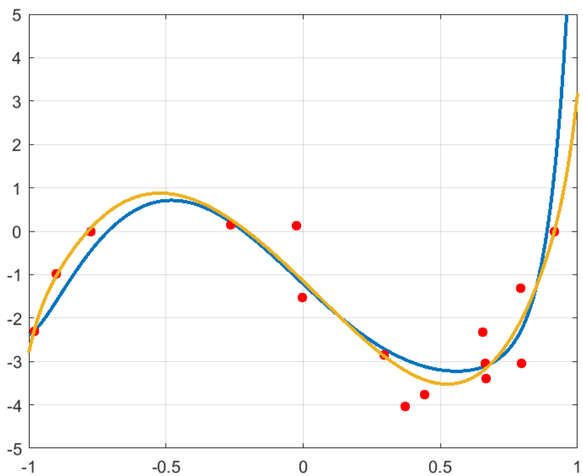



$\lambda = 10^{-8}$

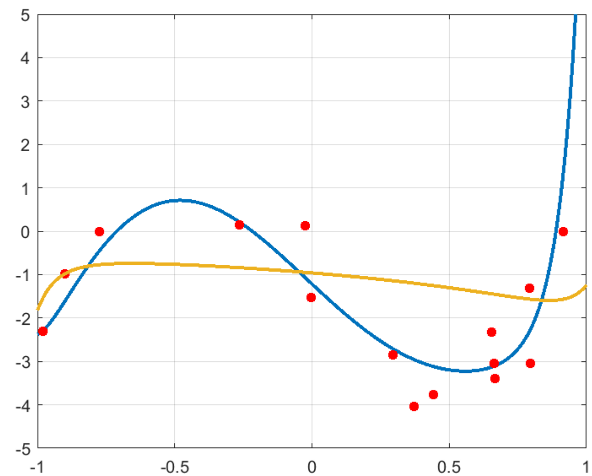



Regularization mitigates overfitting and helps generalization!

$\lambda = 10^{-2}$

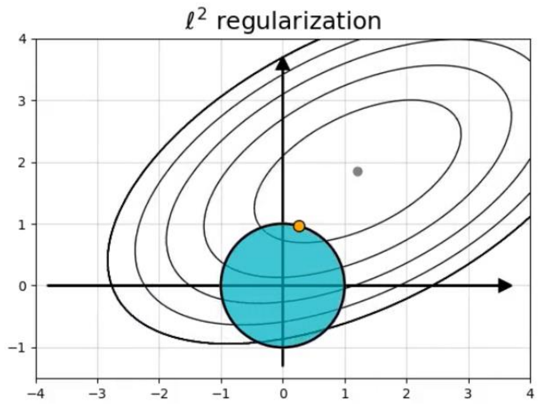



$\lambda = 10$



# Equivalence to constrained problems

- Theorem: the regularized problems are equivalent to unregularized problems with norm constraints.

