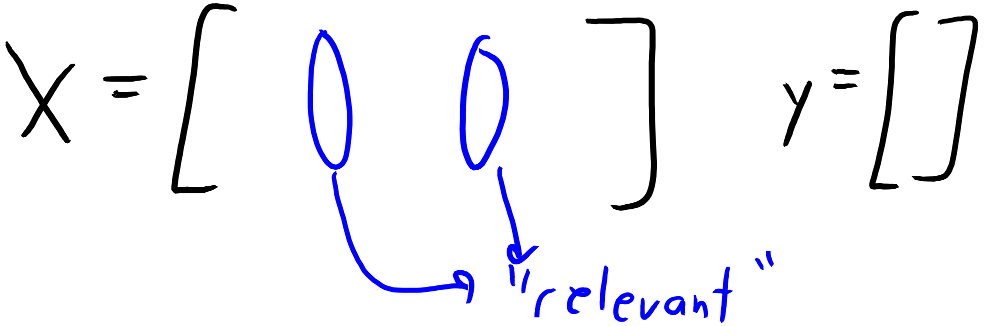
CPSC 340:
Machine Learning and Data Mining

Regularization

Fall 2022

Last Time: Feature Selection

- Last time we discussed feature selection:
  - Choosing set of "relevant" features.



- Most common approach is search and score:
  - Define "score" and "search" for features with best score.
- But it's hard to define the "score" and it's hard to "search".
  - So we often use greedy methods like forward selection.
- Methods work ok on "toy" data, but are frustrating on real data.
  - Different methods may return very different results.
  - Defining whether a feature is "relevant" is complicated and ambiguous.

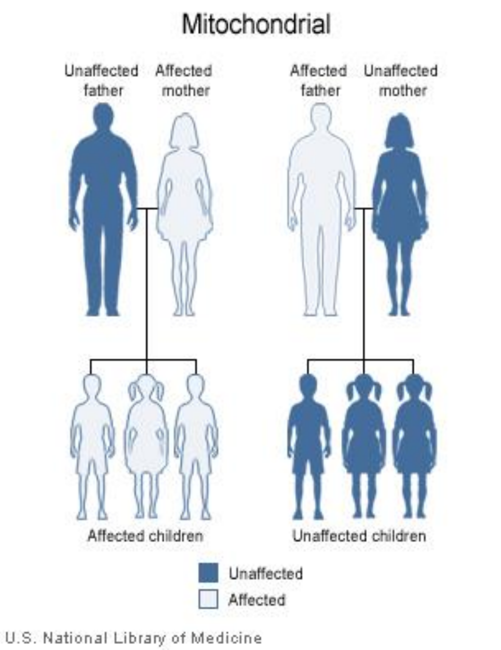
Last Time: Is "Relevance" Clearly Defined?

- Consider a supervised classification task:

gender	mom	dad
F	1	0
M	0	1
F	0	0
F	1	1

SNP
1
0
0
1

- True model:
  – (SNP = mom) with very high probability.
  – (SNP != mom) with some very low probability.
- What about (maternal) "grandma"?
  – Irrelevant since provides no extra information beyond "mom".
  – But relevant if you do not have the "mom" feature.



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

Is “Relevance” Clearly Defined?

- What if we don’t know “mom” or “grandma”?

gender	dad
F	0
M	1
F	0
F	1

SNP
1
0
0
1

- Now there are no relevant variables, right?
  - But “dad” and “mom” must have some common maternal ancestor.
  - “Mitochondrial Eve” estimated to be ~200,000 years ago.
- A “relevant” feature may have a tiny effect.

Is "Relevance" Clearly Defined?

- What if we don't know "mom" or "grandma"?

gender	dad
F	0
M	1
F	0
F	1

SNP
1
0
0
1

- Now there are no relevant variables, right?
  - What if "mom" likes "dad" because he has the same SNP as her?

- Confounding factors can change "relevance" of variables.

Is "Relevance" Clearly Defined?

- What if we add "sibling"?

gender	dad	sibling
F	0	1
M	1	0
F	0	0
F	1	1

SNP
1
0
0
1

- Sibling is "relevant" for predicting SNP, but it is not the cause.

- "Relevance" for prediction does not imply a causal relationship.
  – Causality can even be reversed…

Is “Relevance” Clearly Defined?

- What if don’t have “mom” but we have “baby”?

gender	dad	baby
F	0	1
M	1	1
F	0	0
F	1	1

SNP
1
0
0
1

- “Baby” is relevant when (gender == F).
  - “Baby” is relevant (though causality is reversed).
  - Is “gender” relevant?
    - If we want to find relevant causal factors, “gender” is not relevant.
    - If we want to predict SNP, “gender” is relevant.
- “Relevance” may depend on values of certain features.
  - “Context-specific” relevance.

Is “Relevance” Clearly Defined?

- Warnings about feature selection:
  - If features can be predicted from features, you can’t know which to pick.
  - A feature is only “relevant” in the context of available features.
  - A “relevant” feature may have a tiny effect.
  - Confounding factors can change whether features are relevant.
  - “Relevance” for prediction does not imply a causal relationship.
  - “Relevance” may be conditional on values of certain features.

Is this hopeless?

- We often want to do feature selection we so have to try!
- Different methods are affected by problems in different ways.
- These “problems” don’t have right answers but have wrong answers:
  - Variable dependence (“mom” and “mom2” have same information).
    - But should take at least one.
  - Conditional independence (all “grandma” information is captured by “mom”).
    - Should take “grandma” only if “mom” missing.
- These “problems” have application-specific answers:
  - Tiny effects.
  - Context-specific relevance (is “gender” relevant if given “baby”?).
- See bonus slides for discussion of causality and confounding issues.
  - Unless you control data collection, standard feature selection methods cannot address those issues.

My advice if you want the "relevant" variables.

