CPSC 340:
Machine Learning and Data Mining

Convex Functions

Summer 2021

Admin

- Assignment 3: due Friday
- Assignment 4: out Friday, due next Friday
- Midterm: coming up Tuesday
- Practice midterm out
  - See "Midterm Prep Megathread"

- This lecture is the last lecture for midterm material
- Second lecture will be a bonus lecture

In This Lecture

- Convex Functions (20 minutes)
- Least Squares with Outliers (25 minutes)

Last Time: Gradient Descent for Least Squares

- The least squares objective and gradient:

$$f(w) = \frac{1}{2}\|Xw - y\|^2 \qquad \nabla f(w) = X^T(Xw - y)$$

- Gradient descent iterations for least squares:

$$w^{t+1} = w^t - \alpha^t \underbrace{X^T(Xw^t - y)}_{\nabla f(w^t)}$$

- Cost of gradient descent iteration is O(__) (no need to form X^TX).

Bottleneck is computing $\nabla f(w^t) = X^T(Xw^t - y)$

Xw^t: $O(nd)$

$Xw^t - y$: $O(n)$

$X^T(Xw^t - y)$: $O(nd)$

Normal Equations vs. Gradient Descent

- Least squares via normal equations vs. gradient descent:
  - Normal equations cost $O(nd^2 + d^3)$.
  - Gradient descent costs O(___) to run for 't' iterations.
    - Each of the 't' iterations costs O(nd).
  - Normal equations only solve linear least squares problems.
    - Gradient descent solves many other problems.

Beyond Gradient Descent

- Gradient descent can be faster when 'd' is very large:
  - If solution is "good enough" for a 't' less than minimum(d,d^2/n).
  - Proportional to "condition number" of X^TX (no direct 'd' dependence).

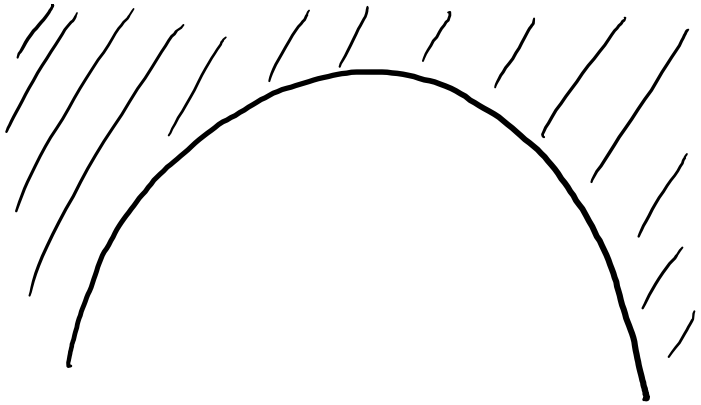
- There are many variations on gradient descent.
  - Methods employing a "line search" to choose the step-size.
  - "Conjugate" gradient and "accelerated" gradient methods.
  - Newton's method (which uses second derivatives).
  - Quasi-Newton and Hessian-free Newton methods.
  - Stochastic gradient (later in course).

- This course focuses on gradient descent and stochastic gradient:
  - They're simple and give reasonable solutions to most ML problems.
  - But the above can be faster for some applications.

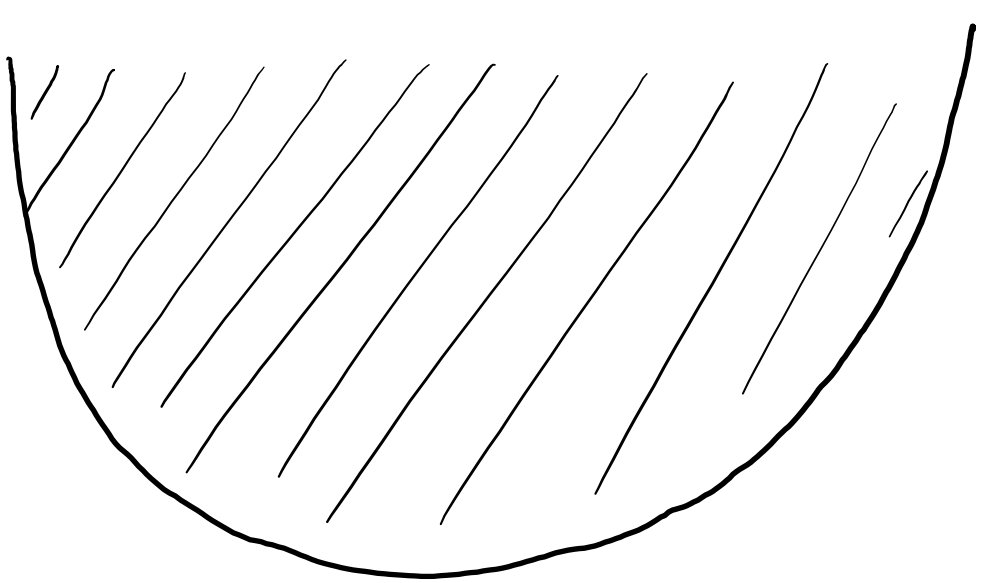
Coming Up Next

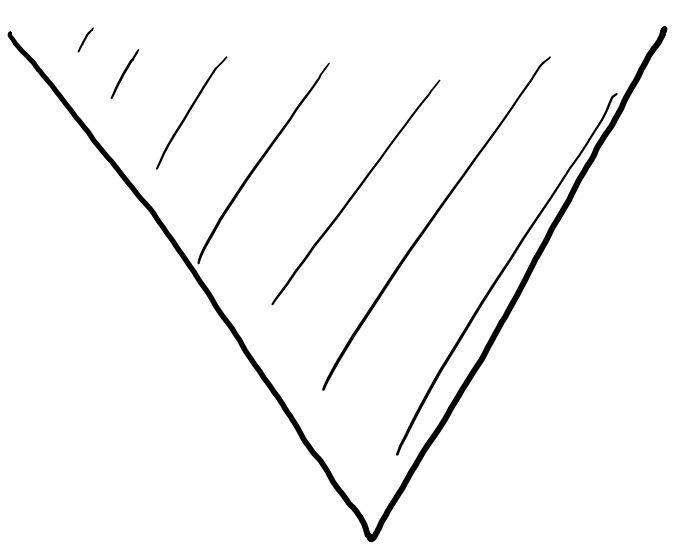
CONVEX FUNCTIONS

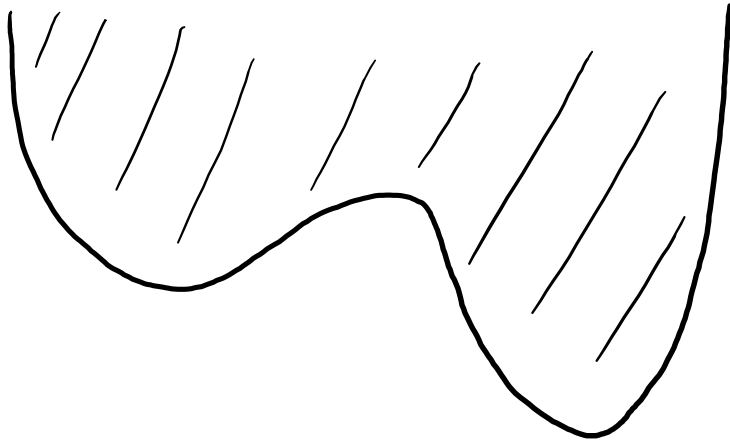
Convex Functions



- Is finding a 'w' with $\nabla f(w) = 0$ good enough?
  - Yes, for convex functions.