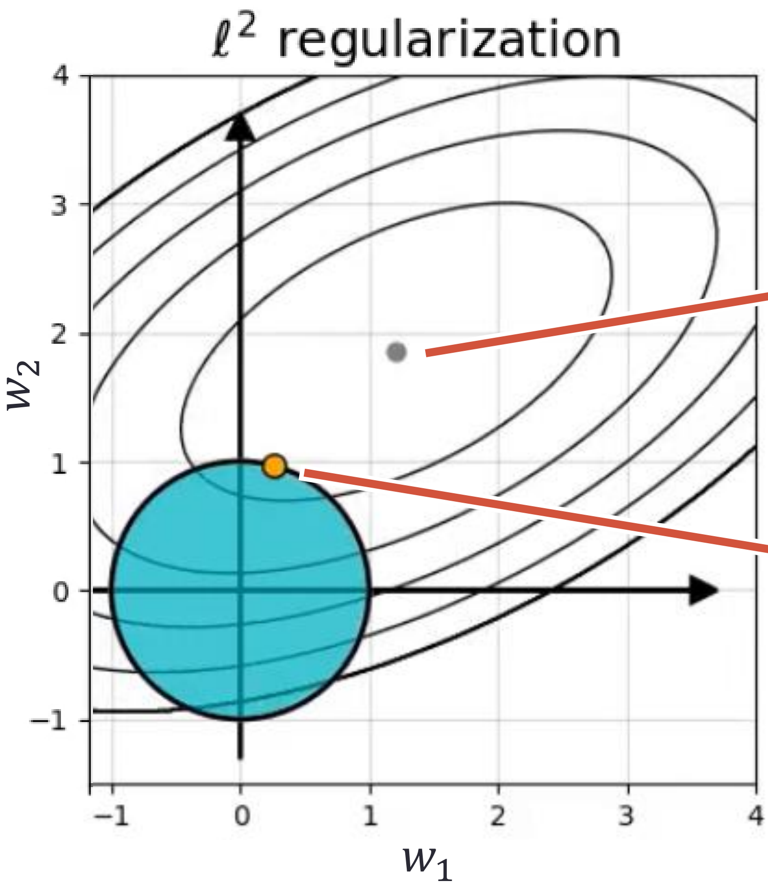



$$\boldsymbol{w}^{\text{Ridge}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\left(\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda\|\boldsymbol{w}\|_2^2\right)$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmin}}\,\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2\,, s.t.\ \|\boldsymbol{w}\|_2^2 \le c$$

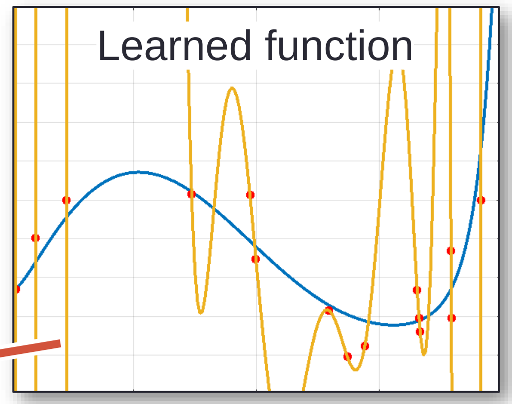
where $\lambda, c$ are related.

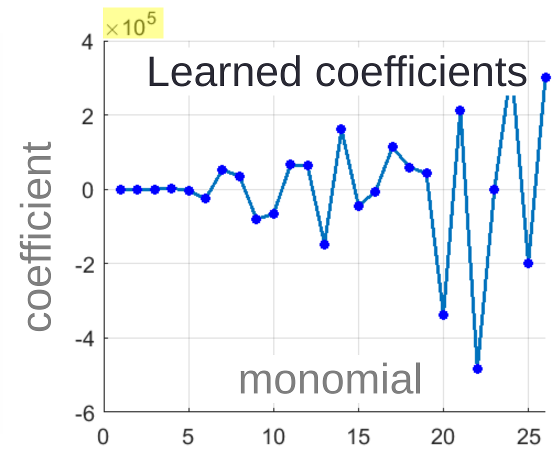
# Understanding the solution space



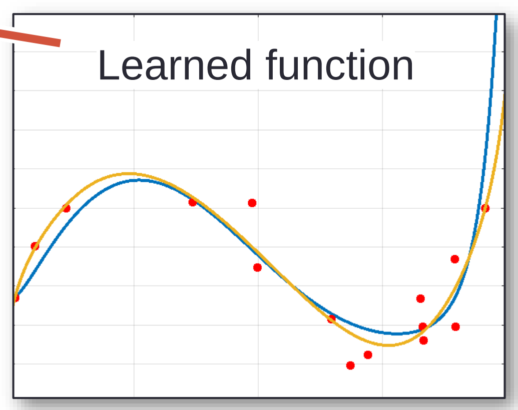

(illustration in 2d)