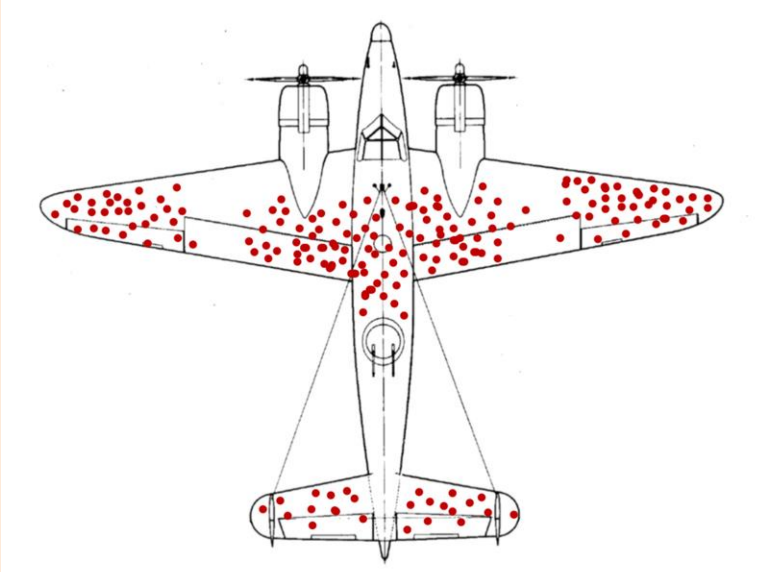
- Try the association approach.
- Try forward selection with different values of λ.
- Try out a few other feature selection methods too.

- Discuss the results with the domain expert.
  - They probably have an idea of why some variables might be relevant.

- Do not be overconfident:
  - These methods are probably not discovering how the world truly works.
  - "The algorithm has found that these variables are helpful in predicting y_i."
    - Then a warning that these models are not perfect at finding relevant variables.

Related: Survivorship Bias

- Plotting location of bullet holes on planes returning from WW2:

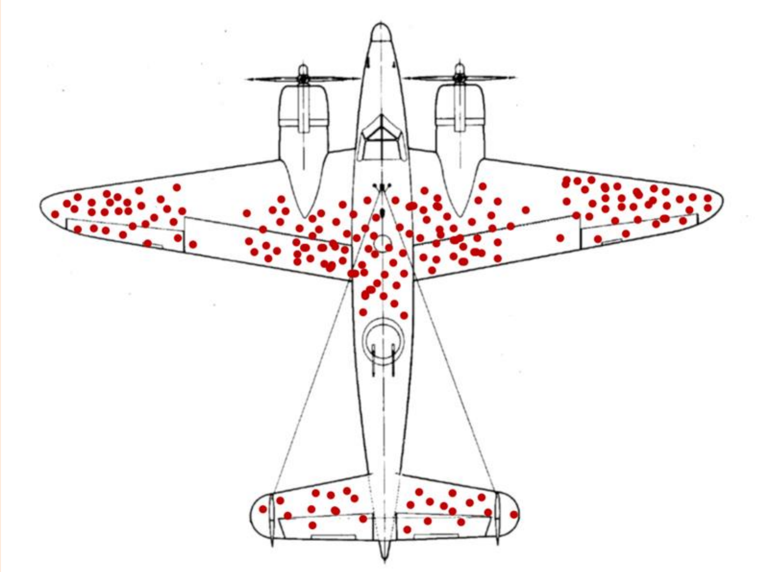


- Where are the “relevant” parts of the plane to protect?
  - “Relevant” parts are actually where there are no bullets.
  - Planes shot in other places did not come back (armor was needed).

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

Related: Survivorship Bias

- Plotting location of bullet holes on planes returning from WW2:

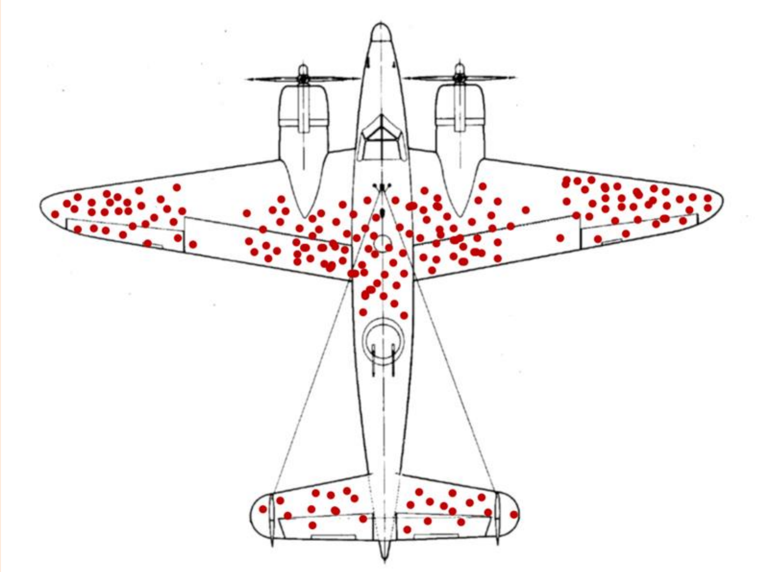


- This is an example of "survivorship bias":
  - Data is not IID because you only sample the "survivors".
  - Causes havoc for feature selection, and ML methods in general.

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

Related: Survivorship Bias

- Plotting location of bullet holes on planes returning from WW2:



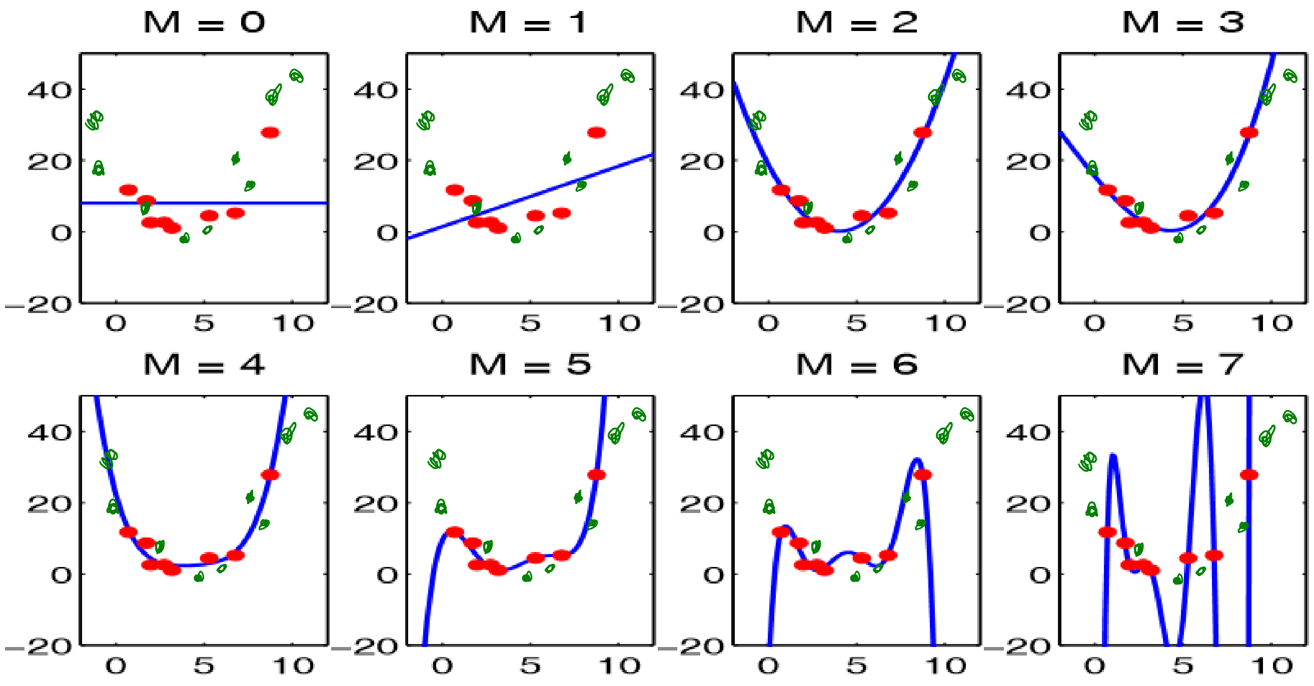
- People come to wrong conclusions due to survivor bias all the time.
  - Article on “secrets of success”, focusing on traits of successful people.
    - But ignoring the number of non-super-successful people with the same traits.
  - Article hypothesizing about various topics (allergies, mental illness, etc.).