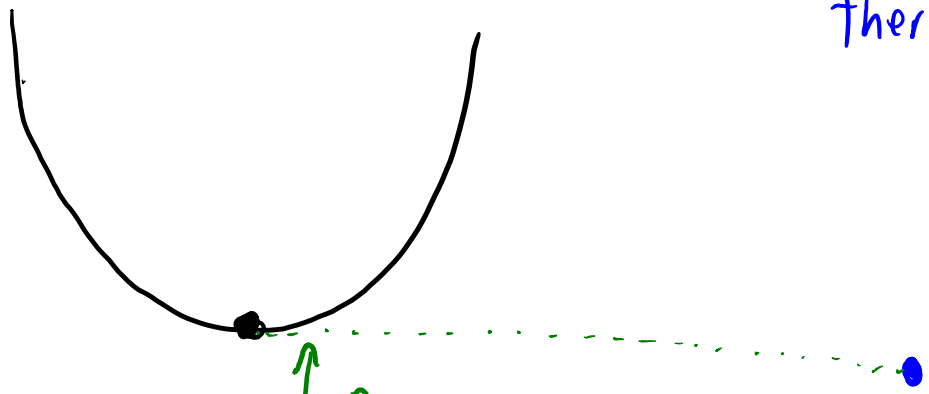
- A function is convex if the _____ is a convex set.
  - All values between any two points above function stay above function.

Convex Functions

- All 'w' with $\nabla f(w) = 0$ for convex functions are _____.

Proof by contradiction:

Consider a local minimum



If this is not global minimum, there must a smaller value.

But this contradicts that we are at a local minimum.

By convexity we can move along line to global minimum and decrease objective.

– Normal equations find a global minimum because least squares is convex.

Q: How do you know if a function is convex?

How do we know if a function is convex?

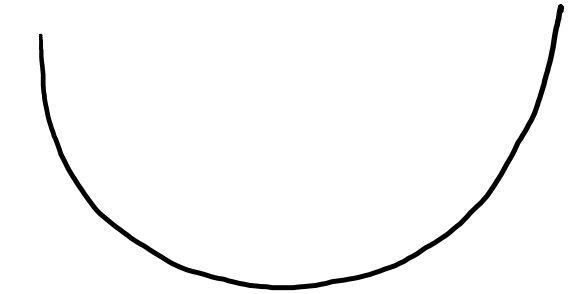
- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.

Consider $f(w) = \frac{1}{2} a w^2$ for $a > 0$

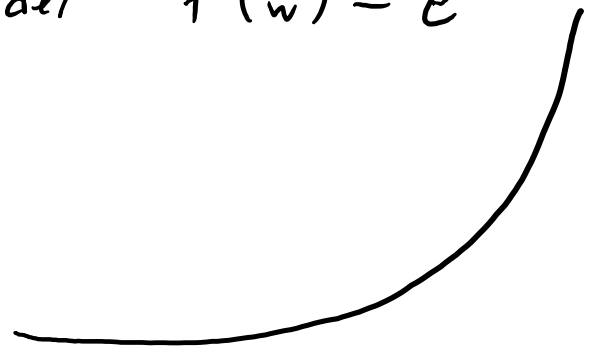
We have $f'(w) = aw$

and $f''(w) = a > 0$

By assumption



Consider $f(w) = e^w$



We have $f'(w) = e^w$

and $f''(w) = e^w > 0$

↳ By definition of exponential function.

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff $f''(w) \geq 0$ for all 'w'.
  - A convex function multiplied by non-negative constant is convex.

We showed that $f(w) = e^w$ is convex, so $f(w) = 10e^w$ is convex.

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.

$\|w\|$, $\|w\|^2$, $\|w\|_1$, $\|w\|_\infty$, $\|w\|_1^2$, and so on are all convex.

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.

$$f(w) = \underbrace{10e^{w}}_{\text{From earlier}} + \underbrace{\frac{\lambda}{2}}_{\text{constant}} \underbrace{\|w\|^2}_{\text{norm squared}} \quad \text{is convex}$$

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.

$$f(w) = w^T x_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id}$$

convex convex convex

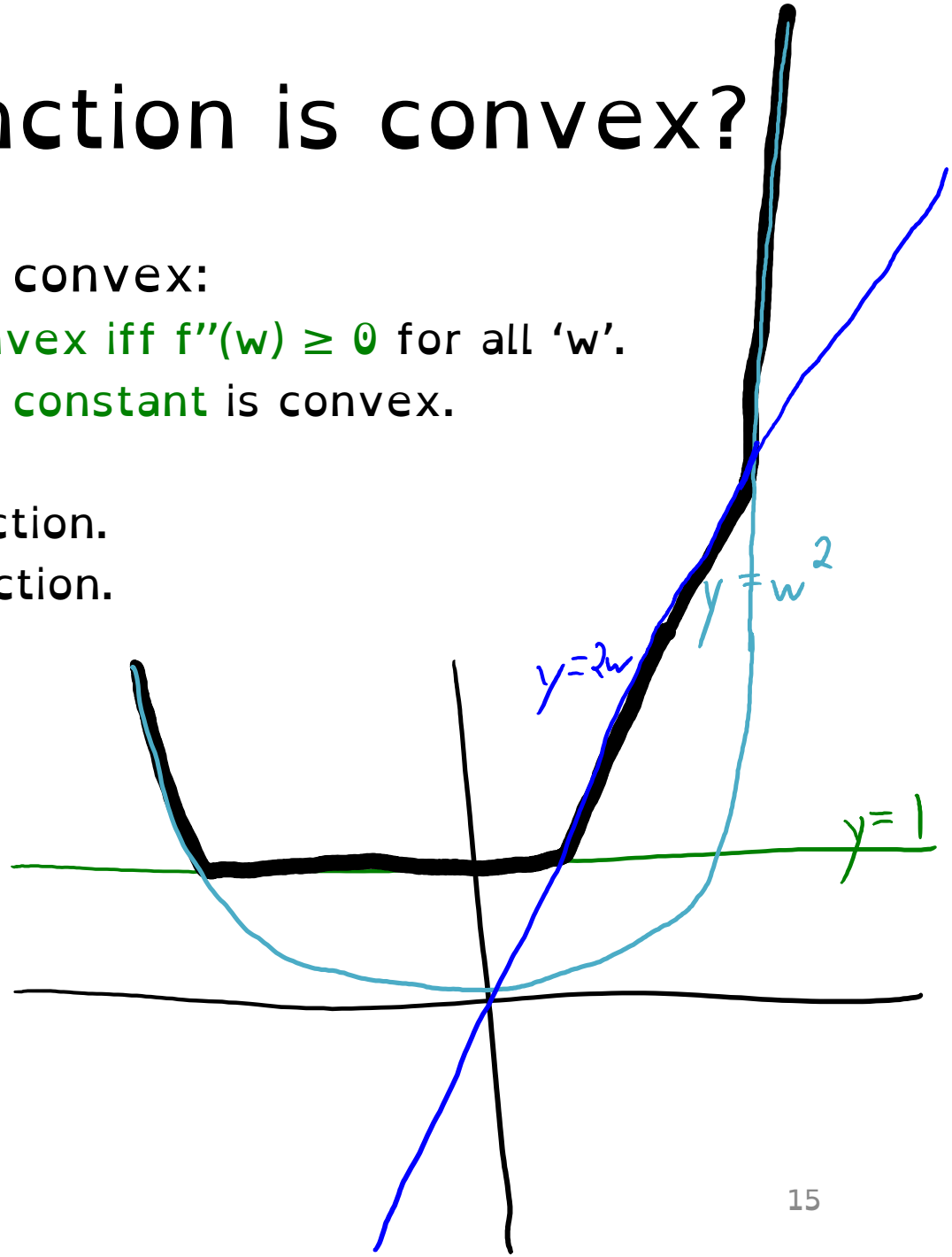
Second derivative of each term is 0.

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.
  - The max of convex functions is a convex function.

$$f(w) = \max\{1, 2w, w^2\}$$ is convex.

convex



How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff f''(w) ≥ 0 for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.
  - The max of convex functions is a convex function.
  - Composition of a convex function and a linear function is convex.

If $f(w) = g(Xw - y)$ then 'f' is convex if 'g' is convex.

linear function

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff $f''(w) \geq 0$ for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.
  - The max of convex functions is a convex function.
  - Composition of a convex function and a linear function is convex.
- But: not true that multiplication of convex functions is convex:
  - If $f(x)=x$ (convex) and $g(x)=x^2$ (convex), $f(x)g(x) = x^3$ (not convex).

