CPSC 340:
Machine Learning and Data Mining

L1 Regularization

Bonus Slides

Hyper-Parameter Optimization

- Other common hyper-parameter optimization methods:
  - Exhaustive search with pruning:
    - If it “looks” like test error is getting worse as you decrease λ, stop decreasing it.
  - Coordinate search:
    - Optimize one hyper-parameter at a time, keeping the others fixed.
    - Repeatedly go through the hyper-parameters
  - Stochastic local search:
    - Generic global optimization methods (simulated annealing, genetic algorithms, etc.).
  - Bayesian optimization (Mike’s PhD research topic):
    - Use RBF regression to build model of how hyper-parameters affect validation error.
    - Try the best guess based on the model.

L1-Regularization Applications

- Used to give super-resolution in imaging black holes.
  - Sparsity arises in a particular basis.

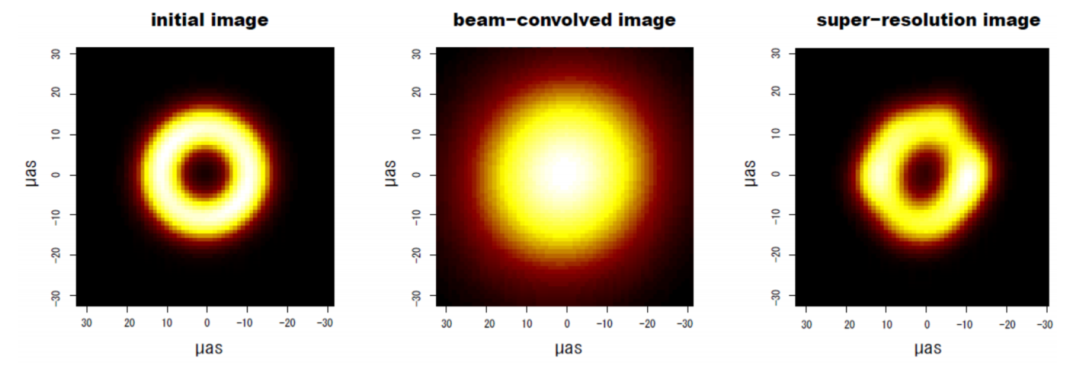


Figure 2. Simulated images of M87. From left to right, the initial model, the image with 0-filling, and the image with LASSO. Improvement of resolution in the LASSO image is significant.

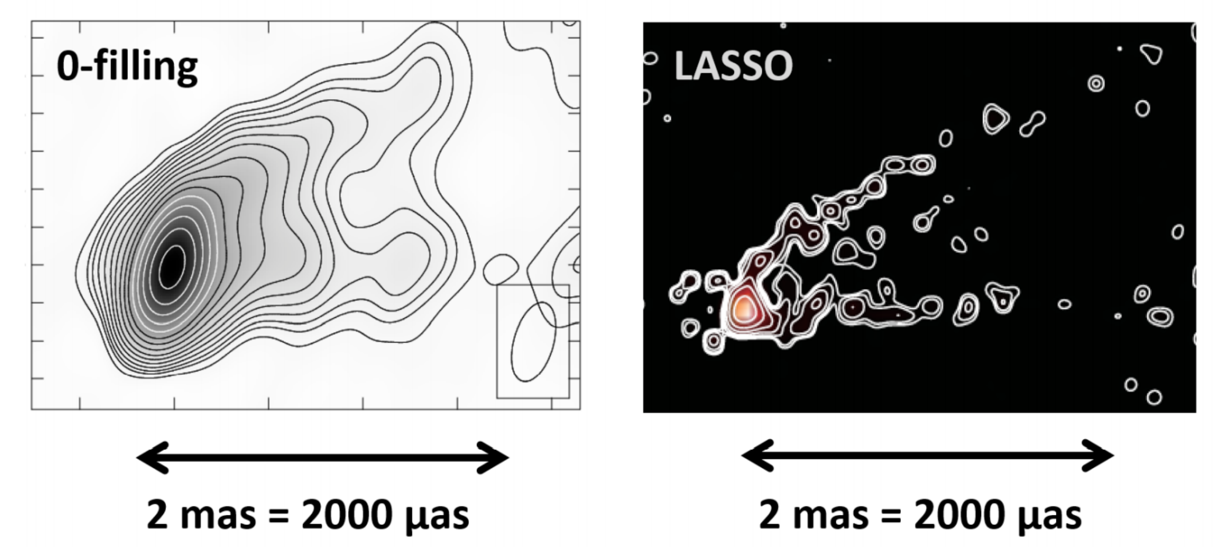


Figure 3. Standard and LASSO images of M87 observed with VLBA at a wavelength of 7 mm. In the two plots, exactly the same data are used. The angular resolution is better in the LASSO image, and the detailed structure of the M87 jet can be traced in more detail.

- Another application:
  - Use L1-regularization with Gaussian RBFs to reduce prediction time.

https://iopscience.iop.org/article/10.1088/1742-6596/699/1/012006/pdf

Regularizers and Sparsity

- L1-regularization gives sparsity but L2-regularization doesn't.
  - But don't they both shrink variables to zero?
- Consider problem where 3 vectors can get minimum training error:

$$w^1 = \begin{bmatrix} 100 \\ 0.02 \end{bmatrix} \qquad w^2 = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \qquad w^3 = \begin{bmatrix} 99.99 \\ 0.02 \end{bmatrix}$$

- Without regularization, we could choose any of these 3.
  - They all have same error, so regularization will "break tie".
- With L0-regularization, we would choose w^2:

$$||w^1||_0 = 2 \qquad ||w^2||_0 = 1 \qquad ||w^3||_0 = 2$$

Regularizers and Sparsity

- L1-regularization gives sparsity but L2-regularization doesn't.
  - But don't they both shrink variables to zero?
- Consider problem where 3 vectors can get minimum training error:

$$w^1 = \begin{bmatrix} 100 \\ 0.02 \end{bmatrix} \qquad w^2 = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \qquad w^3 = \begin{bmatrix} 99.99 \\ 0.02 \end{bmatrix}$$

- With L2-regularization, we would choose w^3:

$$\|w^1\|^2 = 100^2 + 0.02^2 = 10000.0004 \qquad \|w^2\|^2 = 100^2 + 0^2 = 10000 \qquad \|w^3\|^2 = 99.99^2 + 0.02^2 = 9998.0005$$

- L2-regularization focuses on decreasing largest (makes w_j similar).