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

Next Topic: Regularization

Recall: Polynomial Degree and Training vs. Testing

- We've said that complicated models tend to overfit more.



- But what if we need a complicated model?

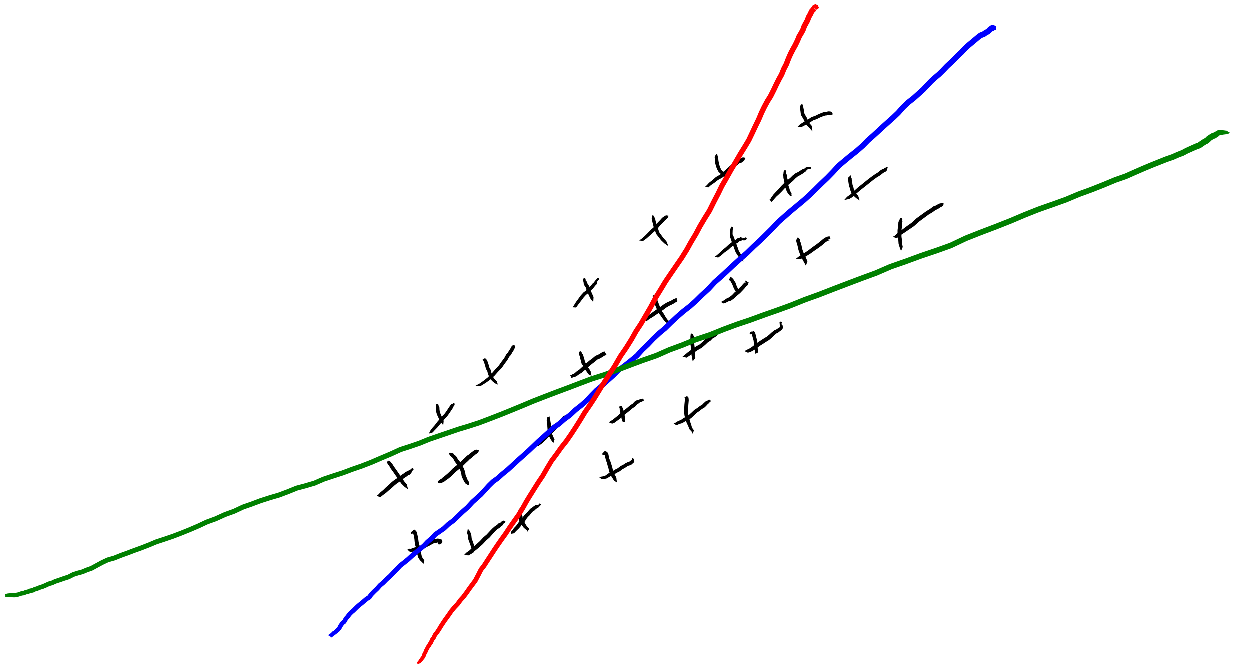
Controlling Complexity

- Usually "true" mapping from x_i to y_i is complex.
  - Might need high-degree polynomial.
  - Might need to combine many features, and do not know "relevant" ones.
- But complex models can overfit.
- So what do we do???

- Our main tools:
  - Model averaging: average over multiple models to decrease variance.
  - Regularization: add a penalty on the complexity of the model.

Would you rather?

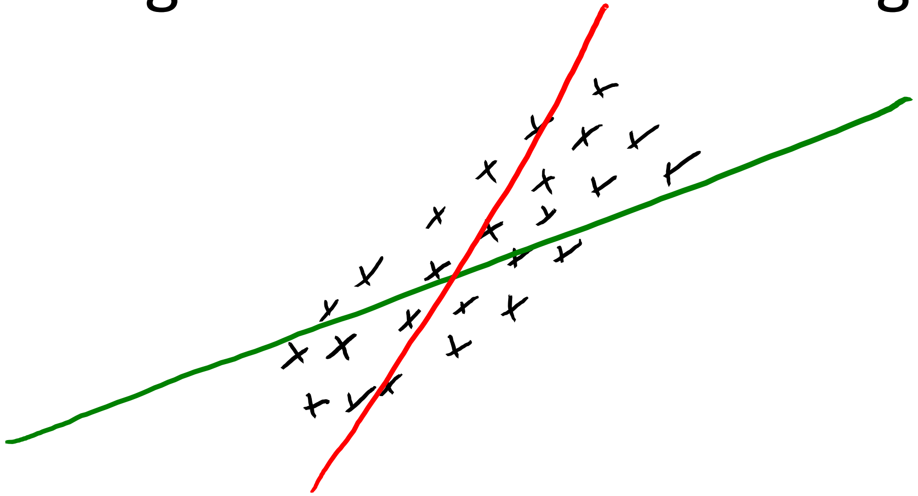
- Consider the following dataset and 3 linear regression models:



- Which line should we choose?