How do we know if a function is convex?

- Some useful tricks for showing a function is convex:
  - 1-variable, twice-differentiable function is convex iff $f''(w) \geq 0$ for all 'w'.
  - A convex function multiplied by non-negative constant is convex.
  - Norms and squared norms are convex.
  - The sum of convex functions is a convex function.
  - The max of convex functions is a convex function.
  - Composition of a convex function and a linear function is convex.
- Also not true that composition of convex with convex is convex:

Even if 'f' is convex and 'g' is convex, $f(g(w))$ might not be convex.

E.g., w^2 is convex and $(w-1)^2$ is convex, but $(w^2-1)^2$ is not convex.

Example: Convexity of Linear Regression

- Consider linear regression objective with squared error:

$$f(w) = \|Xw - y\|^2$$

- We can use that this is a _____ composed with _____:

Let $h(w) = Xw - y$, which is a linear function ('d' inputs, 'n' outputs)

Let $g(r) = \|r\|^2$, which is convex because it's a squared norm.

Then $f(w) = g(h(w))$, which is convex because it's a convex function composed with a linear function

Convexity in Higher Dimensions

- Twice-differentiable 'd'-variable function is convex iff:
  - Eigenvalues of Hessian $\nabla^2 f(w)$ are _____ for all 'w'.
    - aka $\nabla^2 f(w)$ is positive semi-definite

- True for least squares where $\nabla^2 f(w) = X^T X$ for all 'w'.
  - It may not be obvious that this matrix has non-negative eigenvalues.

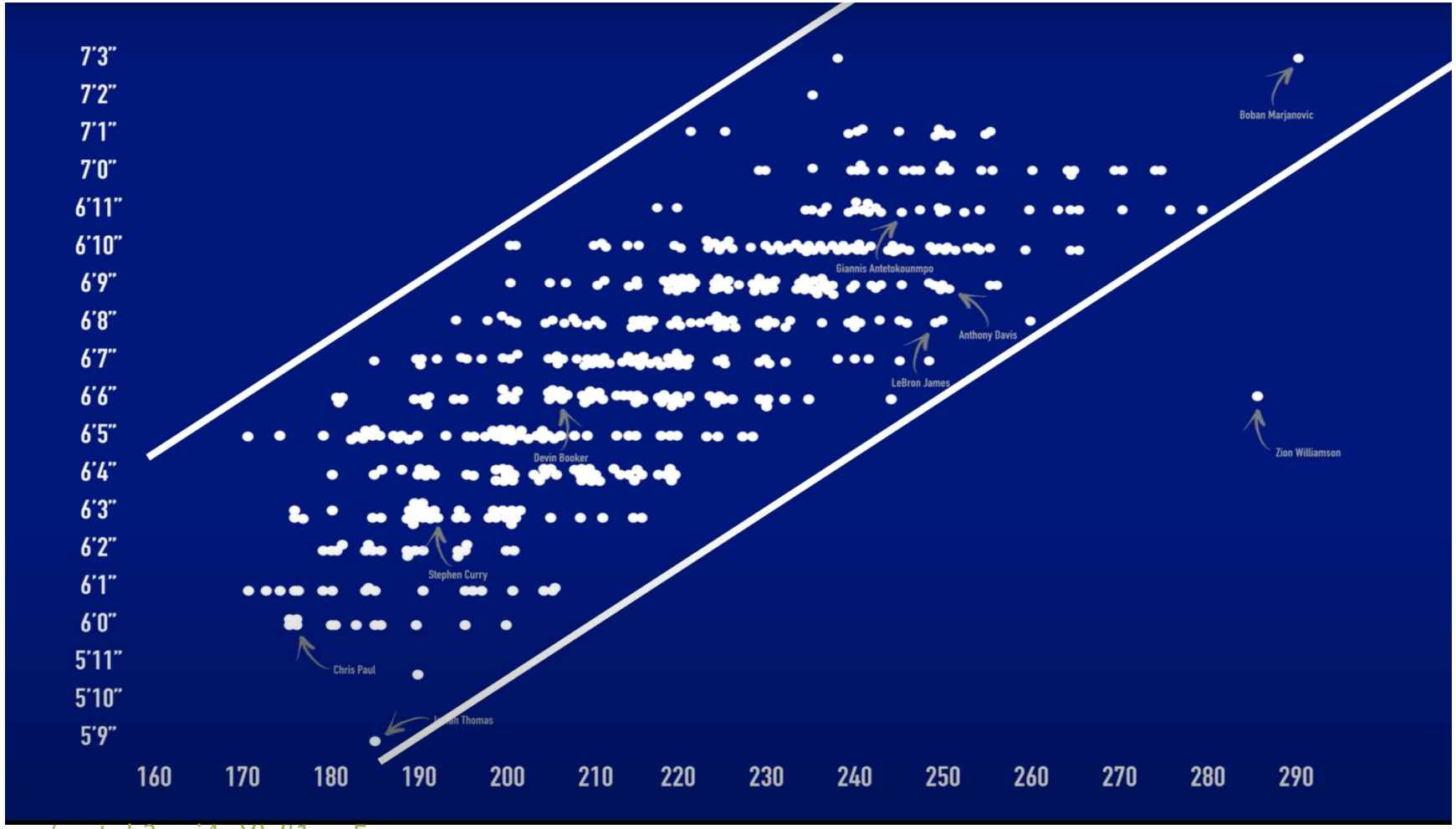
- Unfortunately, sometimes it is hard to show convexity this way.
  - Usually easier to just use some of the rules as we did on the last slide.

Coming Up Next

ROBUST REGRESSION

Least Squares with Outliers

- Height vs. weight of NBA players:

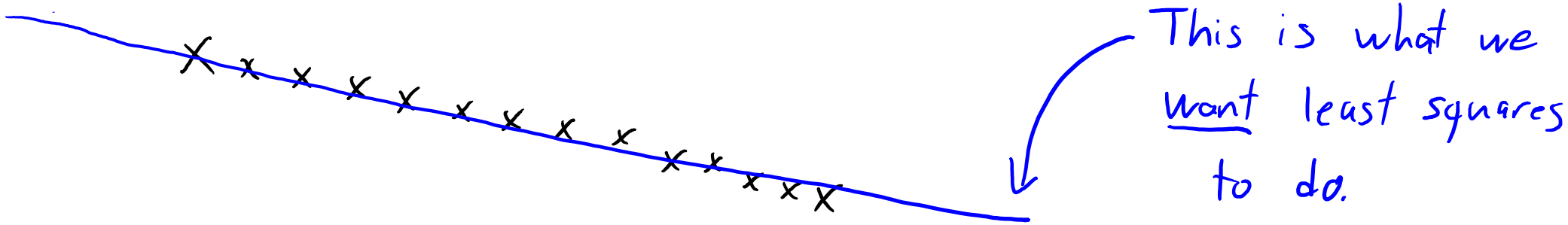


https://www.youtube.com/watch?v=i4eYWl1ewFo

Least Squares with Outliers

- Consider least squares problem with outliers in 'y':

x ← "outlier" that doesn't follow trend

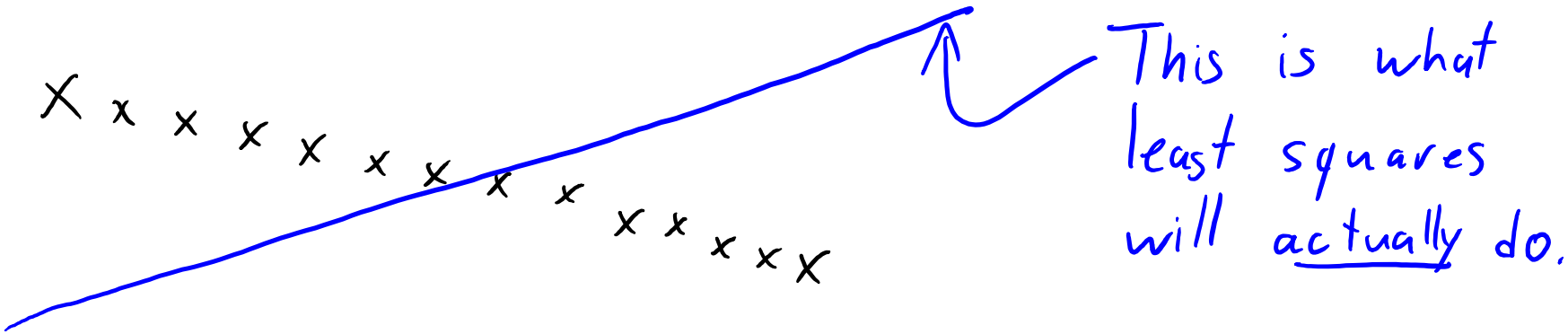


http://setosa.io/ev/ordinary-least-squares-regression

Least Squares with Outliers

- Consider least squares problem with outliers in 'y':

