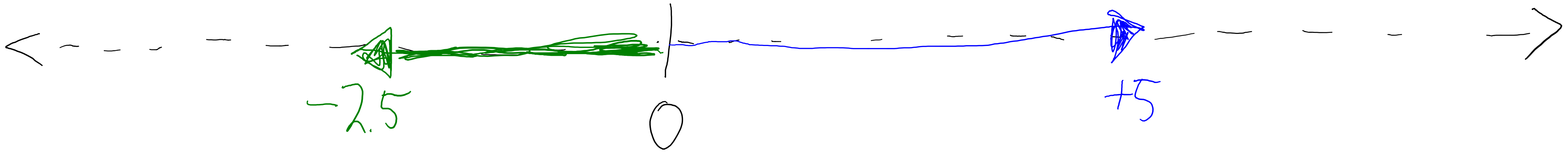
# CPSC 340: Machine Learning and Data Mining

Feature Selection

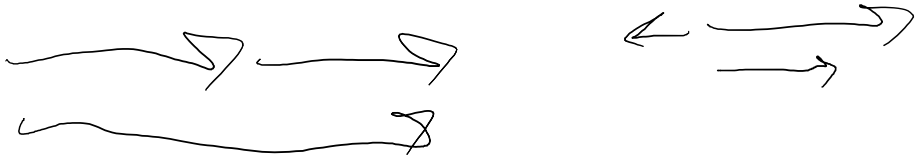
Fall 2015

# Norms in 1-Dimension

- We can view absolute value, $|x|$, as 'size' or 'length' of a number:




- It satisfies three intuitive properties of 'length':
  1. Only '0' has a 'length' of zero.
  2. If you multiply 'x' by constant '$\alpha$', length gets multiplied by $|\alpha|$.
  3. Length of 'x+y' is not more than length of 'x' plus length of 'y'.
     "Triangle inequality"

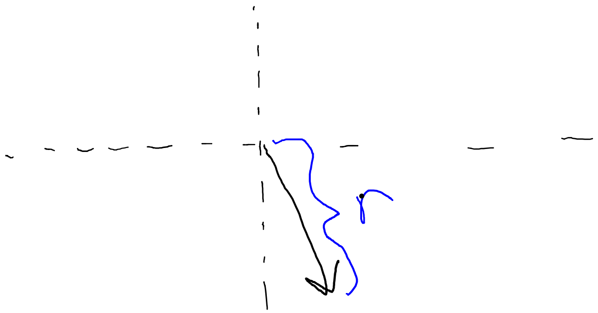


# Norms in 2-Dimensions

- In 1-dimension, only the absolute value satisfies the 3 properties.
- In 2-dimensions, there is no unique function satisfying them.
- We call any function satisfying them a ‘norm’:
  - These are measures of ‘length’ in 2-dimensions.
- Three most common examples:

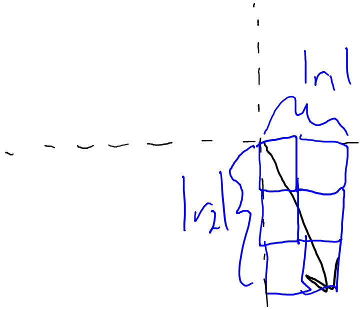
$L_2$ or "Euclidean" norm:

$$\|r\|_2 = \sqrt{r_1^2 + r_2^2}$$




$L_1$ or "Manhattan" norm:

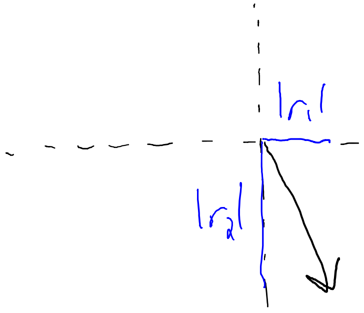
$$\|r\|_1 = |r_1| + |r_2|$$




5 "blocks"

$L_\infty$ or "max" norm:

$$\|r\|_\infty = \max\{|r_1|, |r_2|\}$$




3 "blocks" in longest direction.

# Norms in d-Dimensions

- These norms also satisfy the 3 properties in d-dimensions:

$L_2$: $\|r\|_2 = \sqrt{\sum_{i=1}^{d} r_i^2}$

$L_1$: $\|r\|_1 = \sum_{i=1}^{d} |r_i|$

$L_\infty$: $\|r\|_\infty = \max_i \{|r_i|\}$

If 'r' is a vector containing residual $(y_i - w^T x_i)$ of a linear regression model, norm of residual measures 'size' of error in some way:

$L_1$: all errors are equal

$L_2$: bigger errors are more important.

