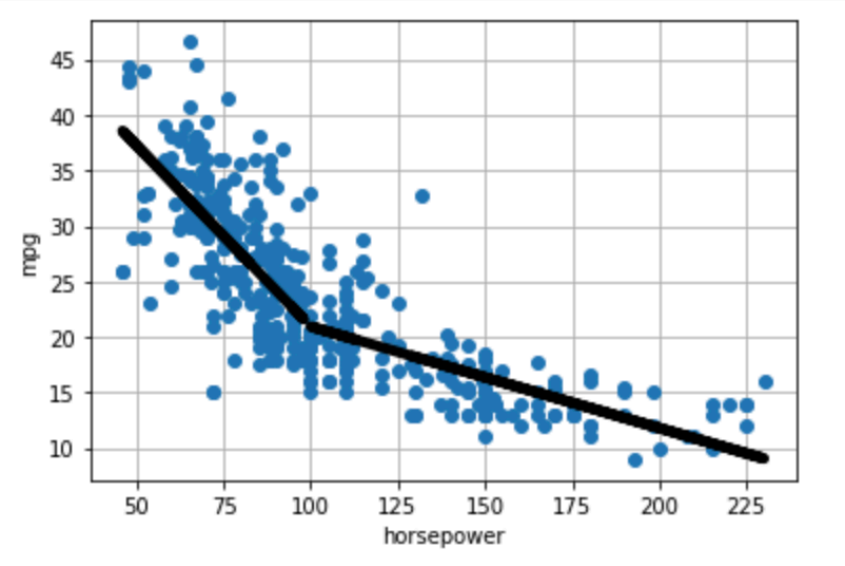
CS-UY 4563: Lecture 5
Model Selection and Regularization

NYU Tandon School of Engineering, Prof. Christopher Musco

COURSE ADMIN

- Multiple linear regression lab due **tomorrow night**.
- Second written homework posted due **next Tuesday 2/18.**

Practice with gradients, function transformations, reduction from piecewise regression to multiple linear regression.



COURSE ADMIN

- TA office hours moved to **11am - 1pm in 219 Rogers Hall** – this will be their permanent location.
- I won't have office hours this week.

LOSS MINIMIZATION

Basic machine learning problem:

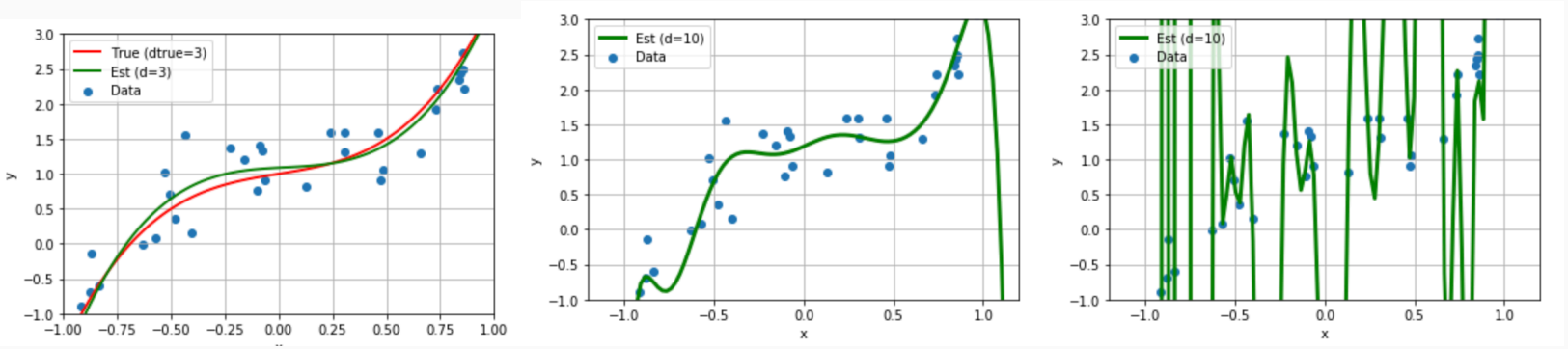
- Given model $f_{\boldsymbol{\theta}}$ and loss function $L_{\text{train}}(f_{\boldsymbol{\theta}})$.
- Choose $\boldsymbol{\theta}^*$ to minimize $L_{\text{train}}(f_{\boldsymbol{\theta}})$.