Regularizers and Sparsity

- L1-regularization gives sparsity but L2-regularization doesn't.
  - But don't they both shrink variables to zero?
- Consider problem where 3 vectors can get minimum training error:

$$w^1 = \begin{bmatrix} 100 \\ 0.02 \end{bmatrix} \qquad w^2 = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \qquad w^3 = \begin{bmatrix} 99.99 \\ 0.02 \end{bmatrix}$$

- With L1-regularization, we would choose w^2:

$$\|w^1\|_1 = 100 + 0.02 \qquad \|w^2\|_1 = 100 + 0 \qquad \|w^3\|_1 = 99.99 + 0.02$$
$$= 100.02 \qquad\qquad = 100 \qquad\qquad = 100.01$$

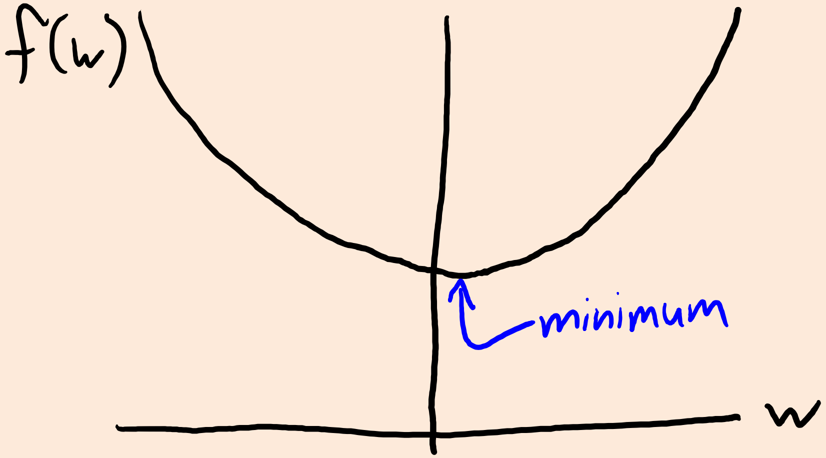
- L1-regularization focuses on decreasing all w_j until they are 0.

Sparsity and Least Squares

- Consider 1D least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2$$

- This is a convex 1D quadratic function of ‘w’ (i.e., a parabola):



- This variable does not look relevant (minimum is close to 0).
  - But for finite ‘n’ the minimum is unlikely to be exactly zero.

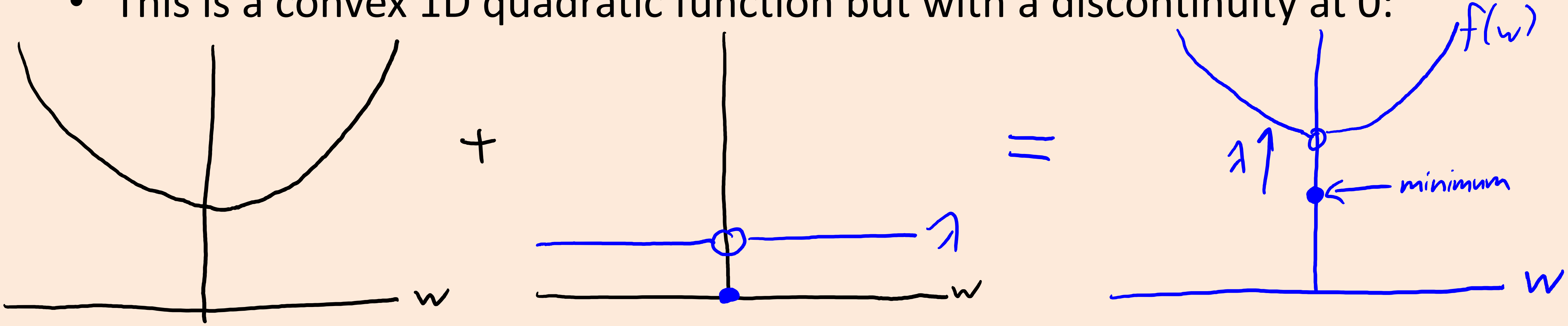
$f'(0) = 0$ only happens if $\sum_{i=1}^{n} y_i x_i = 0$ (bonus)

Sparsity and L0-Regularization

- Consider 1D L0-regularized least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 + \underbrace{\lambda \|w\|_0}_{\begin{cases}\lambda & \text{if } w \neq 0 \\ 0 & \text{if } w = 0\end{cases}}$$

- This is a convex 1D quadratic function but with a discontinuity at 0:



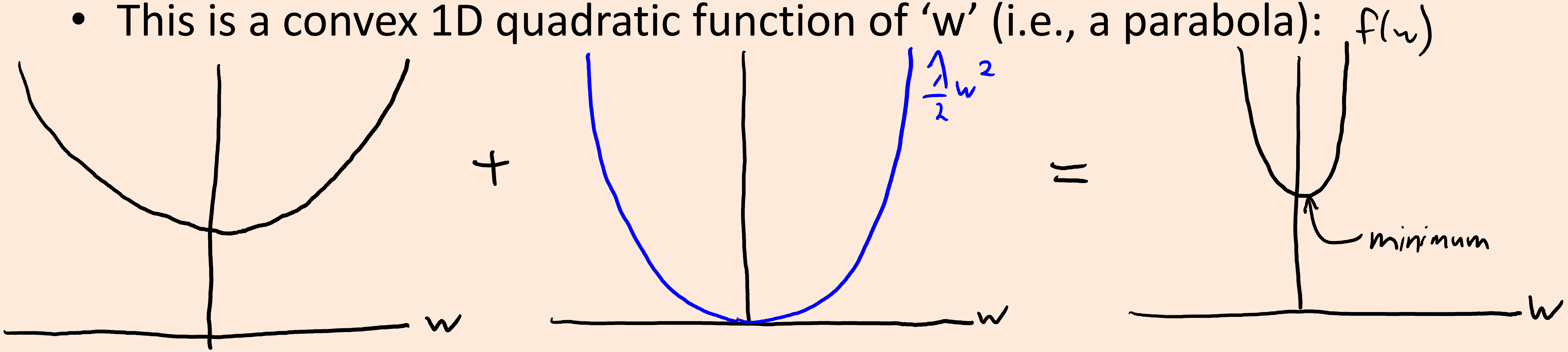
- L0-regularized minimum is often exactly at the ‘discontinuity’ at 0:
  - Sets the feature to exactly 0 (does feature selection), but is non-convex.