$L_\infty$: only biggest error is important.

# Summary of Last Lecture

1. Error function (L2, L1, Huber) affects how errors are 'weighted'.
2. L1 and L∞ error functions are non-differentiable:
   – Finding 'w' minimizing these errors is harder.
3. We can approximate these with differentiable functions:
   – L1 can be approximated with Huber.
   – L∞ can be approximated with log-sum-exp.
4. Gradient descent finds local minimum of differentiable function.
5. For convex functions, any local minimum is a global minimum.
   – Non-convex minimization is very hard, but some people do it anyways.
   – Starting from different initializations can help!

# Motivation: Allergy Testing with Regression

- Recall the food allergy example:

| Egg | Milk | Fish | Wheat | Shellfish | Peanuts | … |
|---|---|---|---|---|---|---|
| 0 | 0.7 | 0 | 0.3 | 0 | 0 | |
| 0.3 | 0.7 | 0 | 0.6 | 0 | 0.01 | |
| 0 | 0 | 0 | 0.8 | 0 | 0 | |
| 0.3 | 0.7 | 1.2 | 0 | 0.10 | 0.01 | |

| Sick? |
|---|
| 1 |
| 1 |
| 0 |
| 1 |

# Motivation: Allergy Testing with Regression

- Instead of sick/not-sick, consider measuring immunoglobin levels:

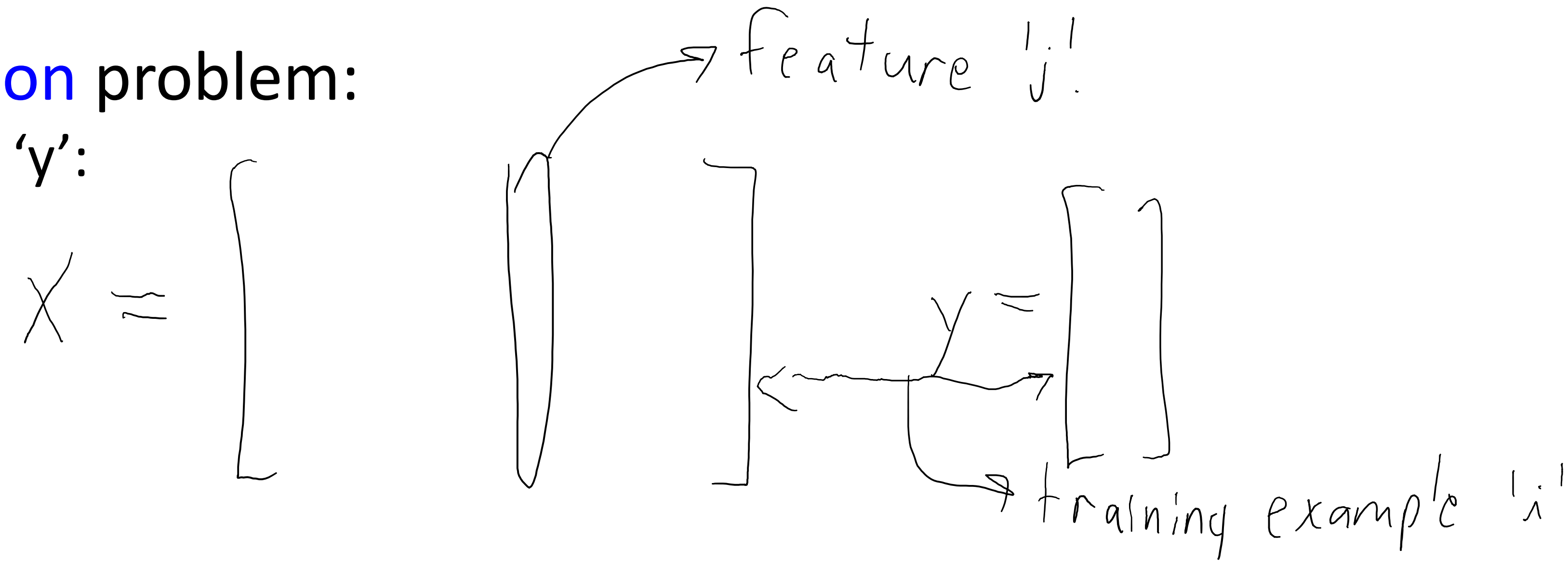
| Egg | Milk | Fish | Wheat | Shellfish | Peanuts | … |
|---|---|---|---|---|---|---|
| 0 | 0.7 | 0 | 0.3 | 0 | 0 | |
| 0.3 | 0.7 | 0 | 0.6 | 0 | 0.01 | |
| 0 | 0 | 0 | 0.8 | 0 | 0 | |
| 0.3 | 0.7 | 1.2 | 0 | 0.10 | 0.01 | |