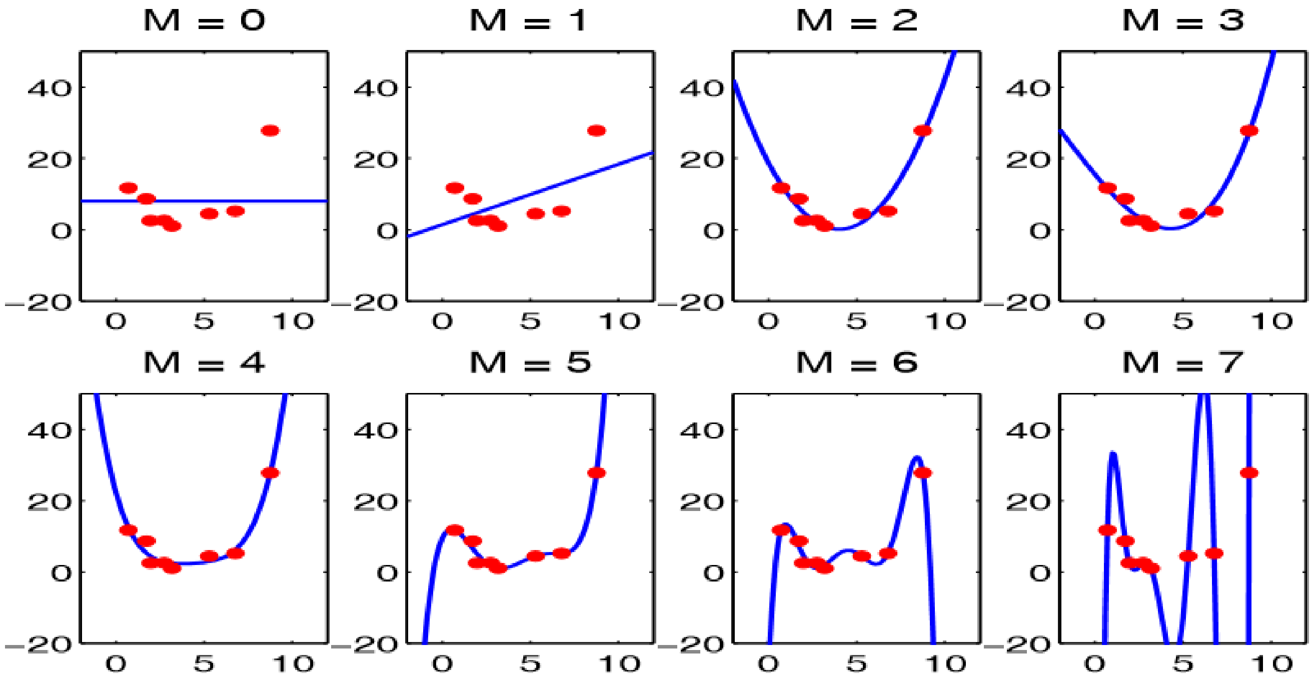
Would you rather?

- Consider the following dataset and 3 linear regression models:



- What if you are forced to choose between red and green?
  - And assume they have the same training error.
- You should pick green.
  - Since slope is smaller, small change in x_i has a smaller change in prediction y_i.
    - Green line's predictions are less sensitive to having 'w' exactly right.
  - Since green 'w' is less sensitive to data, test error might be lower.

Size of Regression Weights and Overfitting



- The regression weights w_j with degree-7 are huge in this example.
- The degree-7 polynomial would be less sensitive to the data,

  if we "regularized" the w_j so that they are small.

$\hat{y}_i = 0.0001(x_i)^7 + 0.03(x_i)^3 + 3$ vs. $\hat{y}_i = 1000(x_i)^7 - 500(x_i)^6 + 890x_i$

L2-Regularization

- Standard regularization strategy is L2-regularization:

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{d} w_j^2 \quad \text{or} \quad f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2$$

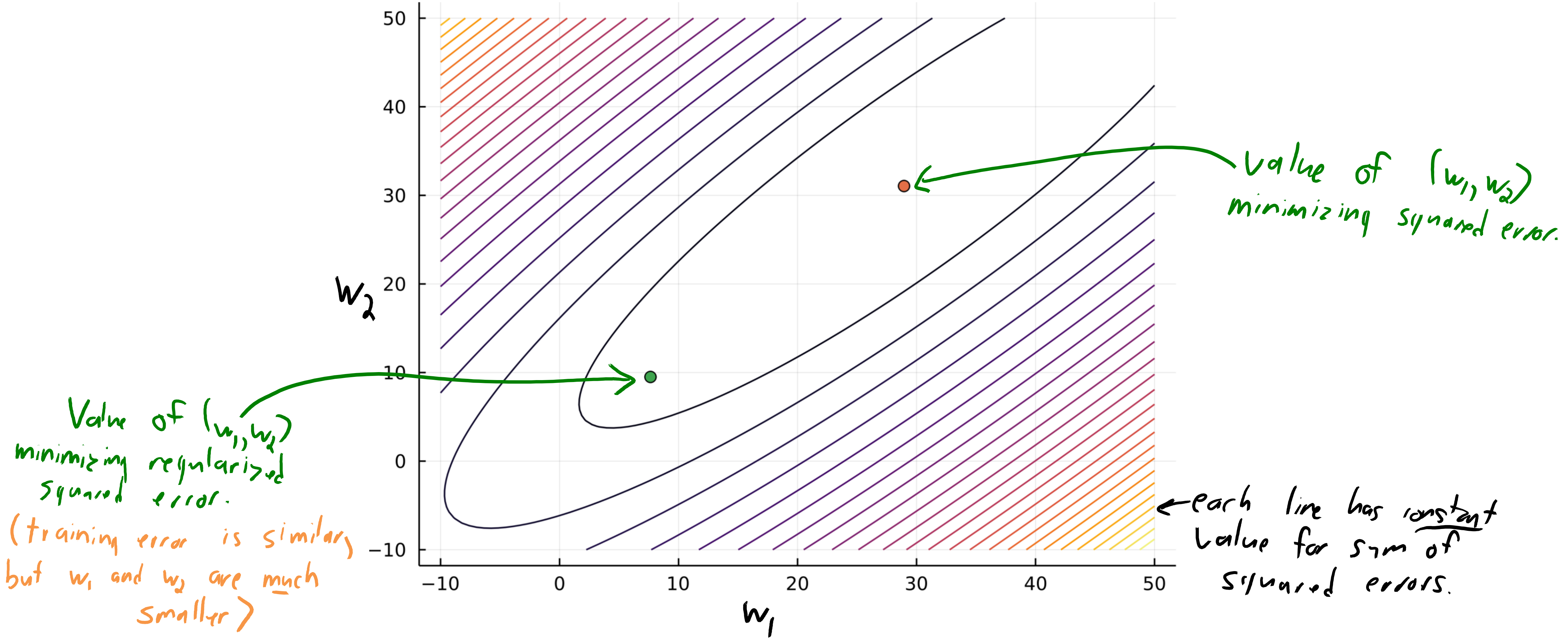
  - For some regularization parameter $\lambda > 0$.
- Intuition: large slopes w_j tend to lead to overfitting.