Model selection problem:

- Given choice of many models $f^{(1)}_{\boldsymbol{\theta}_1}, f^{(2)}_{\boldsymbol{\theta}_2}, \ldots, f^{(q)}_{\boldsymbol{\theta}_q}$.
- Choose $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_q^*$ to minimize $L_{\text{train}}(f_{\boldsymbol{\theta}_1}), \ldots, L_{\text{train}}(f_{\boldsymbol{\theta}_q})$.
- Need to choose the "best" model for our data.

MODEL SELECTION EXAMPLE

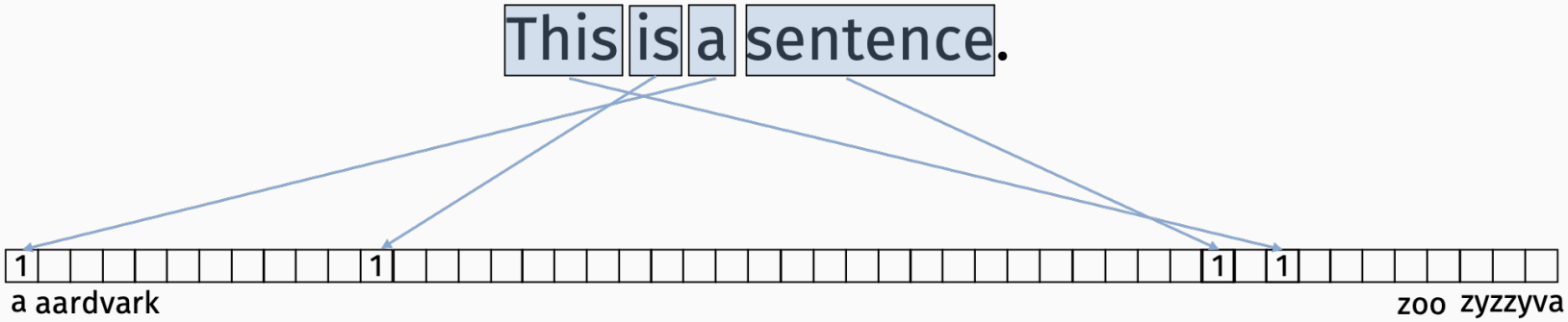
Polynomial regression models with different degree. See `demo_polyfit.ipynb`.



- Model $f^{(1)}_{\boldsymbol{\theta}_1}$: all linear functions.
- Model $f^{(2)}_{\boldsymbol{\theta}_2}$: all quadratic functions.
- Model $f^{(3)}_{\boldsymbol{\theta}_3}$: all cubic functions.
- ...

bag-of-words models and **n-grams**

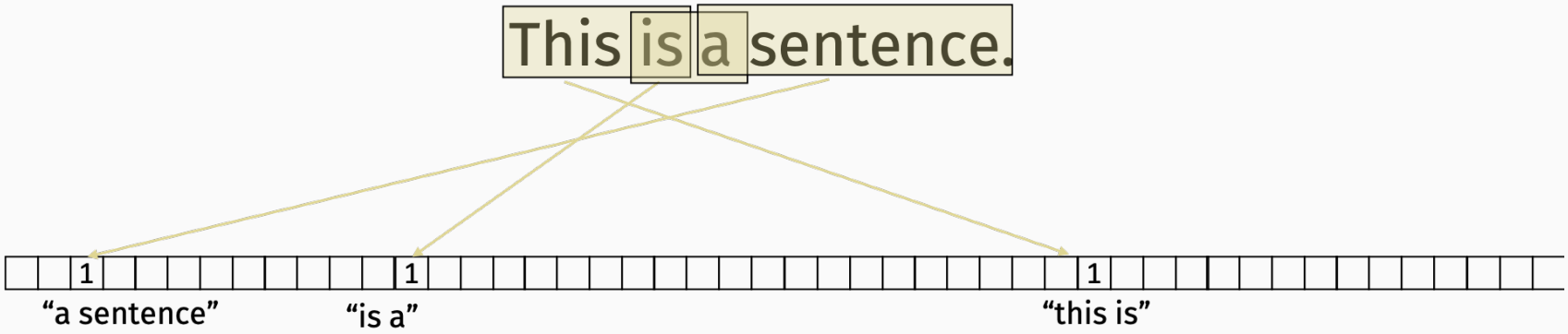
Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.