| IgE |
|---|
| 700 |
| 740 |
| 50 |
| 950 |

- Now formulated as a regression problem.
- Instead of prediction, want to find out which foods cause high IgE.
  - 'Feature selection' (similar to finding association rules).
  - Similar to choosing degree in polynomial basis, but there is no ordering.

# Feature Selection

- General feature selection problem:
  - Given our usual 'X' and 'y':




  - We think some features/columns of 'X' are irrelevant to predict 'y'.
- Output could be:
  - Set of 'relevant' features. ('milk', 'oranges', 'ice cream')
  - A model that uses the 'best' set of features.
- One of most important problems in ML/statistics, but very very messy.
  - We focus on linear regression, but ideas apply for classification/non-linear.

# Choose the Largest Regression Weights?

- Simple/common approach to feature selection:
  1. Fit least squares weights ‘w’ using all the features.
  2. Choose the features ‘j’ with biggest weights, $|w_j|$.
- Intuitive: big $|w_j|$ means ‘$x_{ij}$’ has big affect on ‘$y_i$’.
  – E.g., we expect ‘$w_j$’ for ‘milk’ feature to be high: high milk => high IgE.
- Only makes sense if feature have independent effects:
  – Otherwise, systematically misses relevant variables.
- Example:
  – You are allergic to ‘lactose’ and to a protein in ‘milk’.
  – But you only ever see ‘lactose’ and ‘milk’ together.
  – Linear regression could give big weight to ‘milk’ and smaller weight to ‘lactose’.
  – Or could be reversed, or both could be medium-sized.

# Choose the Largest Correlations?

- Another simple/common approach to feature selection:
  - For each feature 'j', compute similarity between all '$x_{ij}$' and '$y_i$'.
    - E.g., correlation, distance, mutual information, etc.
  - Return top 'k' features, or all features above some threshold.
- Not sensitive to milk-lactose issue:
  - Uses independent statistic on each variable.
  - Finds that 'milk' and 'lactose' are relevant.
- Sensitive to the Taco Tuesday issue:
  - You could find that 'Tuesday' is very correlated with IgE level.
  - But only because you go to Taco Tuesdays:
    - If you knew the value of 'Taco', the variable 'Tuesday' is irrelevant.
- This approach systematically includes irrelevant variables.

# Philosophical Digression

- Is ‘Tuesday’ actually a relevant variable?
  - If you don’t know the value of ‘taco’, *it is relevant* for prediction.
  - So ‘relevance’ is relative to what other information is available.

- A second issue with this example is causality:
  - ‘Tuesday’ *does not directly cause* an increase in IgE, so it is not relevant.
  - But if you don’t have an ‘intervention’ like ‘forced not to go to taco Tuesdays’, you may never be able to determine this.
  - Similarly, ‘histamine’ is relevant for predicting IgE, but IgE causes histamine.

- If the effect size is very small, is the variable relevant?
  - Presumably, any variable could give some information about $y_i$.
  - We are probably *only interested in non-trivial* effect sizes.

# Common Approaches to Feature Selection

- 3 main approaches to feature selection:
  1. Hypothesis testing.
  2. Search and score.
  3. L1-Regularization.
- None is ideal, but good to know advantages/disadvantages.

1. Hypothesis testing or ‘constraint-based’ approaches:
   – Fixes ‘largest correlation’ method to address Taco Tuesday.
   – Assumes we have test of conditional dependence:
     - Usually, ‘partial’ correlation or ‘conditional’ mutual information.

# Hypothesis Testing in Action

- Testing whether ‘taco’ is relevant: e.g., high cosine similarity
  - Test if ‘taco’ and ‘IgE’ are dependent:
    - Yes, they are.
  - Next test if ‘taco’ and ‘IgE’ are dependent, given ‘Tuesday’:
    - Yes, they still are: return ‘relevant’.
- Testing whether ‘Tuesday’ is relevant:
  - Test if ‘Tuesday’ and ‘IgE’ are dependent:
    - Yes, they are.
  - Next test if ‘Tuesday’ and IgE’ are dependent, given ‘taco’:
    - No, they are not: return ‘not relevant’.