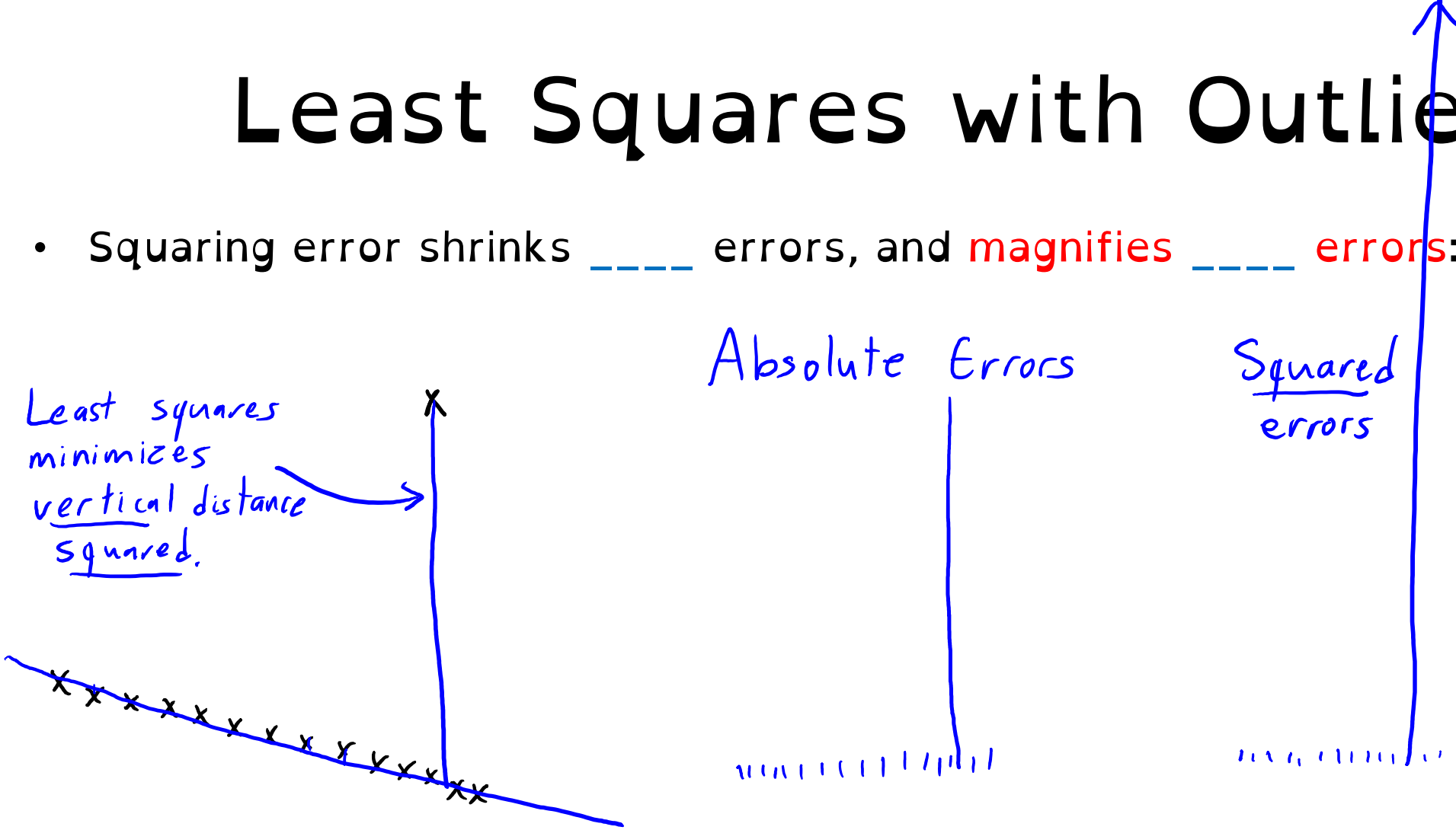
x ← "outlier" that doesn't follow trend



- Least squares is very sensitive to outliers.

Least Squares with Outliers

- Squaring error shrinks ____ errors, and magnifies ____ errors:

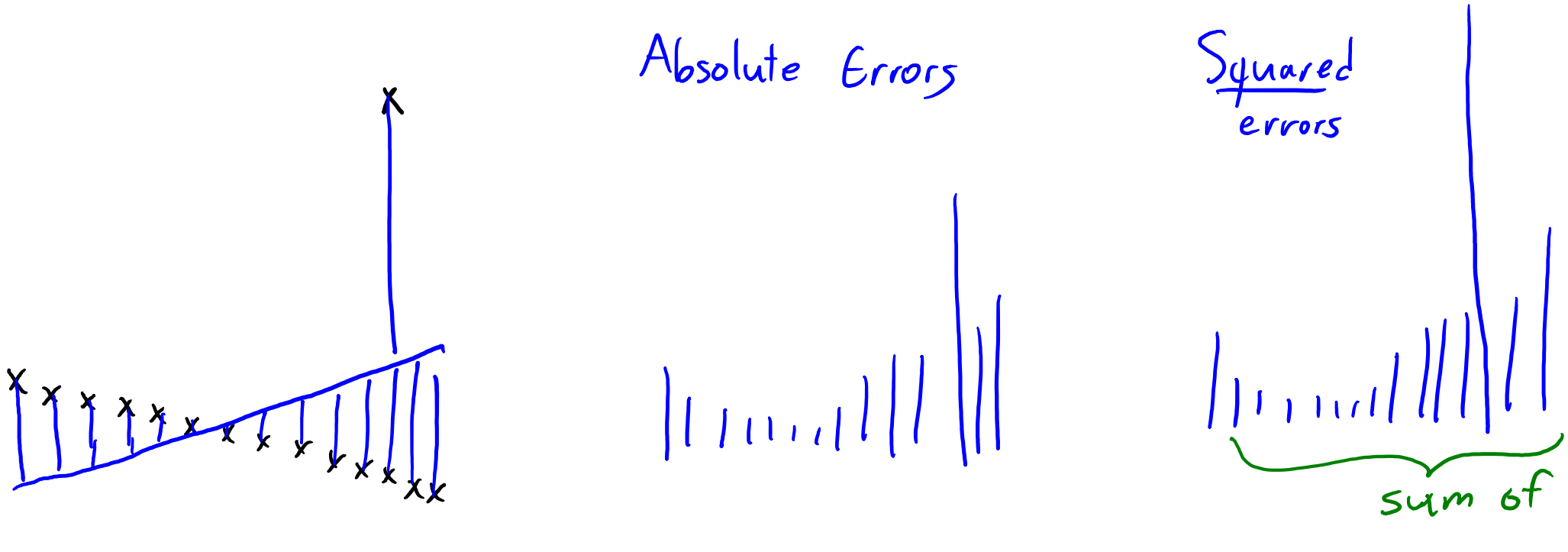


- Outliers (large error) influence 'w' much more than other points.

https://seeing-theory.brown.edu/regression-analysis/index.html

Least Squares with Outliers

- Squaring error shrinks small errors, and magnifies large errors:



- Outliers (large error) influence 'w' much more than other points.
  - Good if outlier means 'plane crashes', bad if it means 'data entry error'.