Sparsity and L2-Regularization

- Consider 1D L2-regularized least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 + \frac{\lambda}{2}w^2$$

- This is a convex 1D quadratic function of ‘w’ (i.e., a parabola):



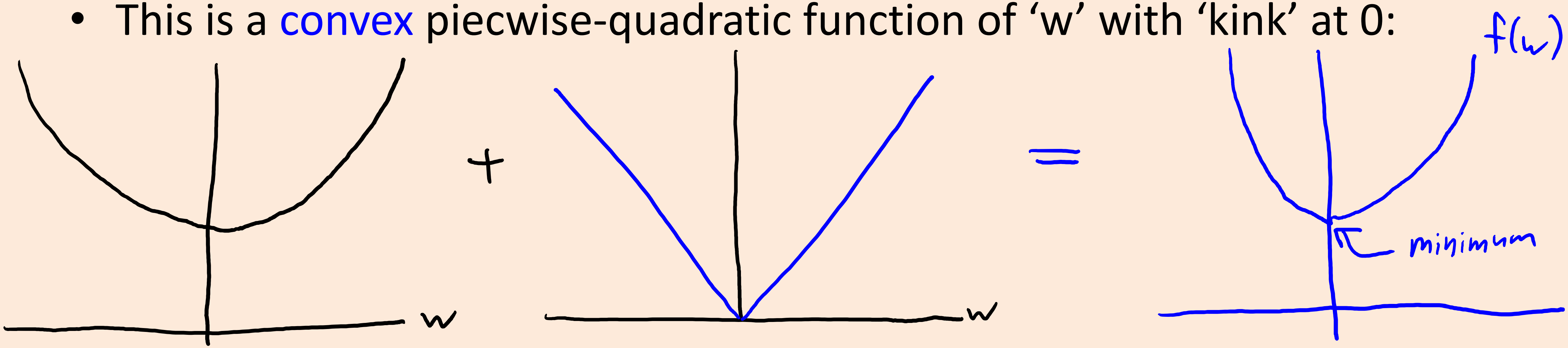
- L2-regularization moves it closer to zero, but not all the way to zero.
  - It doesn’t do feature selection (“penalty goes to 0 as slope goes to 0”). → $f'(0) = 0$ only if $\sum_{i=1}^{n} y_i x_i = 0$

Sparsity and L1-Regularization

- Consider 1D L1-regularized least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 + \lambda|w|$$

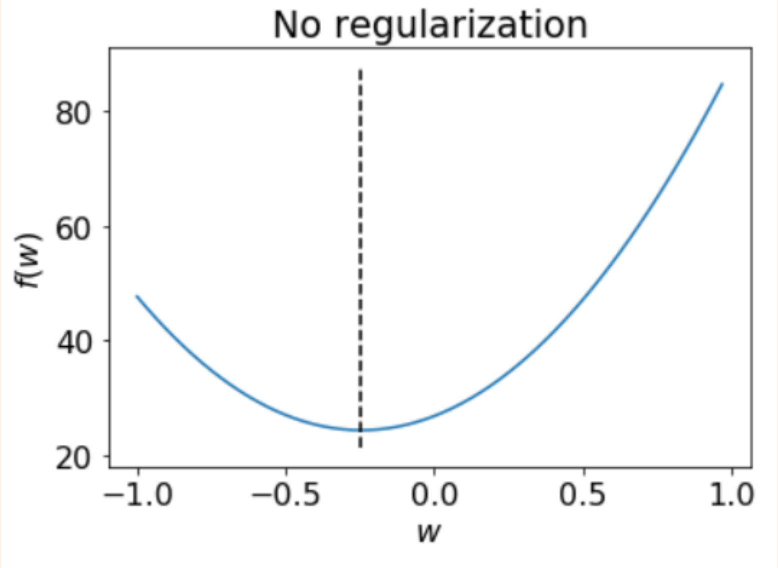
- This is a convex piecwise-quadratic function of ‘w’ with ‘kink’ at 0:

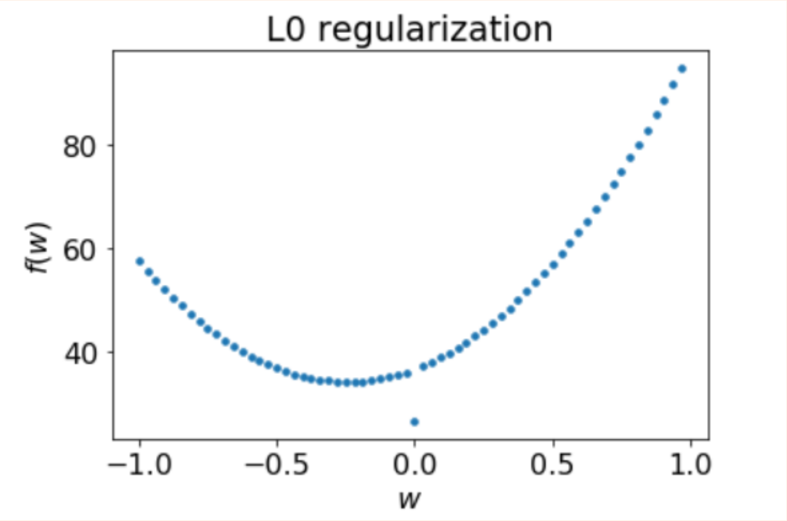


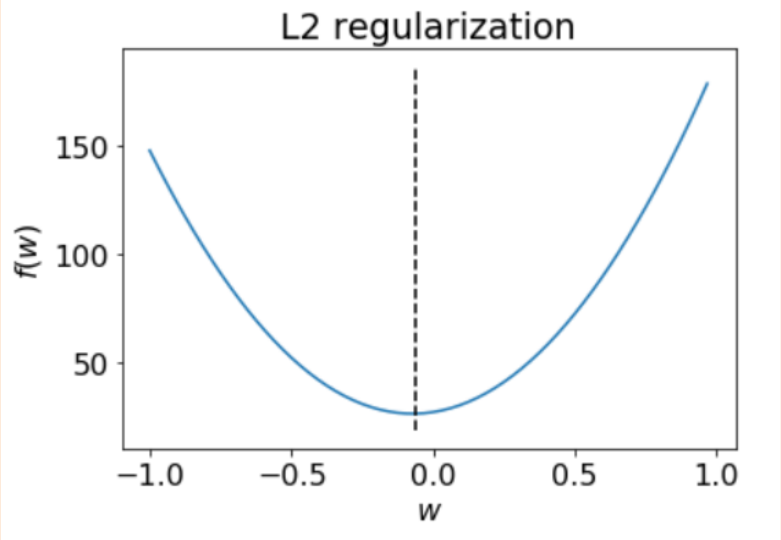
- L1-regularization tends to set variables to exactly 0 (feature selection).
  - Penalty on slope is λ even if you are close to zero.
  - Big λ selects few features, small λ allows many features.

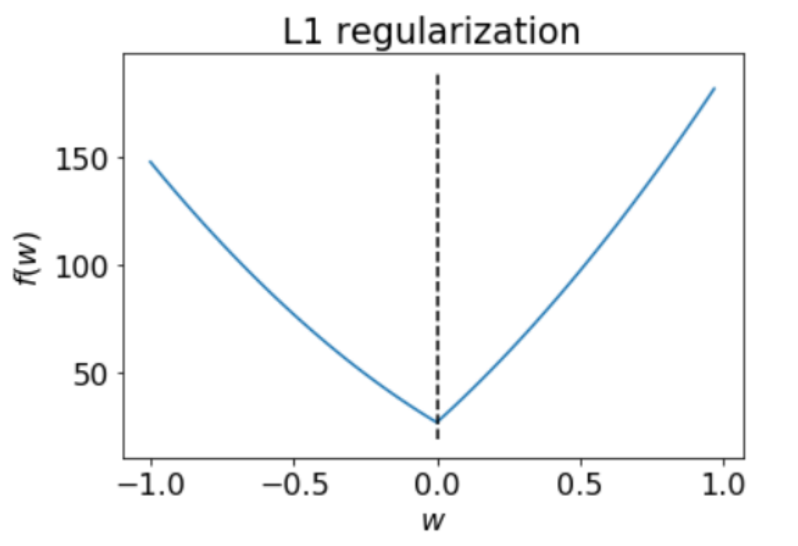
→ Happens when $\left|\sum_{i=1}^{n} x_i y_i\right| \leq \lambda$ (bonus)

Sparsity and Regularization (with d=1)









Why doesn't L2-Regularization set variables to 0?

- Consider an L2-regularized least squares problem with 1 feature:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2 + \frac{\lambda}{2}w^2$$

- Let's solve for the optimal 'w':

$$f'(w) = \sum_{i=1}^{n} x_i(wx_i - y_i) + \lambda w$$

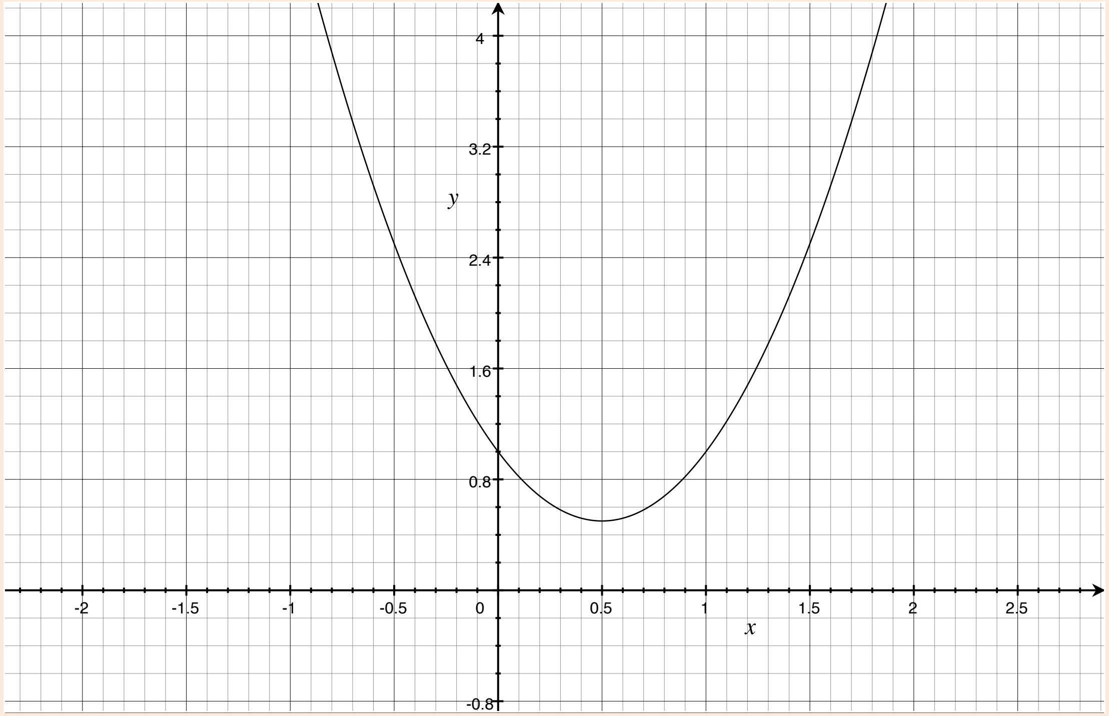
Set equal to 0: $\sum_{i=1}^{n} x_i^2 w - \sum_{i=1}^{n} x_i y_i + \lambda w = 0 \xrightarrow{\text{re-arrange}} w\Big(\underbrace{\sum_{i=1}^{n} x_i^2}_{\|x\|^2} + \lambda\Big) = \underbrace{\sum_{i=1}^{n} x_i y_i}_{y^Tx}$