# Feature Selection Approach 1: Hypothesis Testing

- Constraint-based determination of whether ‘j’ is relevant:
  1. Start with an empty ‘conditioning set’ ‘S’.
  2. Test whether ‘$x_{ij}$’ and ‘$y_i$’ are dependent.
     - If not, return ‘not relevant’.
  3. Choose some variable and add it to ‘S’.
  4. Test whether ‘$x_{ij}$’ and ‘yi’ are conditionally dependent given ‘S’.
     - If not, return ‘not relevant’.
     - Otherwise, return to step 3 until we have added all variables to ‘S’.
  5. If all variables are in ‘S’, return ‘relevant’.

# Hypothesis Testing Issues

- Advantages:
  - Deals with Taco Tuesday issue.
  - Algorithm can *explain* decisions.
  - Allows fancy non-parametric measures of dependence.
- Disadvantage:
  - Usual warning about testing multiple hypotheses.
  - You could be 'dependent' but with trivial effect size.
  - Does not deal with milk-lactose issue:
    - Could say they are both irrelevant.
    - 'Faithfulness' assumption: pretend things like this can't happen.
  - Hard to determine optimal order that you add variables to 'S'.

milk independent of IgE given lactose

lactose independent of IgE given milk.

# Feature Selection Approach 2: Search and Score

- Two components behind search and score methods:
  - Score: function that says how 'good' a set of variables are.
  - Search: find set of variables with a high score.

there are so many combinations
you will find
some combination
where an irrelevant
variable happens
to improve
validation
error.

- Usual score functions:
  1. Validation/cross-validation error:
     - Good if your main goal is prediction.
     - Prone to false positives: tends to add irrelevant variables due to overfitting.
  2. L0 "norm":
     - Balance training error against number of non-zero features.

# L0-Norm

Why do we care about $(w_j == 0)$?

$y_i = w_1 x_{i1} + w_2 x_{i2} + \cdots$

- The L0 "norm" is the number of non-zero values.
  - Not actually a norm: violates 2 of 3 properties.
- L0-norm regularization for features selection:

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|y - Xw\|^2 + \lambda \|w\|_0.$$

if $(w_2 == 0)$ ignore $x_{i2}$

So $(w_j == 0)$ is equivalent to removing feature "$j$".

- Balances between training error and number of features.
  - Different values of λ give the common feature selection scores:
    - Akaike information criterion (AIC).
    - Bayesian information criterion (BIC).
    - Both recover correct features under strong assumptions.

# Search and Score Issues

- Advantages:
  - Deals with Taco Tuesday issue.
  - Takes into account size of the effect.
- Disadvantages:
  - Difficult to define ‘correct’ score:
    - Cross-validation often selects too many.
    - L0-norm selects too few/many depending on $\lambda$.
  - Only partially deals with milk-lactose issue:
    - L0-norm will only pick one of them.
    - Cross-validation could pick one or both.
  - Under most scores, it’s hard to find optimal features.

# Practical Search Methods

- Usual search procedures:
  1. Exhaustive search:
     - Returns optimal solution, but only feasible if ‘d’ is very small.
  2. Forward selection:
     - Start with no features, add the one that increase the score the most, repeat.
     - Sub-optimal, but often works well.
  3. Backward selection:
     - Start with all features, remove the one that decreases the score the most, repeat.
  4. Stagewise: combine forward/backward selection.

# Feature Selection Approach 3: L1-Regularization (LASSO)

- Consider regularizing by the L1-norm:

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|y - Xw\|^2 + \lambda\|w\|_1$$

- Like L2-norm, it's convex and has many magical properties.
- Like L0-norm, it encourages elements of 'w' to be exactly zero.

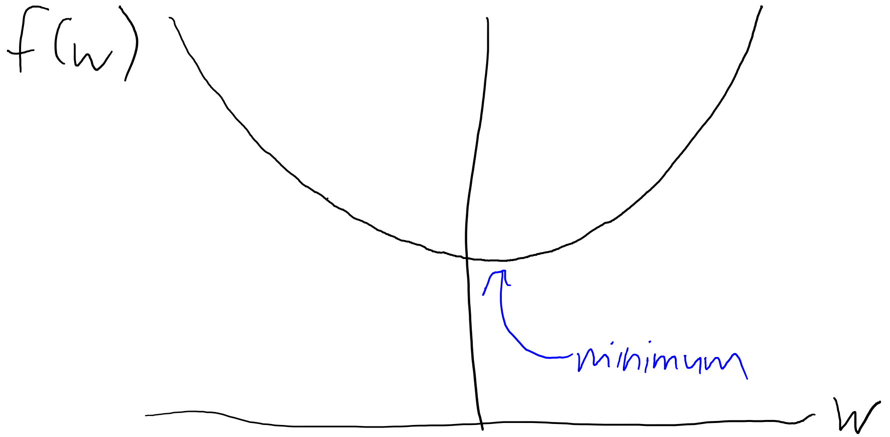
- We call a vector with many elements set to 0 a sparse vector.
- We can simultaneously regularized and select features.
  - And it's very fast, too.