- Objective balances getting low error vs. having small slopes 'w_j'.
  - "You can increase the training error if it makes 'w' much smaller."
  - Nearly-always reduces overfitting.

- In terms of fundamental trade-off:
  - Regularization increases training error.
  - Regularization decreases approximation error.

L2-Regularization

- Visualizing squared error as a function of parameters (d=2):



L2-Regularization

- L2-regularized least squares:

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

  - Regularization parameter λ > 0 controls "strength" of regularization.
    - Large λ puts large penalty on slopes (worse training error, better approximation).

- How should you choose λ?
  - Theory: as 'n' grows λ should be in the range O(1) to (√n).
  - Practice: optimize validation set or cross-validation error.
    - This almost always decreases the test error.

L2-Regularization “Shrinking” Example

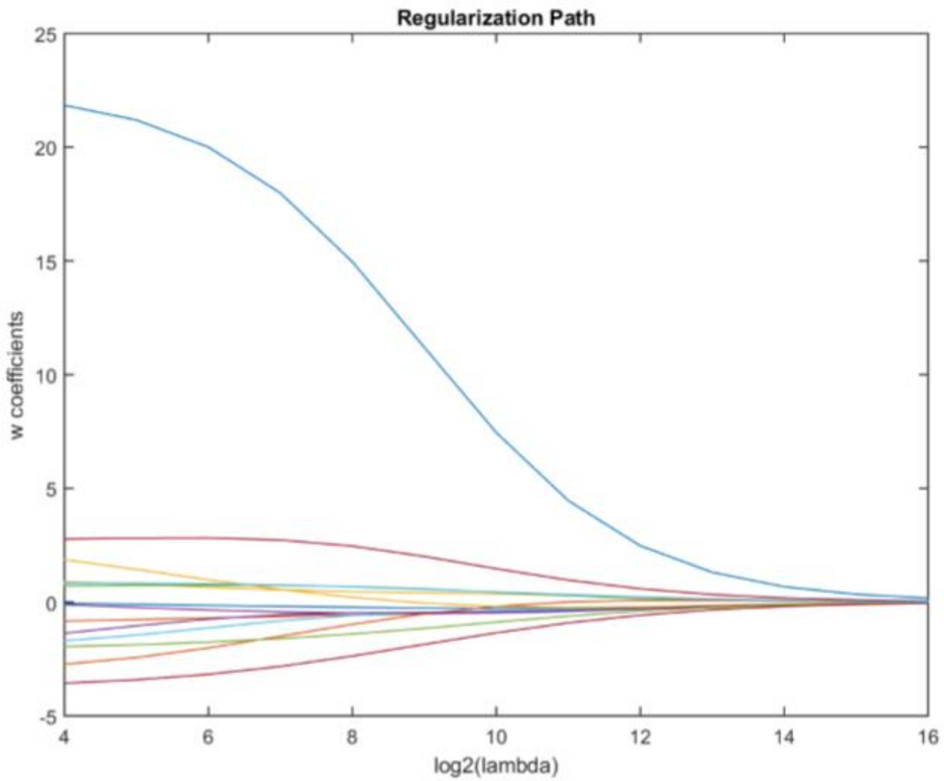
- Solution to a “least squares with L2-regularization” for different λ:

λ	w_1	w_2	w_3	w_4	w_5	$\|\|Xw - y\|\|^2$	$\|\|w\|\|^2$
0	-1.88	1.29	-2.63	1.78	-0.63	285.64	15.68
1	-1.88	1.28	-2.62	1.78	-0.64	285.64	15.62
4	-1.87	1.28	-2.59	1.77	-0.66	285.64	15.43
16	-1.84	1.27	-2.50	1.73	-0.73	285.71	14.76
64	-1.74	1.23	-2.22	1.59	-0.90	286.47	12.77
256	-1.43	1.08	-1.70	1.18	-1.05	292.60	8.60
1024	-0.87	0.73	-1.03	0.57	-0.81	321.29	3.33
4096	-0.35	0.31	-0.42	0.18	-0.36	374.27	0.56

- We get least squares with $\lambda = 0$.
  - But we can achieve similar training error with smaller $||w||$.
- $||Xw - y||$ increases with λ, and $||w||$ decreases with λ.
  - Though individual w_j can increase or decrease with lambda.
  - Because we use the L2-norm, the large ones decrease the most.

Regularization Path

- Regularization path is a plot of the optimal weights 'w_j' as 'λ' varies:



- Starts with least squares with $\lambda = 0$, and w_j converge to 0 as λ grows.

Solving L2-Regularized Least Squares Problem

- Solving for $\nabla f(w)=0$ to compute L2-regularized least squares:
  - Objective:
    $$f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2$$
    $$= \frac{1}{2}w^TX^TXw - w^TX^Ty + \frac{1}{2}y^Ty + \frac{\lambda}{2}w^Tw \quad \text{(expand)}$$
  - Gradient:
    $$\nabla f(w) = X^TXw - X^Ty + 0 + \lambda w$$
  - Setting gradient equal to zero vector:
    $$X^TXw - X^Ty + \lambda w = 0$$
    $$X^TXw + \lambda w = X^Ty \quad \text{(move terms with no 'w' to right)}$$
  - Factorize 'w' on the left side (identity matrix makes dimensions match):
    $$(X^TX + \lambda I)w = X^Ty$$
    $$w = \underbrace{(X^TX + \lambda I)^{-1}}_{\text{you can show that this matrix is always invertible.}} X^Ty$$

Gradient Descent for L2-Regularized Least Squares

- The L2-regularized least squares objective and gradient:

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

- Gradient descent iterations for L2-regularized least squares:

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

- Cost of gradient descent iteration is still O(nd).
  - Can show number of iterations decrease as λ increases (not obvious).

Why use L2-Regularization?

- It's a weird thing to do, but Mark says "always use regularization".
  - "Almost always decreases test error" should already convince you.

- But here are 6 more reasons:
  1. Solution 'w' is unique.
  2. $X^T X$ does not need to be invertible (no collinearity issues).
  3. Less sensitive to changes in X or y.
  4. Gradient descent converges faster (bigger λ means fewer iterations).
  5. Stein's paradox: if $d \geq 3$, 'shrinking' moves us closer to 'true' w.
  6. Worst case: just set λ small and get the same performance.