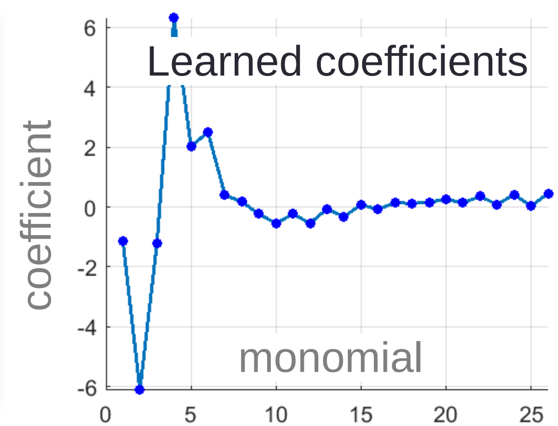
Minimal train loss solution (unregularized solution)




Bounded norm solution (regularized solution)



# LASSO ($\ell 1$ regularization)

- Least Absolute Shrinkage and Selection Operator
- Regularize solutions with the $\ell 1$ norm:

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \left( \frac{1}{m} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1 \right)$$

- Often induces sparse solutions (few nonzero entries)
- No closed-form solution!
- Optimization
  - Loss remains convex but no longer differentiable.
  - Could run subgradient descent, but more suitable algorithms exist.
    - Think: how will the subgradient method perform on $f(x) = |x|$ ?
  - Further reading: Why not vanilla GD?, Proximal gradient methods for learning, FISTA

# Equivalence to constrained problems

- Theorem: the regularized problems are equivalent to unregularized problems with norm constraints.

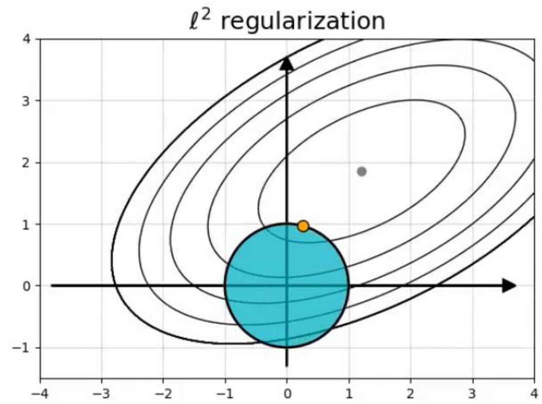
$$\boldsymbol{w}^{\text{Ridge}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\left(\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda\|\boldsymbol{w}\|_2^2\right)$$

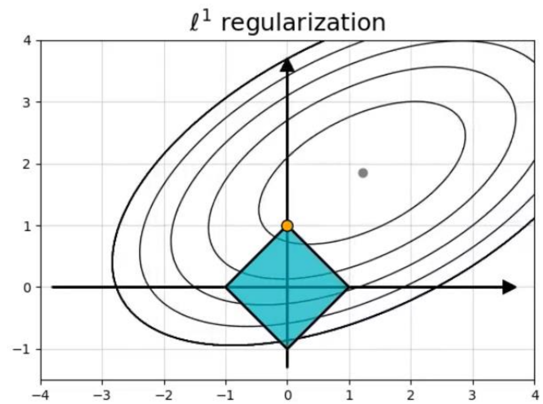
$$= \underset{\boldsymbol{w}}{\operatorname{argmin}}\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2, s.t.\ \|\boldsymbol{w}\|_2^2 \le c$$

$$\boldsymbol{w}^{\text{LASSO}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\left(\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda\|\boldsymbol{w}\|_1\right)$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmin}}\frac{1}{m}\Sigma_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2, s.t.\ \|\boldsymbol{w}\|_1 \le c$$