Robust Regression

- Robust regression objectives focus less on large errors (outliers).
- For example, use absolute error instead of squared error:

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

- Now decreasing 'small' and 'large' errors is equally important.
- Instead of minimizing L2-norm, minimizes _____ of residuals:

Least squares:

$$f(w) = \frac{1}{2}\|Xw - y\|^2$$

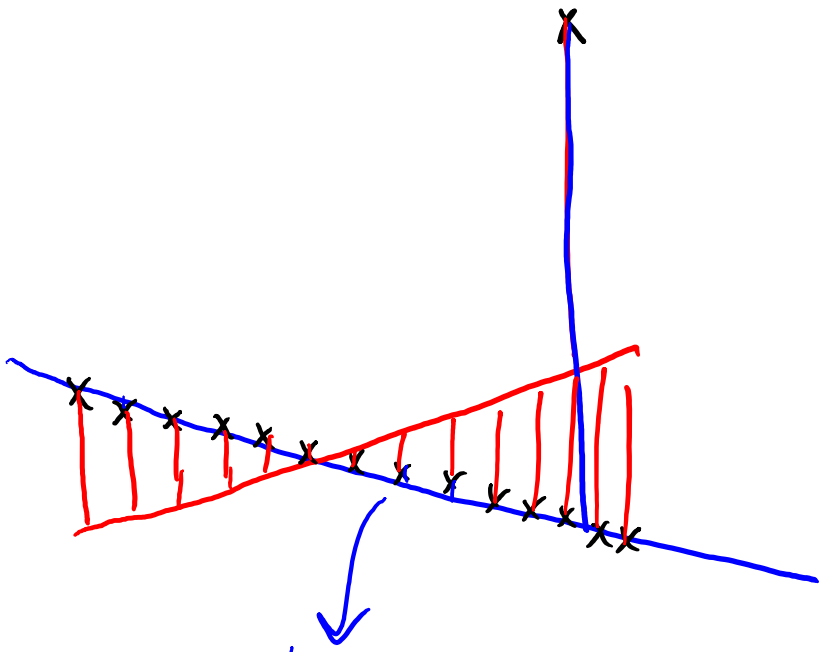
Least absolute error:

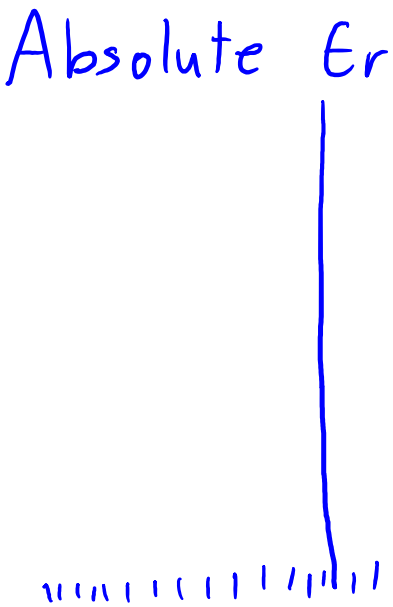
$$f(w) = \|Xw - y\|_1$$

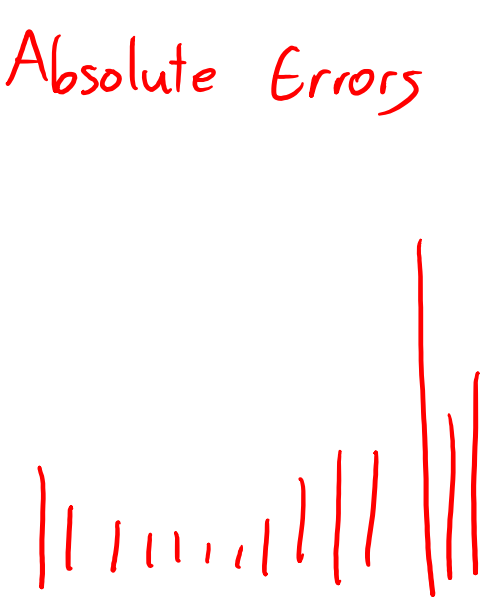
$$\sum_{i=1}^{n} |w^T x_i - y_i| = \sum_{i=1}^{n} |r_i| = \|r\|_1 = \|Xw - y\|_1$$

Least Squares with Outliers

- Absolute error is more robust to outliers:



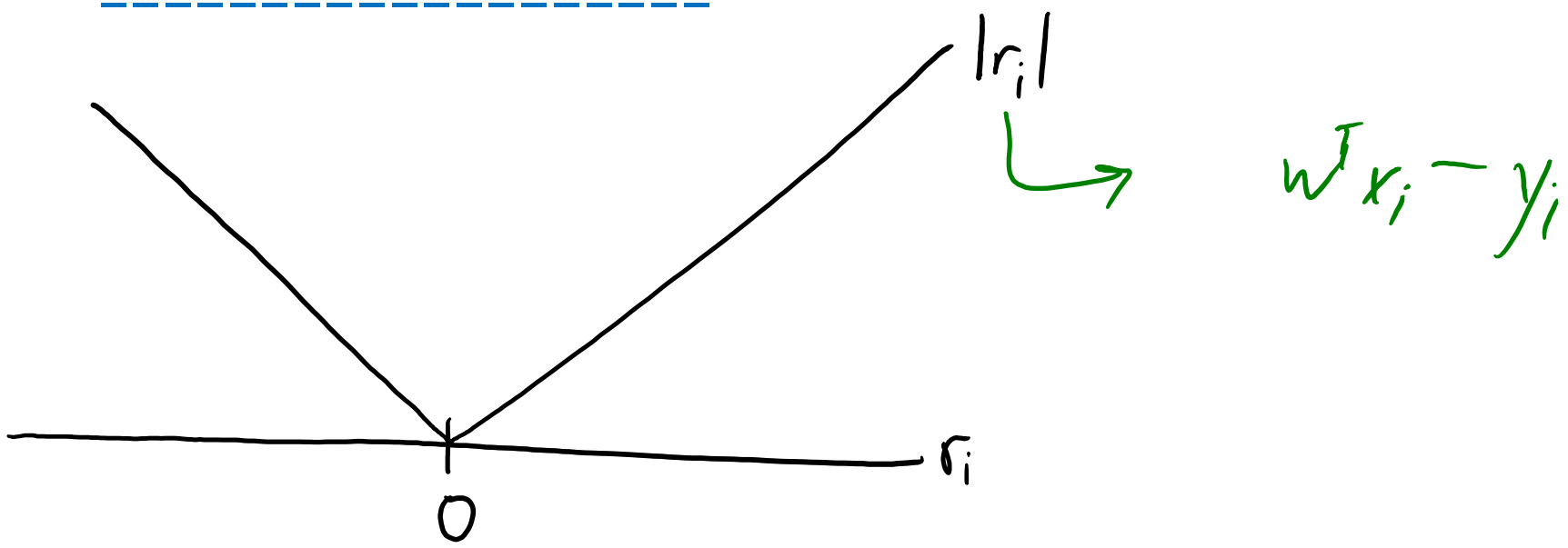




Linear model 'w' minimizing $f(w) = \|Xw - y\|_1 = \sum_{i=1}^{n} |w^T x_i - y_i|$

Regression with the L1-Norm

- Unfortunately, minimizing the absolute error is harder.
  - We don't have "normal equations" for minimizing the L1-norm.
  - Absolute value is _____ at 0.



  - Generally, harder to minimize non-smooth than smooth functions.
    - Unlike smooth functions, the gradient may not get smaller near a minimizer.