Next Topic: Standardizing Features

Features with Different Scales

- Consider continuous features with different scales:

Egg (#)	Milk (mL)	Fish (g)	Pasta (cups)
0	250	0	1
1	250	200	1
0	0	0	0.5
2	250	150	0

- Should we convert to some standard 'unit'?
  - It doesn't matter for decision trees or naïve Bayes.
    - They only look at one feature at a time.
  - It does not matter for least squares:
    - w_j*(100 mL) gives the same model as w_j*(0.1 L) with a different w_j.

Features with Different Scales

- Consider continuous features with different scales:

Egg (#)	Milk (mL)	Fish (g)	Pasta (cups)
0	250	0	1
1	250	200	1
0	0	0	0.5
2	250	150	0

- Should we convert to some standard 'unit'?
  - It matters for k-nearest neighbours:
    - "Distance" will be affected more by large features than small features.
  - It matters for regularized least squares:
    - Penalizing $(w_j)^2$ means different things if features 'j' are on different scales.

Standardizing Features

$X = \begin{bmatrix} & \bigcirc & \end{bmatrix}$ average of column 'j'

- It is common to standardize continuous features:
  - For each feature:
    1. Compute mean and standard deviation:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \qquad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}$$

    2. Subtract mean and divide by standard deviation ("z-score")

Replace x_{ij} with $\frac{x_{ij}-\mu_j}{\sigma_j}$

  - Now measures "standard deviations from mean".
    - And changes in 'w_j' have similar effect for any feature 'j'
- How should we standardize test data?
  - Wrong approach: use mean and standard deviation of test data.
    - Training and test mean and standard deviation might be very different.
  - Right approach: use mean and standard deviation of training data.

Standardizing Features

$X = \begin{bmatrix} & 0 & \end{bmatrix}$

average of column 'j'

- It is common to standardize continuous features:
  - For each feature:
    1. Compute mean and standard deviation:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \qquad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2}$$

    2. Subtract mean and divide by standard deviation ("z-score")

Replace x_{ij} with $\frac{x_{ij}-\mu_j}{\sigma_j}$

  - Now measures "standard deviations from mean".
    - And changes in 'w_j' have similar effect for any feature 'j'
- If we're doing 10-fold cross-validation:
  - Compute μ_j and σ_j based on the 9 training folds (e.g., average over 9/10s of data).
    - Standardize the remaining ("validation") fold with this "training" μ_j and σ_j.
  - Re-standardize for different folds (violate golden rule if standardize before split).

Standardizing Target

- In regression, we sometimes standardize the targets y_i.
  - Puts targets on the same standard scale as standardized features:

Replace y_i with $\frac{y_i - \mu_y}{\sigma_y}$

- With standardized target, setting w = 0 predicts average y_i:
  - High regularization makes us predict closer to the average value.
- Again, make sure you standardize test data with the training stats.
  - And do not forget to "un-standardize" predictions to get back to original space.
- Other common transformations of y_i are logarithm/exponent:

Use $\log(y_i)$ or $\exp(\gamma y_i)$

  - Makes sense for geometric/exponential processes.

Regularizing the y-Intercept?

- Should we regularize the y-intercept?

- No! Why encourage it to be closer to zero? (It could be anywhere.)
  - You should be allowed to shift function up/down globally.

- Yes! It makes the solution unique and it easier to compute ‘w’.

- Compromise: regularize by a smaller amount than other variables.

$$f(w, w_0) = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i + w_0 - y_i)^2 + \frac{\lambda}{2}\|w\|^2 + \frac{\lambda_0}{2} w_0^2$$

Summary

- “Relevance” is really hard to define.
  - Post-lecture bonus: “rough guide” to how different methods deal with this issue.
- Regularization:
  - Adding a penalty on model complexity.
- L2-regularization: penalty on L2-norm of regression weights ‘w’.
  - Trades training error against size of weights, almost always improves test error.
- Standardizing features:
  - For some models it makes sense to have features on the same scale.

- Next time: learning with an exponential number of irrelevant features.

Rough Guide to Feature Selection

Method\Issue	Dependence	Conditional Independence	Tiny effects	Context-Specific Relevance
Association (e.g., measure correlation between features 'j' and 'y')	Ok (takes "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Ignores	Bad (misses features that must interact, "gender" irrelevant given "baby")

Rough Guide to Feature Selection

Method\Issue	**Dependence**	**Conditional Independence**	**Tiny effects**	**Context-Specific Relevance**
Association (e.g., measure correlation between features 'j' and 'y')	Ok (takes "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Ignores	Bad (misses features that must interact, "gender" irrelevant given "baby")
Regression Weight (fit least squares, take biggest $\|w_j\|$)	Bad (can take irrelevant but collinear, can take none of "mom1-3")	Ok (takes "mom" not "grandma", if linear and 'n' large.	Ignores (unless collinear)	Ok (if linear, "gender" relevant give "baby")

Rough Guide to Feature Selection

Method\Issue	Dependence	Conditional Independence	Tiny effects	Context-Specific Relevance
Association (e.g., measure correlation between features 'j' and 'y')	Ok (takes "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Ignores	Bad (misses features that must interact, "gender" irrelevant given "baby")
Regression Weight (fit least squares, take biggest $\|w_j\|$)	Bad (can take irrelevant but collinear, can take none of "mom1-3")	Ok (takes "mom" not "grandma", if linear and 'n' large.	Ignores (unless collinear)	Ok (if linear, "gender" relevant give "baby")
Search and Score w/ Validation Error	Ok (takes at least one of "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Allows	Ok ("gender" relevant given "baby")

Rough Guide to Feature Selection

Method\Issue	Dependence	Conditional Independence	Tiny effects	Context-Specific Relevance
Association (e.g., measure correlation between features 'j' and 'y')	Ok (takes "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Ignores	Bad (misses features that must interact, "gender" irrelevant given "baby")
Regression Weight (fit least squares, take biggest $\|w_j\|$)	Bad (can take irrelevant but collinear, can take none of "mom1-3")	Ok (takes "mom" not "grandma", if linear and 'n' large.	Ignores (unless collinear)	Ok (if linear, "gender" relevant give "baby")
Search and Score w/ Validation Error	Ok (takes at least one of "mom" and "mom2")	Bad (takes "grandma", "great-grandma", etc.)	Allows (many false positives)	Ok ("gender" relevant given "baby")
Search and Score w/ L0-norm	Ok (takes exactly one of "mom" and "mom2")	Ok (takes "mom" not grandma if linear-ish).	Ignores (even if collinear)	Ok ("gender" relevant given "baby")

Alternative to Search and Score: good old p-values

- Hypothesis testing ("constraint-based") approach:
  - Generalization of the "association" approach to feature selection.
  - Performs a sequence of conditional independence tests.

$$x_{ij} \perp y_i \mid x_{is}$$

feature 'j' label other features 's'

"If I know features in 's' does feature 'j' tell me anything about label?"