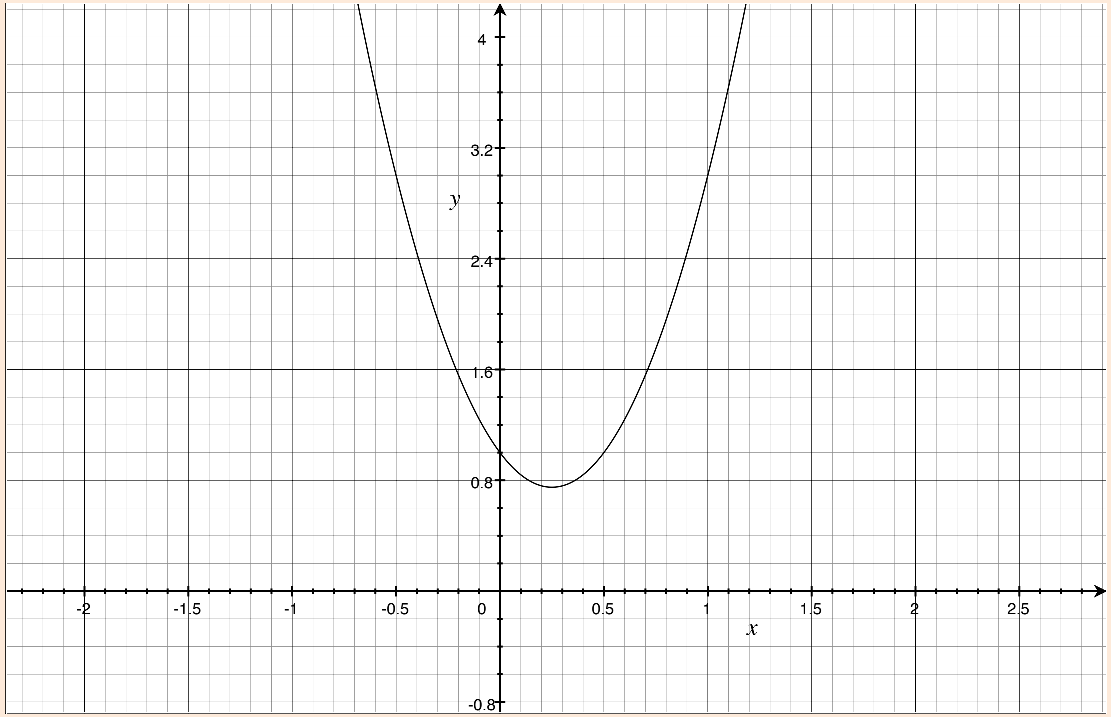
or $w = \frac{y^Tx}{\|x\|^2 + \lambda}$

- So as λ gets bigger, 'w' converges to 0.
- However, for all finite λ 'w' will be non-zero unless $y^Tx = 0$ exactly.
  - But it's very unlikely that y^Tx will be exactly zero.

Why doesn’t L2-Regularization set variables to 0?

- Small λ Big λ





- Solution further from zero

Solution closer to zero
(but not exactly 0)

Why does L1-Regularization set things to 0?

- Consider an L1-regularized least squares problem with 1 feature:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(wx_i - y_i)^2 + \lambda|w|$$

- If (w = 0), then "left" limit and "right" limit are given by:

$$f^-(0) = \sum_{i=1}^{n} x_i(0x_i - y_i) - \lambda$$
$$= \sum_{i=1}^{n} x_i y_i - \lambda$$

$$f^+(0) = \sum_{i=1}^{n} x_i(0x_i - y_i) + \lambda$$
$$= \sum_{i=1}^{n} x_i y_i + \lambda$$

- So which direction should "gradient descent" go in?

$$-f^-(0) = -y^Tx + \lambda$$
$$-f^+(0) = -y^Tx - \lambda$$

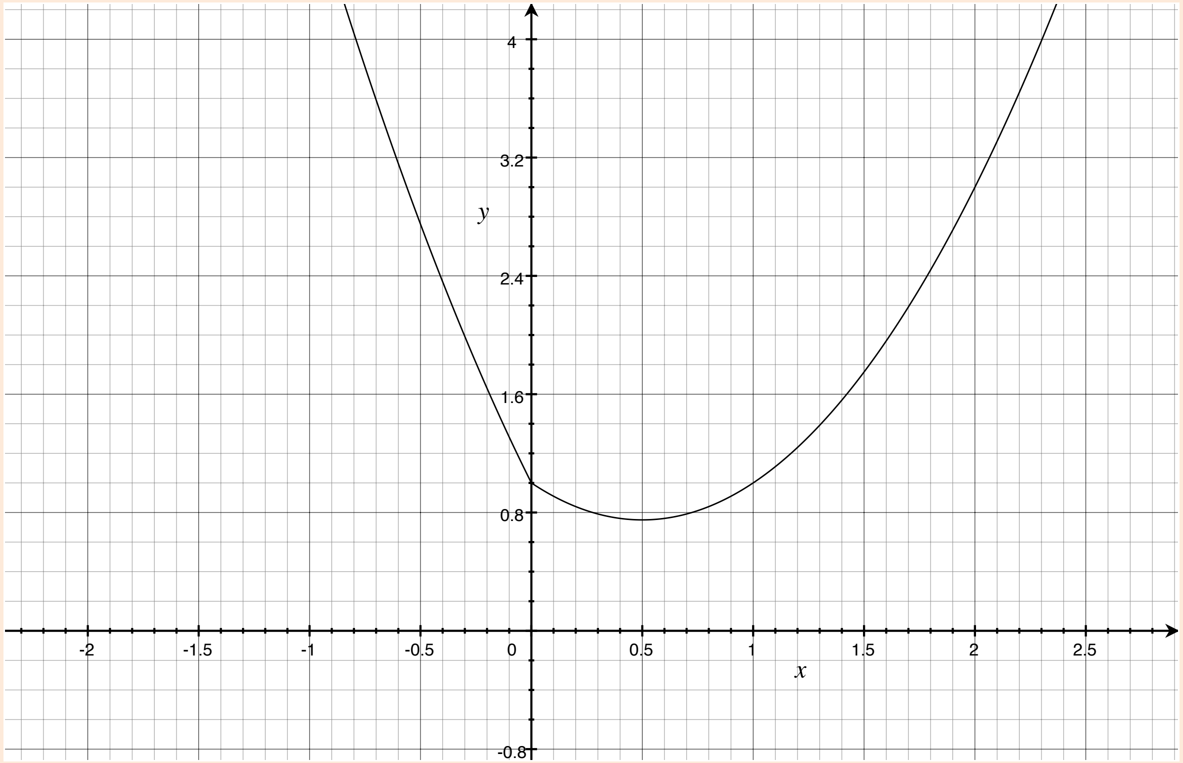
If these are positive $(-y^Tx > \lambda)$, we can improve by increasing 'w'.