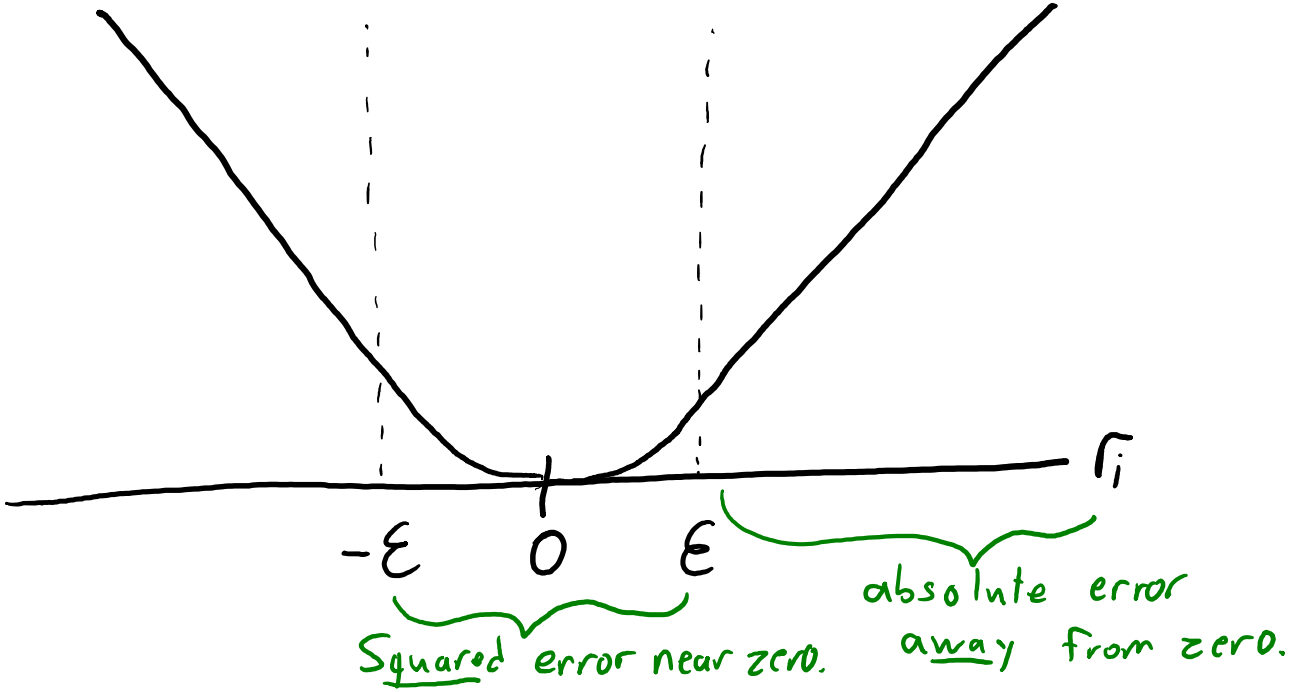
Q: What can we do to apply gradient descent?

Smooth Approximations to the L1-Norm

- There are differentiable approximations to absolute value.
  - Common example is Huber loss:

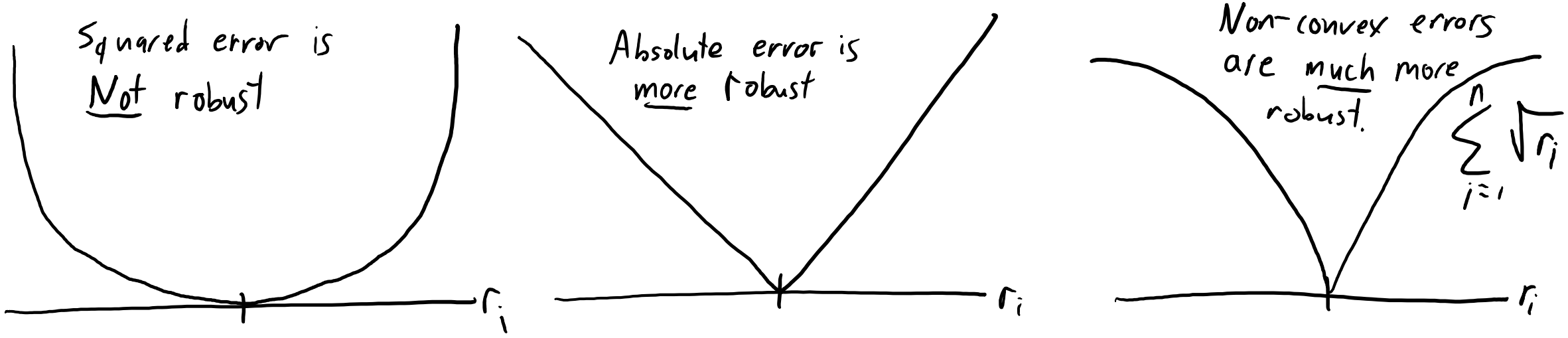
$$f(w) = \sum_{i=1}^{n} h(w^T x_i - y_i)$$

$$h(r_i) = \begin{cases} \frac{1}{2} r_i^2 & \text{for } |r_i| \leq \varepsilon \\ \varepsilon(|r_i| - \frac{1}{2}\varepsilon) & \text{otherwise} \end{cases}$$

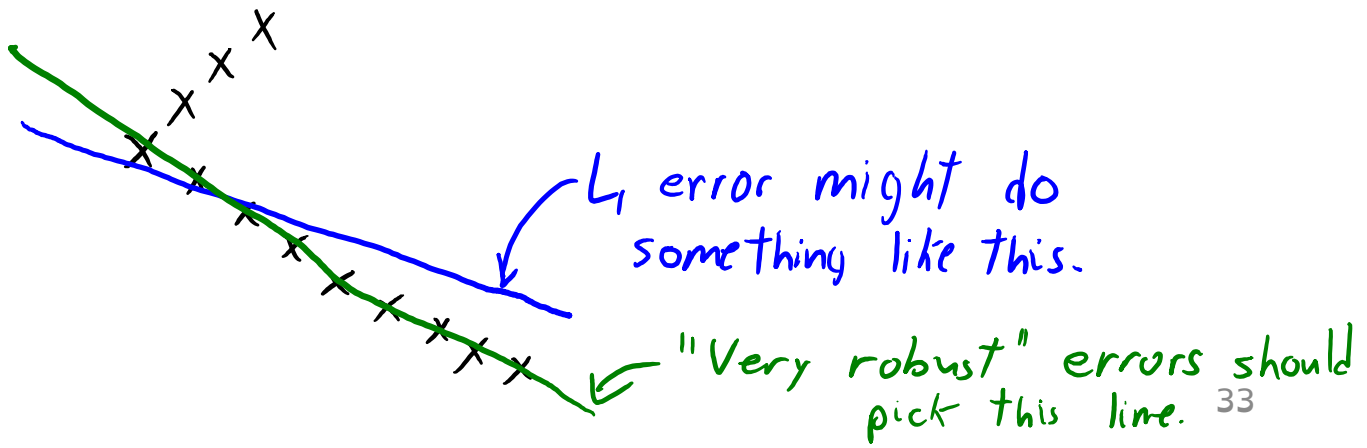


  - Note that 'h' is differentiable: if $|r_i| \leq \varepsilon$, $h'(r_i) = r_i$. else: $h'(r_i) = \begin{cases} +\varepsilon & \text{if } r_i > 0 \\ -\varepsilon & \text{otherwise} \end{cases}$
  - This 'f' is convex but setting $\nabla f(x) = 0$ does not give a _____.
    - But we can minimize the Huber loss using gradient descent.