  - If they are independent (like "p < .05"), say that 'j' is "irrelevant".
- Common way to do the tests:
  - "Partial" correlation (numerical data).
  - "Conditional" mutual information (discrete data).

Testing-Based Feature Selection

- Hypothesis testing ("constraint-based") approach:
- Two many possible tests, "greedy" method is for each 'j' do:

First test if $x_{ij} \perp y_i$

If still dependent test $x_{ij} \perp y_i \mid x_{is}$ where 's' has one feature

If still dependent test $x_{ij} \perp y_i \mid x_{is}$ where 's' now has two features

Often choose features to minimize dependence.

⋮

If still dependent when 's' includes all other features, declare 'j' relevant.

- "Association approach" is the greedy method where you only do the first test (subsequent tests remove a lot of false positives).

Hypothesis-Based Feature Selection

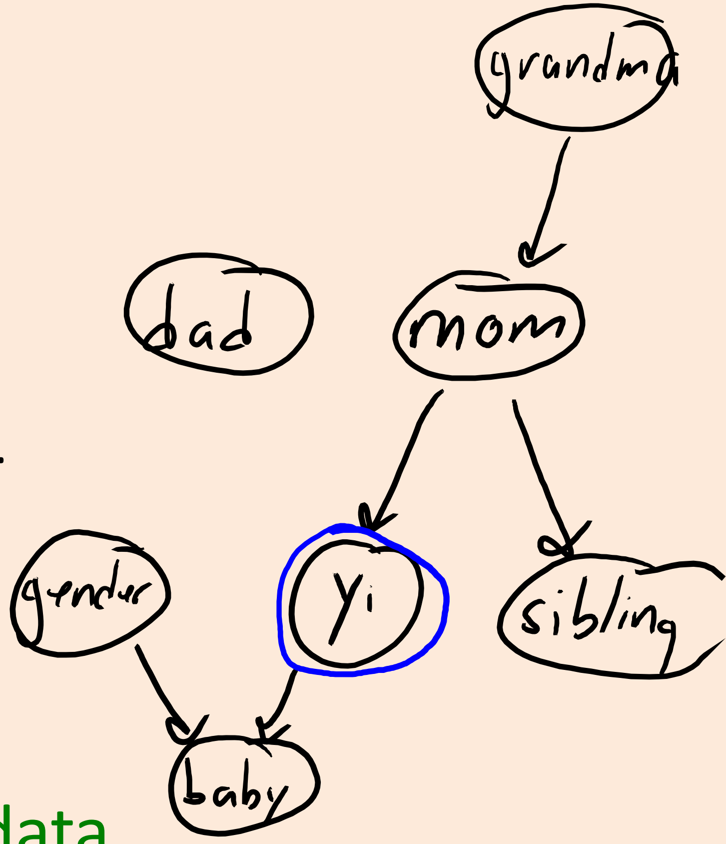
- Advantages:
  - Deals with conditional independence.
  - Algorithm can explain why it thinks 'j' is irrelevant.
  - Doesn't necessarily need linearity.
- Disadvantages:
  - Deals badly with exact dependence: doesn't select "mom" or "mom2" if both present.
  - Usual warning about testing multiple hypotheses:
    - If you test $p < 0.05$ more than 20 times, you're going to make errors.
  - Greedy approach may be sub-optimal.
- Neither good nor bad:
  - Allows tiny effects.
  - Says "gender" is irrelevant when you know "baby".
  - This approach is sometimes better for finding relevant factors, not to select features for learning.

Causality

- None of these approaches address causality or confounding:
  - "Mom" is the only relevant direct causal factor.
  - "Dad" is really irrelevant.
  - "Grandma" is causal but is irrelevant if we know "mom".

- Other factors can help prediction but aren't causal:
  - "Sibling" is predictive due to confounding of effect of same "mom".
  - "Baby" is predictive due to reverse causality.
  - "Gender" is predictive due to common effect on "baby".

- We can sometimes address this using interventional data…



Interventional Data

- The difference between observational and interventional data:
  - If I see that my watch says 10:45, class is almost over (observational).
  - If I set my watch to say 10:45, it doesn't help (interventional).

- The intervention can help discover causal effects:
  - "Watch" is only predictive of "time" in observational setting (so not causal).

- General idea for identifying causal effects:
  - "Force" the variable to take a certain value, then measure the effect.
    - If the dependency remains, there is a causal effect.
    - We "break" connections from reverse causality, common effects, or confounding.

Causality and Dataset Collection

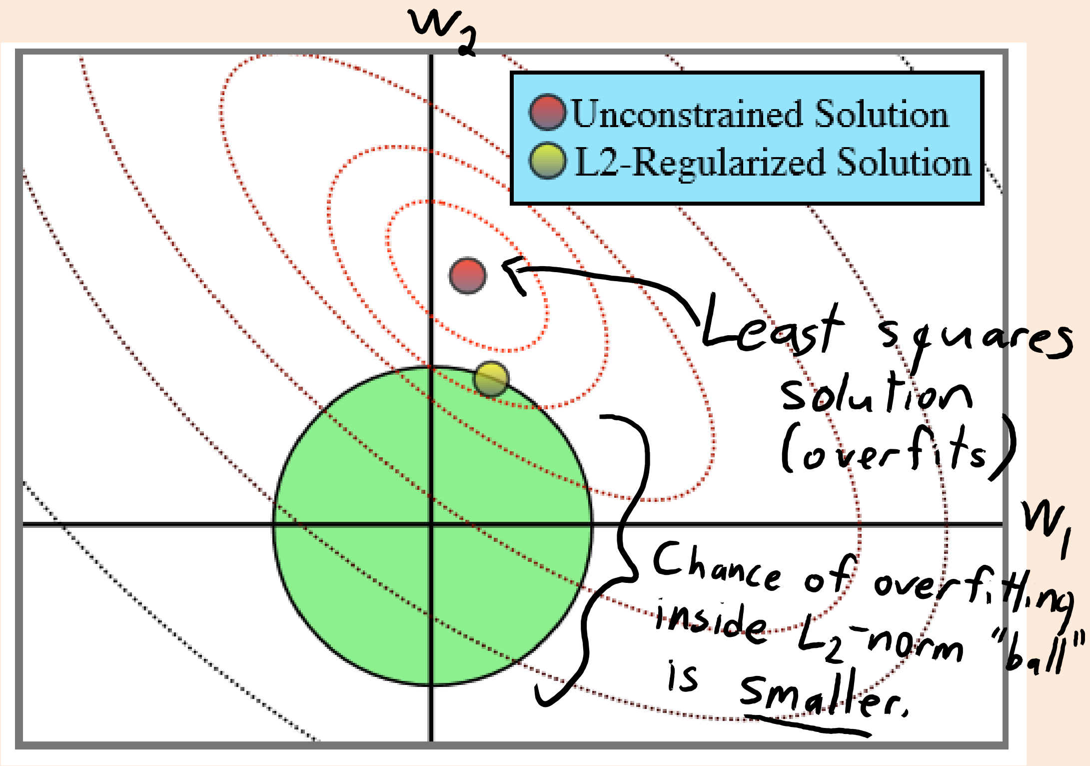
- This has to do with the way you collect data:
  - You can't "look" for variables taking the value "after the fact".
  - You need to manipulate the value of the variable, then watch for changes.
- This is the basis for randomized control trial in medicine:
  - Randomly assigning pills "forces" value of "treatment" variable.
    - Randomization means they aren't taking the pill due to confounding factors.
    - Differences between people who did and did not take pill should be caused by pill.
  - Include a "control" as a value to prevent placebo effect as confounding.
- See also Simpson's Paradox:
  - https://www.youtube.com/watch?v=ebEkn-BiW5k