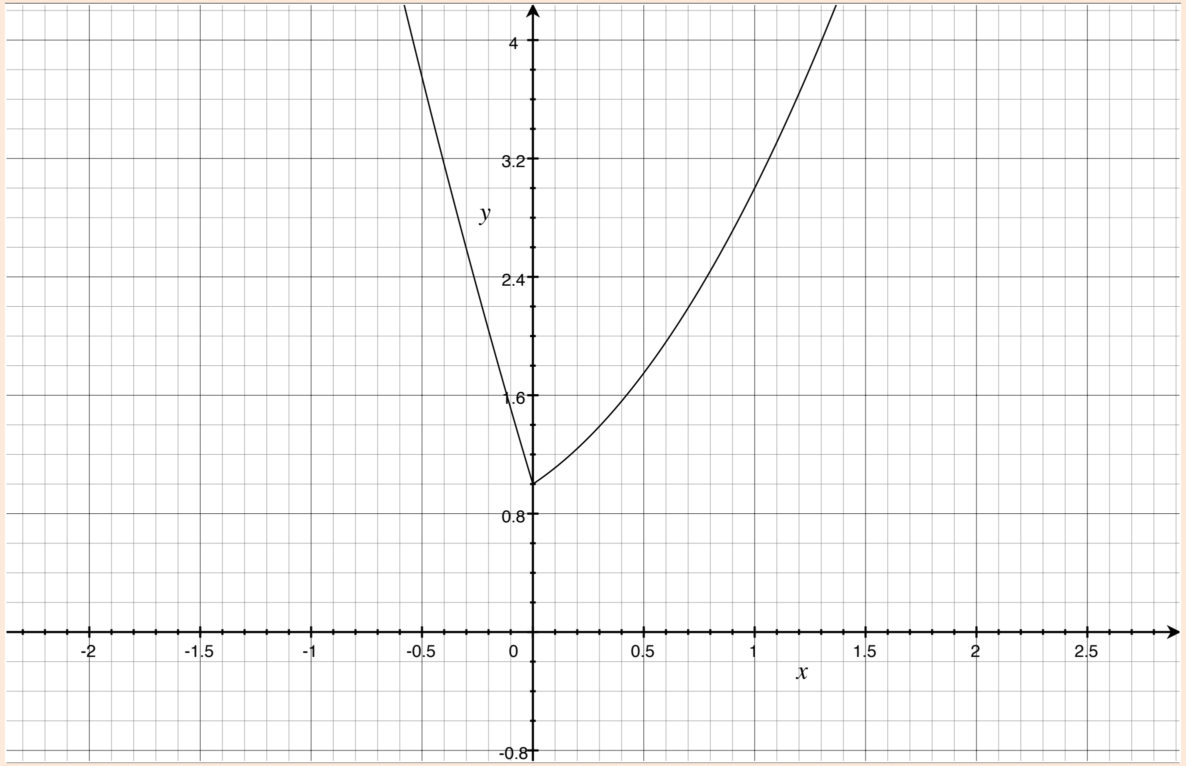
If these are negative $(y^Tx > \lambda)$, we can improve by decreasing 'w'.

But if left and right "gradient descent" directions point in opposite directions $(|y^Tx| \leq \lambda)$, minimum is 0.

Why does L1-Regularization set things to 0?

- Small λ Big λ





- Solution nonzero Solution exactly zero

(minimum of left parabola is past origin, but right parabola is not) (minimum of both parabola are past the origin)

L2-regularization vs. L1-regularization

- So with 1 feature:
  - L2-regularization only sets 'w' to 0 if $y^T x = 0$.
    - There is a only a single possible $y^T x$ value where the variable gets set to zero.
    - And λ has nothing to do with the sparsity.
  - L1-regularization sets 'w' to 0 if $|y^T x| \leq \lambda$.
    - There is a range of possible $y^T x$ values where the variable gets set to zero.
    - And increasing λ increases the sparsity since the range of $y^T x$ grows.
- Note that it's important that the function is non-differentiable:
  - Differentiable regularizers penalizing size would need $y^T x = 0$ for sparsity.

L1-Loss vs. Huber Loss

- The same reasoning tells us the difference between the L1 *loss* and the Huber loss. They are very similar in that they both grow linearly far away from 0. So both are both robust but…
  - With the L1 loss the model often passes exactly through some points.
  - With Huber the model doesn't necessarily pass through any points.

- Why? With L1-regularization we were causing the elements of 'w' to be exactly 0. Analogously, with the L1-loss we cause the elements of 'r' (the residual) to be exactly zero. But zero residual for an example means you pass through that example exactly.

Non-Uniqueness of L1-Regularized Solution

- How can L1-regularized least squares solution not be unique?
  - Isn't it convex?
- Convexity implies that minimum value of f(w) is unique (if exists), but there may be multiple 'w' values that achieve the minimum.