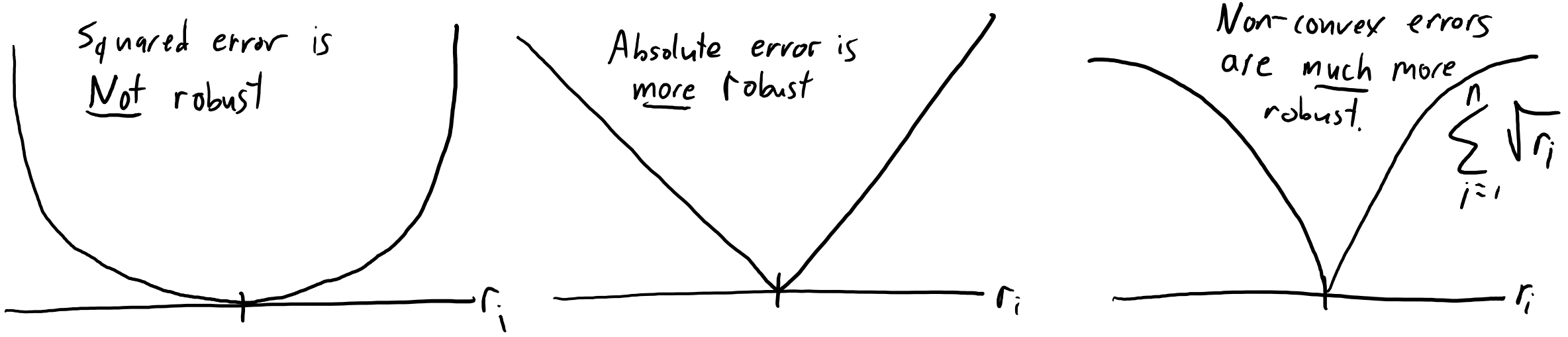
Very Robust Regression



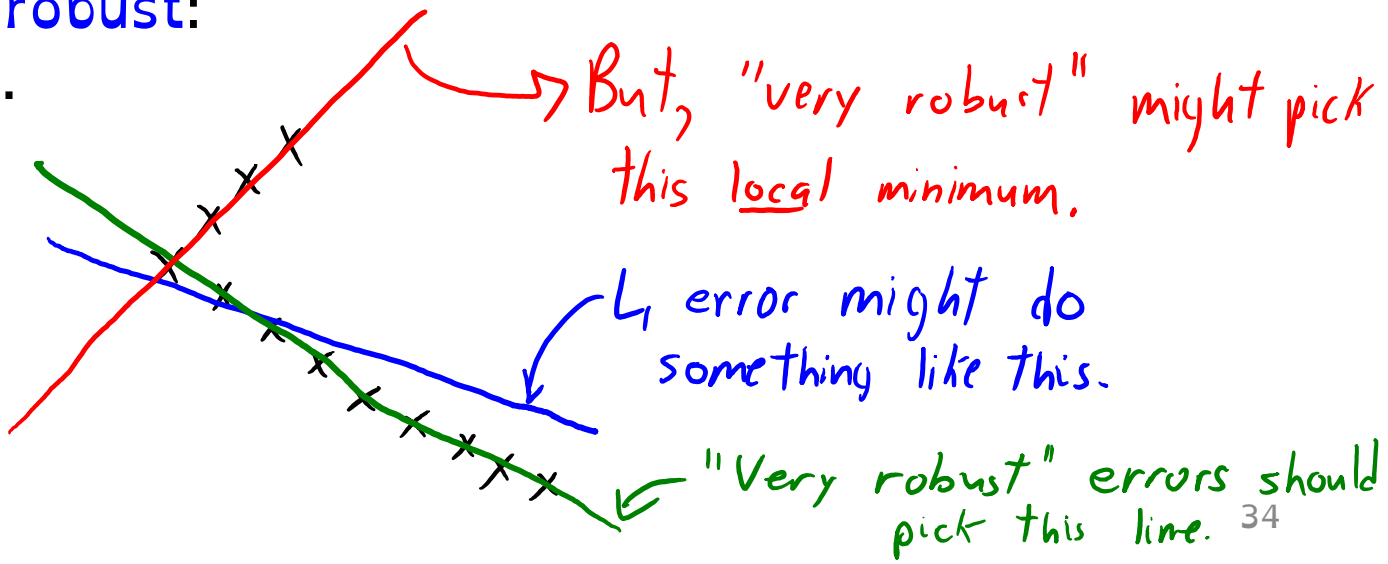
- Non-convex errors can be very robust:
  - Not influenced by outlier groups.



Very Robust Regression



- Non-convex errors can be very robust:
  - Not influenced by outlier groups.
  - But non-convex, so finding global minimum is hard.
  - Absolute value is "most robust" convex loss function.



Coming Up Next

BRITTLE REGRESSION

Motivation for Considering Worst Case

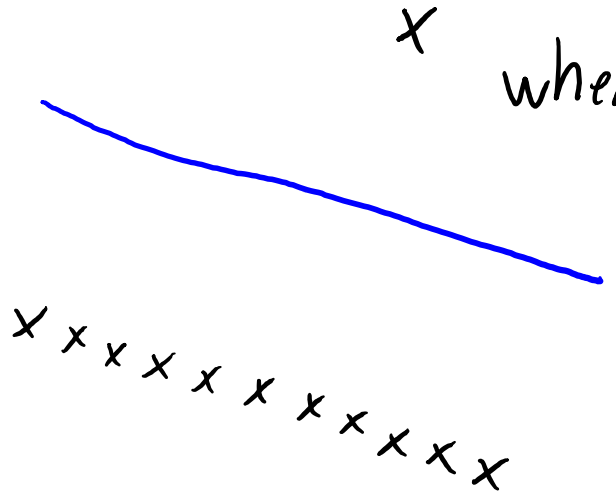
THE PROBLEM WITH AVERAGING STAR RATINGS

https://xkcd.com/937/

"Brittle" Regression

- What if you really care about getting the outliers right?
  - You want best performance on _____.
  - For example, if in worst case the plane can crash.
- In this case you could use something like the infinity-norm:

$$f(w) = \|Xw - y\|_\infty \quad \text{where} \quad \|r\|_\infty = \max_i \{|r_i|\}$$



- Very sensitive to outliers ("brittle"), but worst case will be better.

Log-Sum-Exp Function

- As with the L_1-norm, the L_∞-norm is _____:
  - We can again use a smooth approximation and fit it with gradient descent.

- Convex and smooth approximation to max function is log-sum-exp function:

$$\max_i\{z_i\} \approx \log\left(\sum_i \exp(z_i)\right)$$

  - We'll use this several times in the course.
  - Notation alert: when I write "log" I always mean "natural" logarithm: log(e) = 1.

- Intuition behind log-sum-exp:
  - $\sum_i \exp(z_i) \approx \max_i \exp(z_i)$, as largest element is magnified exponentially (if no ties).
  - And notice that $\log(\exp(z_i)) = z_i$.

Log-Sum-Exp Function Examples

- Log-sum-exp function as smooth approximation to max:

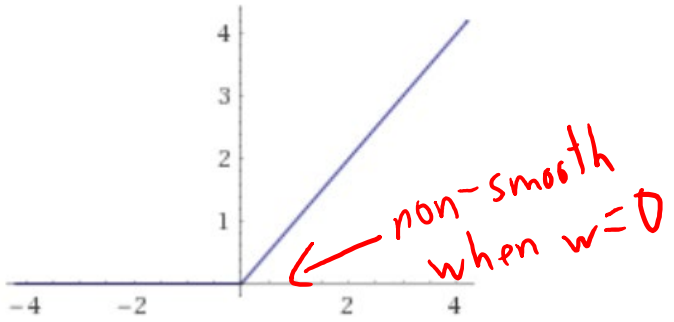
$$\max_i \{z_i\} \approx \log\left(\sum_i \exp(z_i)\right)$$

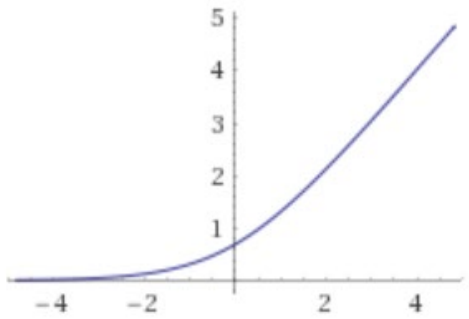
- If there aren't "close" values, it's really close to the max.

If $z_i = \{2, 20, 5, -100, 7\}$ then $\max_i \{z_i\} = 20$ and $\log\left(\sum_i \exp(z_i)\right) \approx 20.000002$

If $z_i = \{2, 20, 19.9, -100, 7\}$ then $\max_i \{z_i\} = 20$ and $\log\left(\sum_i \exp(z_i)\right) \approx 20.688160$

- Comparison of max{0,w} and smooth log(exp(0) + exp(w)):





Part 3 Key Ideas: Linear Models, Least Squares

- Focus of Part 3 is linear models:
  - Supervised learning where prediction is linear combination of features:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id}$$
$$= w^T x_i$$

- Regression:
  - Target y_i is numerical, testing ($\hat{y}_i == y_i$) doesn't make sense.