L2-Regularization

- Standard regularization strategy is L2-regularization:

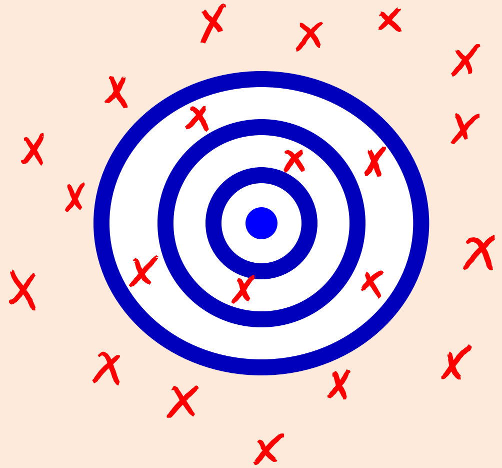
$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w^Tx_i - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{d} w_j^2 \quad \text{or} \quad f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2$$

- Equivalent to minimizing squared error but keeping L2-norm small.



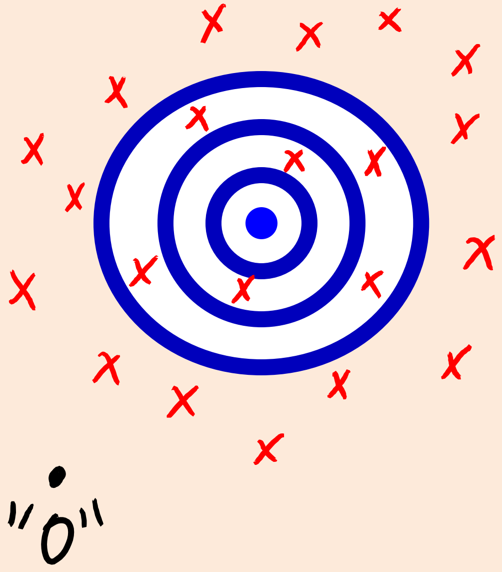
Regularization/Shrinking Paradox

- We throw darts at a target:
  - Assume we don't always hit the exact center.
  - Assume the darts follow a symmetric pattern around center.



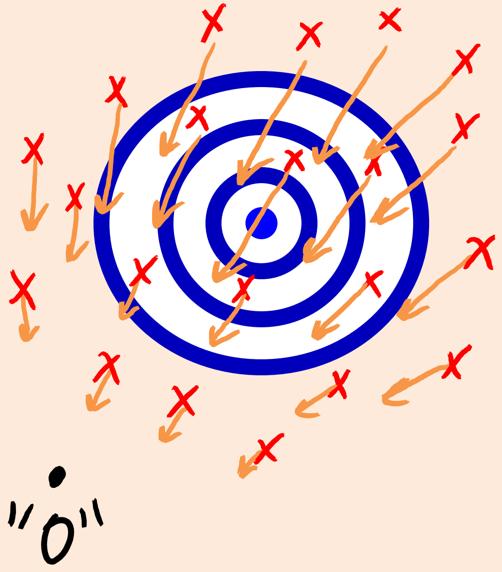
Regularization/Shrinking Paradox

- We throw darts at a target:
  - Assume we don't always hit the exact center.
  - Assume the darts follow a symmetric pattern around center.
- Shrinkage of the darts :
  1. Choose some arbitrary location '0'.
  2. Measure distances from darts to '0'.



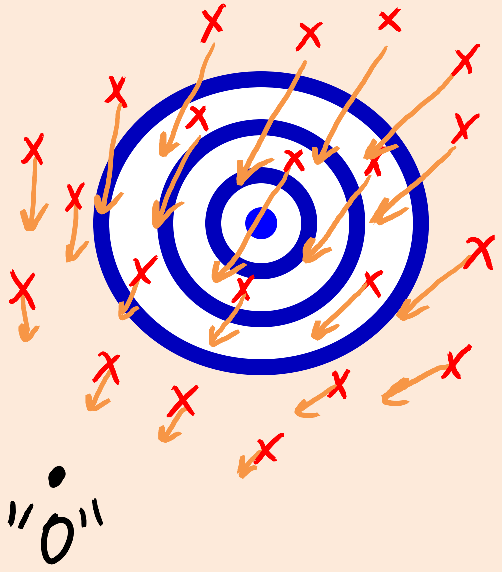
Regularization/Shrinking Paradox

- We throw darts at a target:
  - Assume we don't always hit the exact center.
  - Assume the darts follow a symmetric pattern around center.
- Shrinkage of the darts :
  1. Choose some arbitrary location '0'.
  2. Measure distances from darts to '0'.
  3. Move misses towards '0', by *small* amount proportional to distance from 0.
- If small enough, darts will be closer to center on average.



Regularization/Shrinking Paradox

- We throw darts at a target:
  - Assume we don't always hit the exact center.
  - Assume the darts follow a symmetric pattern around center.
- Shrinkage of the darts :
  1. Choose some arbitrary location '0'.
  2. Measure distances from darts to '0'.
  3. Move misses towards '0', by *small* amount proportional to distance from 0.
- If small enough, darts will be closer to center on average.



Visualization of the related higher-dimensional paradox that the mean of data coming from a Gaussian is not the best estimate of the mean of the Gaussian in 3-dimensions or higher: https://www.naftaliharris.com/blog/steinviz