bag-of-words

bag-of-words models and **n-grams**

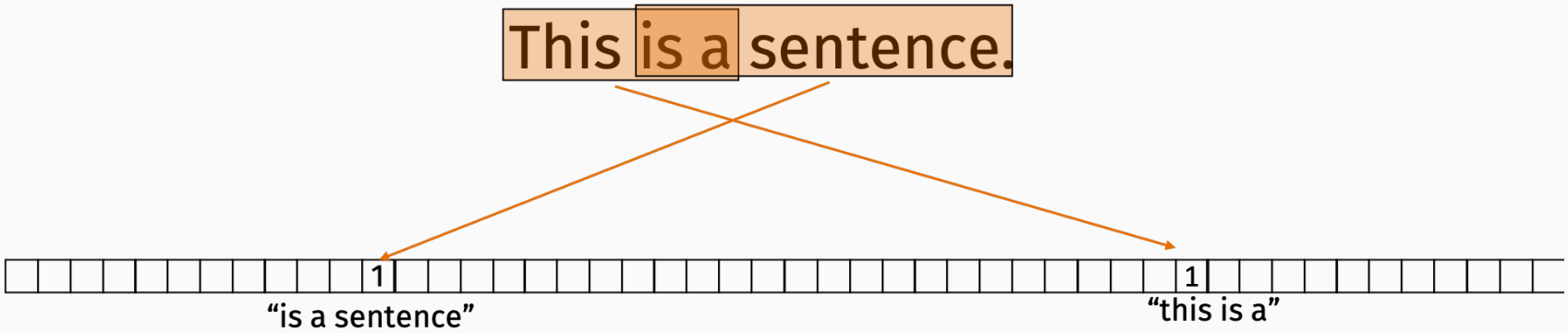
Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.



bi-grams

bag-of-words models and **n-grams**

Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.



tri-grams

MODEL SELECTION EXAMPLE

Models of increasing order:

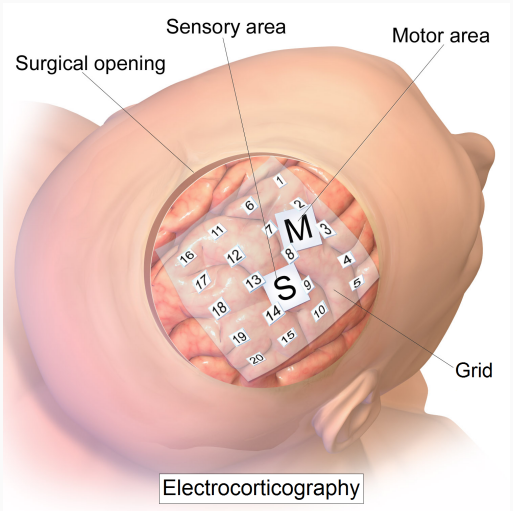
- Model $f^{(1)}_{\boldsymbol{\theta}_1}$: spam filter that looks at **single words**.
- Model $f^{(2)}_{\boldsymbol{\theta}_2}$: spam filter that looks at **bi-grams**.
- Model $f^{(3)}_{\boldsymbol{\theta}_3}$: spam filter that looks at **tri-grams**.
- ...

"interest" "low interest" "low interest loan"

Increased length of **n-gram** means more expressive power.