- Squared error: $\frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$ or $\frac{1}{2}\|Xw - y\|^2$

  - Can find optimal 'w' by solving "normal equations".

Part 3 Key Ideas: Change of Basis, Gradient Descent

- Change of basis: replaces features x_i with non-linear transforms z_i:
  - Add a bias variable (feature that is always one).
  - Polynomial basis.
  - Other basis functions (logarithms, trigonometric functions, etc.).

- For large 'd' we often use gradient descent:
  - Iterations only cost O(nd).
  - Converges to a critical point of a smooth function.
  - For convex functions, it finds a global optimum.

Part 3 Key Ideas: Error Functions, Smoothing

- Error functions:
  - Squared error is sensitive to outliers.
  - Absolute (L_1) error and Huber error are more robust to outliers.
  - Brittle (L_∞) error is more sensitive to outliers.
- L_1 and L_∞ error functions are convex but non-differentiable:
  - Finding 'w' minimizing these errors is harder than squared error.
- We can approximate these with differentiable functions:
  - L_1 can be approximated with Huber.
  - L_∞ can be approximated with log-sum-exp.
- With these smooth (convex) approximations, we can find global optimum with gradient descent.

End of Scope for Midterm Material.

(we're not done Part 3, but nothing after this point will be tested on the midterm)

Summary

- Convex functions:
  - Set of functions with property that $\nabla f(w) = 0$ implies 'w' is a global min.
  - Can (usually) be identified using a few simple rules.
- Least squares with outliers:
  - Reduce influence of outliers, or magnify influence of outliers
  - Use L_1 and L_∞ error functions, which are convex but non-smooth
  - Use smooth approximations with gradient descent to optimize
    - e.g. Huber loss, log-sum-exp, etc.

- Next time:
  - Bonus lecture! Machine learning for Amazon and Games

Review Questions

- Q1: Do convex functions have a unique global minimum?
- Q2: Why is $L_{1/2}$-norm considered non-convex even though L_p-norms are considered convex?
- Q3: Why are there no normal equations for robust and brittle regressions?
- Q4: Why is non-smoothness a problem for gradient descent?

question

stop following 81 views

Norms Norms Norms: Getting from Sums to Norms

I was going over the solutions for A3 and I am still a bit confused on how to get from a sum to a norm in some situations. I know the basic ones that give me $||Xw - y||^2$ and stuff, but when other things are thrown in the mix I get a bit confused. For example, $\sum_{i=1}^{n} v_i(w^T x_i - y_i)^2$ gives $||V^{1/2}(Xw - y)||^2$. From my understanding v is a vector and vi is the number at position I in that vector. How does the summation of these indices result in the diagonal matrix V and not just the vector v?

Furthermore, when we have a summation like $\sum_{j=1}^{d} \lambda_j |w_j|$, it is simplified to $||\Lambda w||_1$. How does the lambda end up inside the L1 norm? I thought that a summation could be simplified to a L1 norm if its terms are wrapped around the absolute value symbol. In this case the lambda is not, so how is it able to appear inside the norm like that?

hw3 midterm_exam

the instructors' answer, *where instructors collectively construct a single answer*

I know that this notation seems intimidating if this is the first time you see it. Fortunately, there are really only a few "rules" you need to figure out, and you'll find that these are use all over the place.

For those particular questions you'll want to memorize the way that the three common norms appear: $\sum_{i=1}^{n} |r_i| = ||r||_1, \sum_{i-1}^{n} r_i^2 = ||r||^2, \max_{i \in \{1,2,\ldots,n\}}\{|r_i|\} = ||r||_\infty$. So when you see max, sum of non-negative values, or sum of squared values you should think of these norms.

Next, notice what multiplying by a diagonal matrix does: if you multiply a vector w (for example) by a diagonal matrix then you multiply each element w_i by the corresponding diagonal element. If you multiply matrix X (for example) by a diagonal matrix then you multiply each row of X by the corresponding diagonal element.

The $V^{1/2}$ shows up because we're multiplying the square.

The other really useful ones to know are $\sum_{i=1}^{n} v_i r_i = v^T r$ if the elements aren't necessarily non-negative, $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j = x^T A x$, and $\sum_{i=1}^{n} x_i r_i = X^T r$.

(All of the above follow from definitions, but it takes some practice to recognize these common forms. That's why we made you get some practice on the assignments, and why we covered this notation before the midterm so that you study it before we start using it a lot. It is incredibly common the ML world.)

edit · good answer | 2

Updated 8 months ago by Mark Schmidt

Constraints, Continuity, Smoothness

- Sometimes we need to optimize with constraints:
  - Later we'll see "non-negative least squares".

$$\min_{w \geq 0} \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

  - A vector 'w' satisfying $w \geq 0$ (element-wise) is said to be "feasible".
- Two factors affecting difficulty are continuity and smoothness.
  - Continuous functions tend to be easier than discontinuous functions.
  - Smooth/differentiable functions tend to be easier than non-smooth.
  - See the calculus review here if you haven't heard these words in a while.

Convexity, min, and argmin

- If a function is convex, then all critical points are global optima.
- However, convex functions don't necessarily have critical points:
  - For example, f(x) = a*x, f(x) = exp(x), etc.
- Also, more than one 'x' can achieve the global optimum:
  - For example, f(x) = c is minimized by any 'x'.

Why use the negative gradient direction?

- For a twice-differentiable 'f', multivariable Taylor expansion gives:

$$f(w^{t+1}) = f(w^t) + \nabla f(w^t)^T (w^{t+1} - w^t) + \frac{1}{2}(w^{t+1} - w^t) \nabla^2 f(v) (w^{t+1} - w^t)$$

for some 'v' between w^{t+1} and w^t.

- If gradient can't change arbitrarily quickly, Hessian is bounded and:

$$f(w^{t+1}) = f(w^t) + \nabla f(w^t)^T (w^{t+1} - w^t) + O(\|w^{t+1} - w^t\|^2)$$

becomes negligible as w^{t+1} gets close to w^t

  - But which choice of w^{t+1} decreases 'f' the most?
    - As $\|w^{t+1} - w^t\|$ gets close to zero, the value of w^{t+1} minimizing $f(w^{t+1})$ in this formula converges to $(w^{t+1} - w^t) = -\alpha^t \nabla f(w^t)$ for some scalar α^t.
    - So if we're moving a small amount, the optimal w^{t+1} is:

$$w^{t+1} = w^t - \alpha_t \nabla f(w^t)$$ for some scalar α_t.

Normalized Steps

Question from class: "can we use $w^{t+1} = w^t - \frac{1}{\|\nabla f(w^t)\|} \nabla f(w^t)$"

This will work for a while, but notice that

$$\|w^{t+1} - w^t\| = \left\| \frac{1}{\|\nabla f(w^t)\|} \nabla f(w^t) \right\|$$

$$= \frac{1}{\|\nabla f(w^t)\|} \|\nabla f(w^t)\|$$

$$= 1$$

So the algorithm <u>never converges</u>

optimizer.py Details

question ☆ stop following 99 views

The minimizer function

Hi all,

I'm just curious how the minimizers given to us works. Are there any resources can give us more details about it?

the instructors' answer, *where instructors collectively construct a single answer*

It's just a basic gradient descent implementation with some clever guesses for the step-size.

The step-size on each iteration is initialized using the method from this classic paper (which works surprisingly well but we don't really know why except in two dimensions):
http://pages.cs.wisc.edu/~swright/726/handouts/barzilai-borwein.pdf

That step-size is evaluated using a standard condition ("Armijo condition") and then it fits a polynomial regression model based on the function and directional derivative values and tries the step-size minimizing this polynomial. Both these tricks are described in Nocedal and Wright's "Numerical Optimization" book.

edit · good answer | 3 Updated 7 months ago by Mark Schmidt