# Sparsity and Least Squares

- Consider 1D least squares objective:

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

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



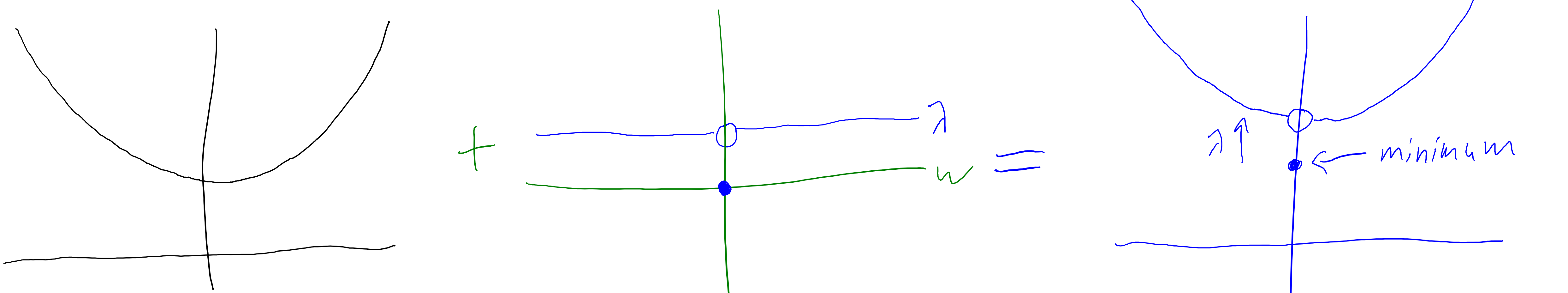

- This variable does not look relevant (minimum is close to 0).
  - If it's really irrelevant, minimum will move to 0 as 'n' goes to infinity.
  - But for finite 'n', minimum of parabola is unlikely to be exactly zero.

# Sparsity and L0-Regularization

- Consider 1D L0-regularized least squares objective:

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

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



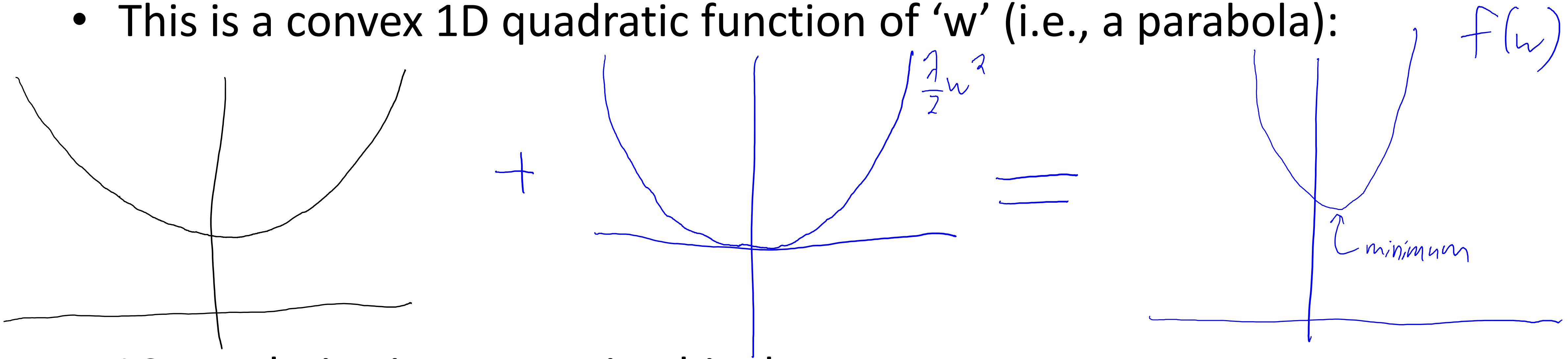

- L0-regularization minimum is often exactly at the 'discontinuity' at 0:
  - It sets the feature to exactly 0, removing it from the model.
  - But this is not a convex function.