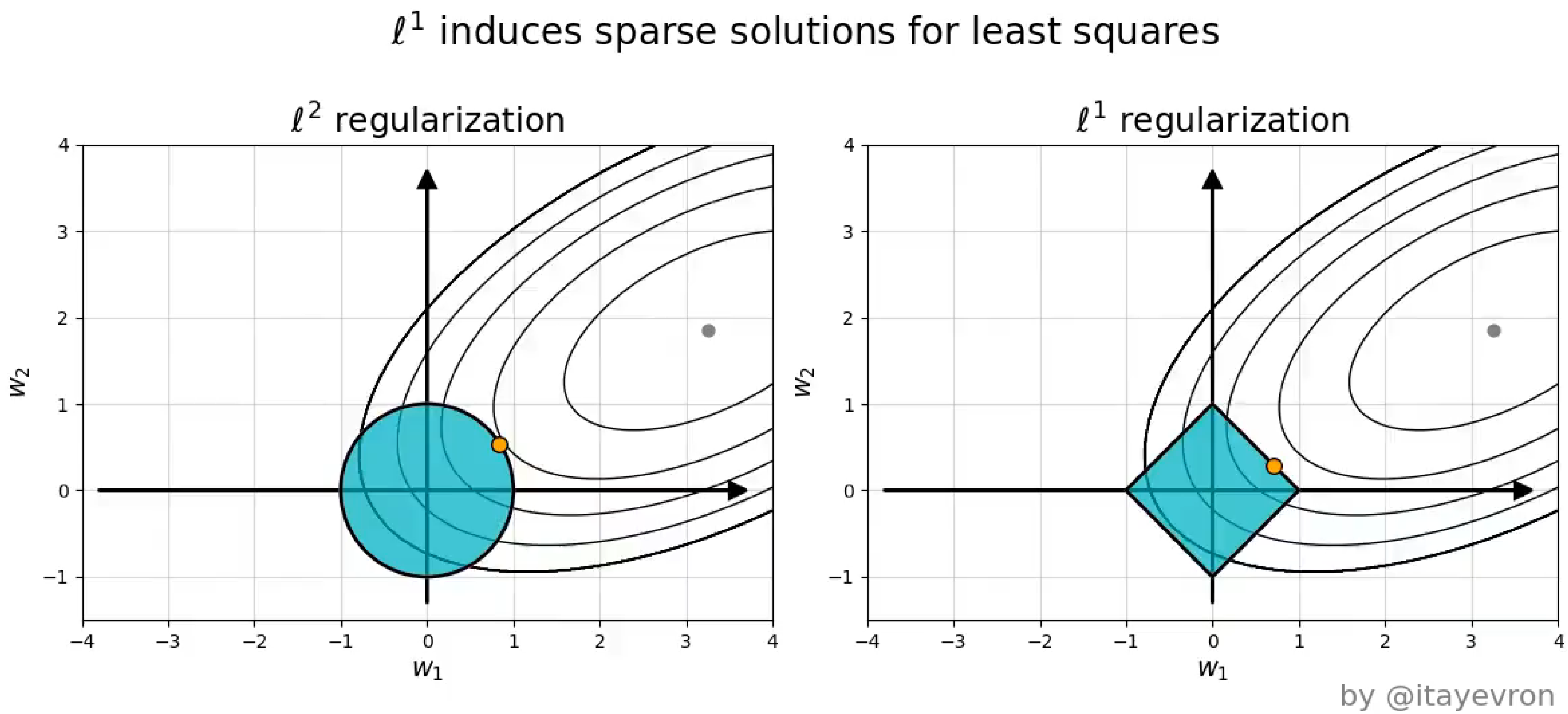
where $\lambda, c$ are related.




- Illustrate why LASSO induces sparser models.