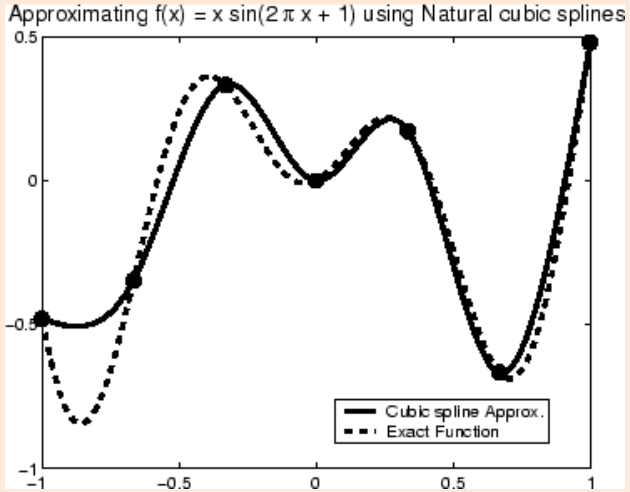
- Consider L1-regularized least squares with d=2, where feature 2 is a copy of a feature 1. For a solution (w_1,w_2) we have:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} = w_1 x_{i1} + w_2 x_{i1} = (w_1 + w_2) x_{i1}$$

- So we can get the same squared error with different w_1 and w_2 values that have the same sum. Further, if neither w_1 or w_2 changes sign, then $|w_1| + |w_2|$ will be the same so the new w_1 and w_2 will be a solution.

Splines in 1D

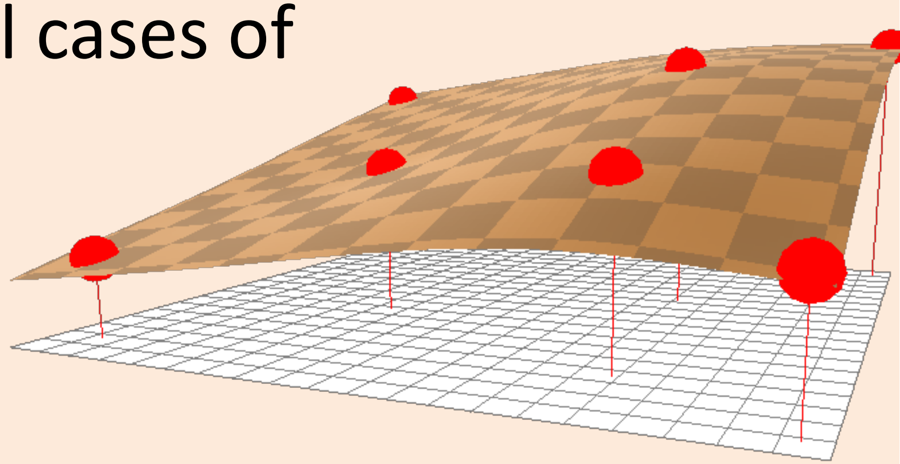
- For 1D interpolation, alternative to polynomials/RBFs are splines:
  - Use a polynomial in the region between each data point.
  - Constrain some derivatives of the polynomials to yield a unique solution.
- Most common example is cubic spline:
  - Use a degree-3 polynomial between each pair of points.
  - Enforce that f’(x) and f’’(x) of polynomials agree at all point.
  - “Natural” spline also enforces f’’(x) = 0 for smallest and largest x.
- Non-trivial fact: natural cubic splines are sum of:
  - Y-intercept.
  - Linear basis.
  - RBFs with $g(\varepsilon) = \varepsilon^3$.
    - Different than Gaussian RBF because it *increases with distance*.



http://www.physics.arizona.edu/~restrepo/475A/Notes/sourcea-/node35.html

Splines in Higher Dimensions

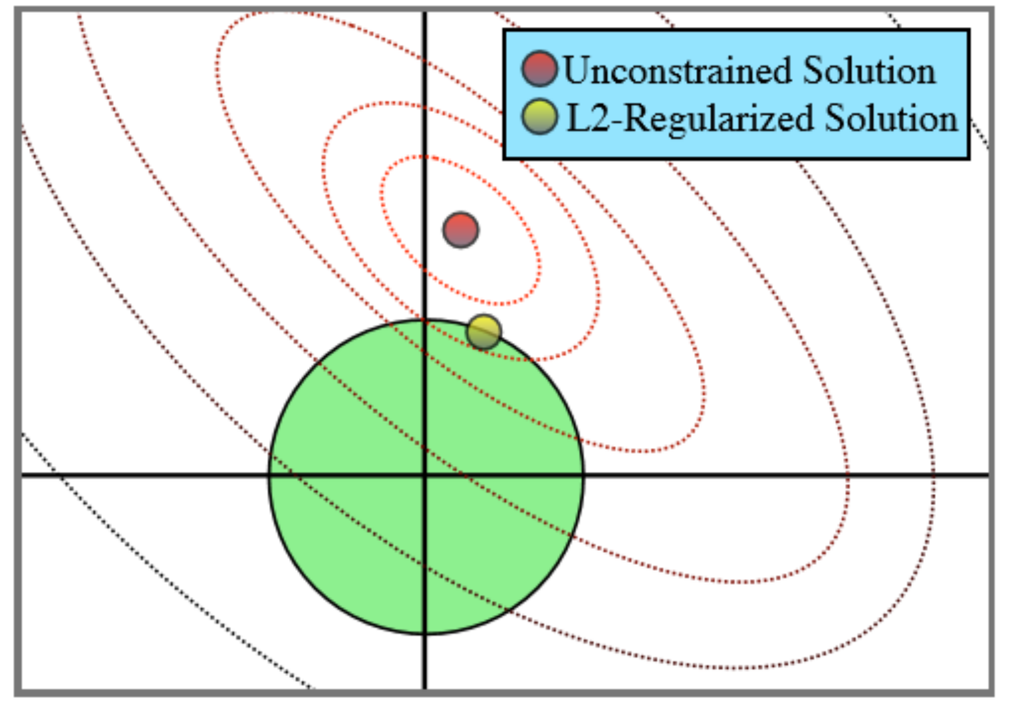
- Splines generalize to higher dimensions if data lies on a grid.
  - Many methods exist for grid-structured data (linear, cubic, splines, etc.).
  - For more general ("scattered") data, there isn't a natural generalization.
- Common 2D "scattered" data interpolation is thin-plate splines:
  - Based on curve made when bending sheets of metal.
  - Corresponds to RBFs with $g(\varepsilon) = \varepsilon^2 \log(\varepsilon)$.
- Natural splines and thin-plate splines: special cases of "polyharmonic" splines:
  - Less sensitive to parameters than Gaussian RBF.



http://step.polymtl.ca/~rv101/thinplates/

L2-Regularization vs. L1-Regularization

- L2-regularization conceptually restricts ‘w’ to a ball.