# Sparsity and L2-Regularization

- Consider 1D L2-regularized least squares objective:

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

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



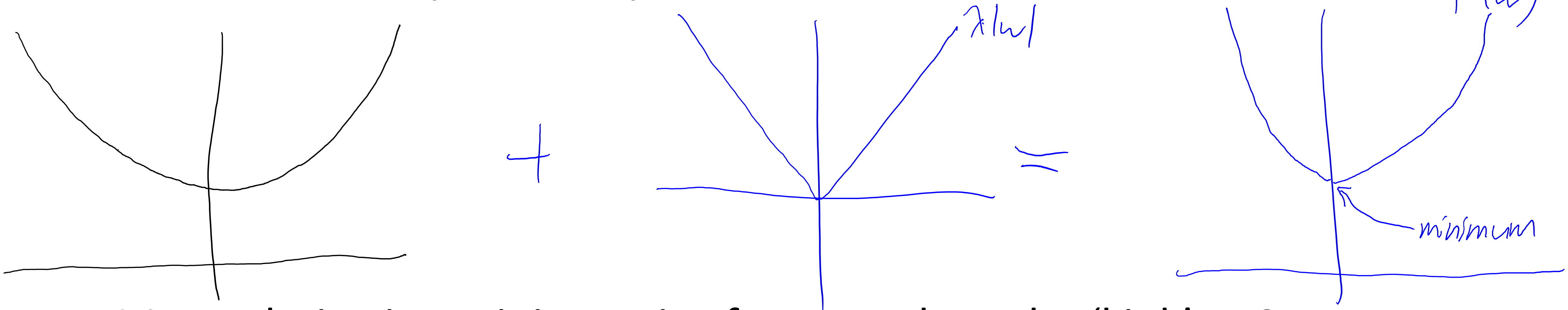

- L2-regularization moves it a bit closer to zero.
  - But there is nothing special about being 'exactly' zero.
  - L2-regularization will still tend to select this feature.

# Sparsity and L1-Regularization

- Consider 1D L1-regularized least squares objective:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w| = \begin{cases} \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda w, & w \geq 0 \\ \frac{1}{2}\sum_{i=1}^{n}(y_i - wx_i)^2 - \lambda w, & w < 0 \end{cases}$$

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




- L1-regularization minimum is often exactly at the ‘kink’ at 0:
  - It sets the feature to exactly 0, removing it from the model.
  - Big $\lambda$ means kink is ‘steep’. Small $\lambda$ makes 0 unlikely to be minimum.

# L2-Regularization vs. L1-Regularization

- L2-Regularization:
  - Insensitive to changes in data.
  - Significantly-decreased variance:
    - Lower test error.
  - Closed-form solution.
  - Solution is unique.
  - All ‘w’ tend to be non-zero.
  - Can learn with *linear* number of irrelevant features.
    - E.g., O(d) relevant features.
- L1-Regularization:
  - Insensitive to changes in data.
  - Significantly-decreased variance:
    - Lower test error.
  - Requires iterative solver.
  - Solution is not unique.
  - Many ‘w’ tend to be zero.
  - Can learn with **exponential** number of irrelevant features.
    - E.g., O(log(d)) relevant features.

# L1-Regularization Issues

- Advantages:
  - Deals with Taco Tuesday issue.
  - Takes into account effect size.
  - Convex (fast with specialized methods).
  - Performs regularization at the same time.
- Disadvantages:
  - Tends to give false positives (selects too many variables).
  - Only partially deals with milk-lactose issue:
    - Could pick one or both.

# Extensions of L1-Regularization

- "Elastic net":
  - Use L2-regularization *and* L1-regularization:
  - Nice properties of L1-regularization plus:
    - Solution is unique.
    - Addresses milk-lactose issue (selects both).
- "Bolasso":
  - Run L1-regularization on boostrap samples.
  - Take features that are non-zero in all samples.
  - Much less sensitive to false positives.
- There are *many* non-convex regularizers:
  - Less prone to false positives.
  - But computing global minimum is hard.

# Summary

- Norms are a way to measure 'size' or 'length' in higher dimensions.
- Feature selection is task of choosing the relevant features.
- Obvious approaches have systematic problems.
- Hypothesis testing: find set 'S' that makes yi and xij independent.
- Search and score: find features that optimize some score.
- L1-regularization: simultaneously regularize and select features.

- Next time:
  - Finding 'important' e-mails, and beating naïve Bayes on spam filtering.