# LASSO induces sparse models



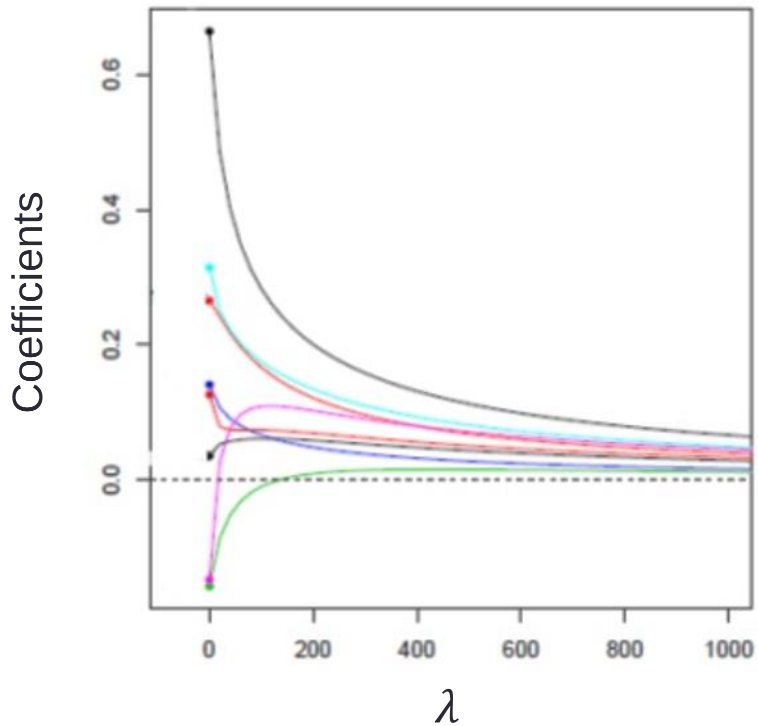

The level sets belong to an unregularized least squares problem.
The orange points have the lowest LS error on each unit “circle”.
Animation can be found on GitHub.

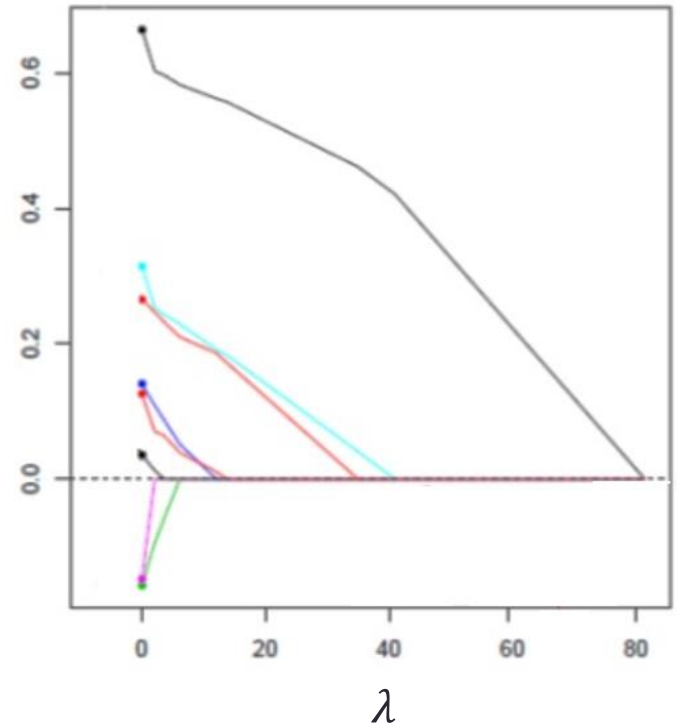
Notice: in both cases we don’t get the solution with the minimal (unregularized) training MSE!

# Different regularization behaviors

$\ell^2$ regularization causes weight decay

$\ell^1$ regularization causes "variable selection"




Extra: Are larger <u>unregularized</u> coefficients necessarily more "important"?

Answer: Not necessarily! See Q3 in Exam A of Winter 2020-21

Slide source: Derek Kane

# Summary

- Linear regression tries to linearly "explain" labels $y$ using feature vectors $\boldsymbol{x}$.
- Often formulated by least squares.
- Regularization can help prevent overfitting.
- Different regularizations induce different solutions.