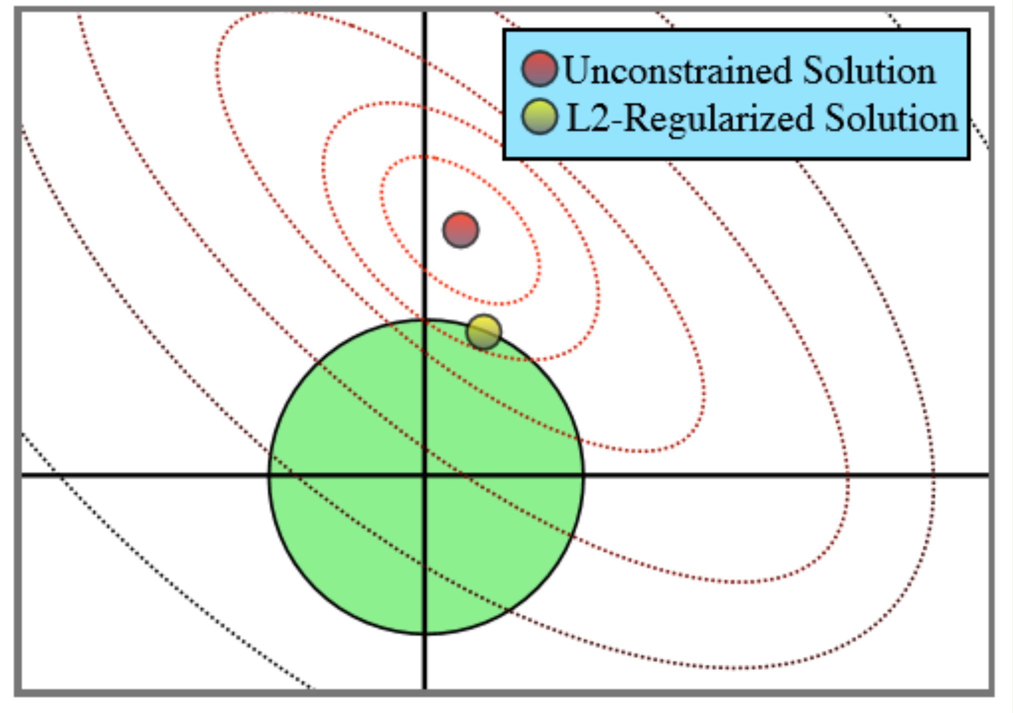


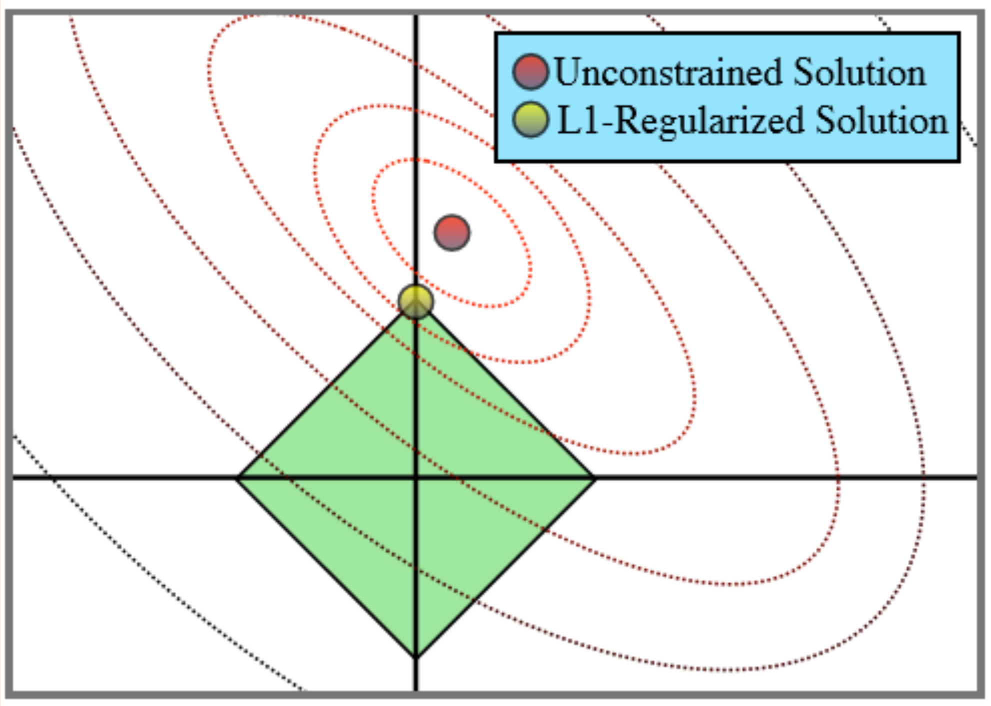
Minimizing $\frac{1}{2}||Xw-y||^2 + \frac{\lambda}{2}||w||^2$

is <u>equivalent</u> to minimizing $\frac{1}{2}||Xw-y||^2$ subject to the constraint that $||w|| \leq \tau$ for some value ‘τ’

L2-Regularization vs. L1-Regularization

- L2-regularization conceptually restricts ‘w’ to a ball.





- L1-regularization restricts to the L1 “ball”:
  - Solutions tend to be at corners where w_j are zero.

Related Infinite Series video