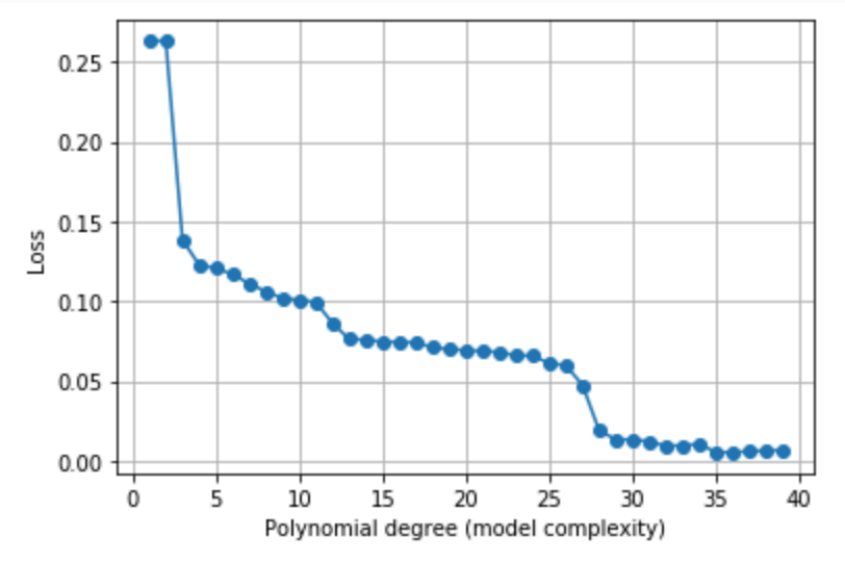
Electrocorticography ECoG (upcoming lab or demo):

- Implant grid of electrodes on surface of the brain to measure electrical activity in different regions.



- Predict hand motion based on ECoG measurements.
- **Model order:** predict movement at time t using brain signals at time $t, t-1, \ldots, t-q$ for varying values of q.

The more **complex** our model class the better our loss:



So training loss alone is not usually a good metric for model selection. Small loss does not imply generalization.

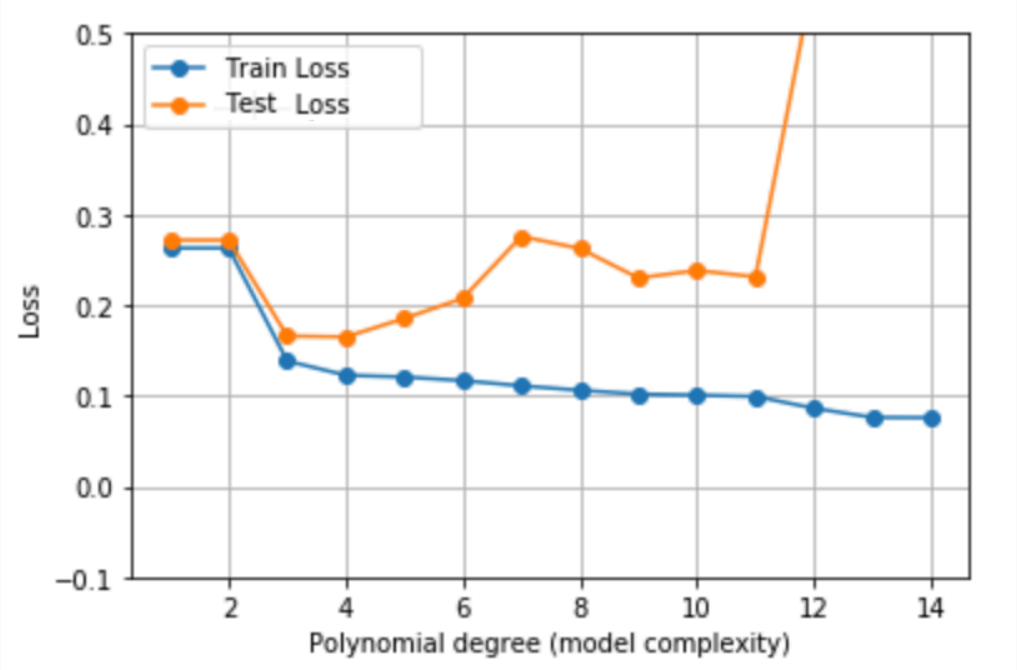
Better approach: Evaluate model on fresh test data which was not used during training.

Test/train split:

- Given data set $(\mathbf{X}, \mathbf{y})$, split into two sets $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ and $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$.
- Train q models $f^{(1)}, \ldots, f^{(q)}$ by finding parameters which minimize the loss on $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$.
- Evaluate loss of each trained model on $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$.

Sometimes you will see the term validation set instead of test set. Sometimes there will be both: use validation set for choosing the model, and test set for getting a final performance measure.

TRAIN-TEST PARADIGM

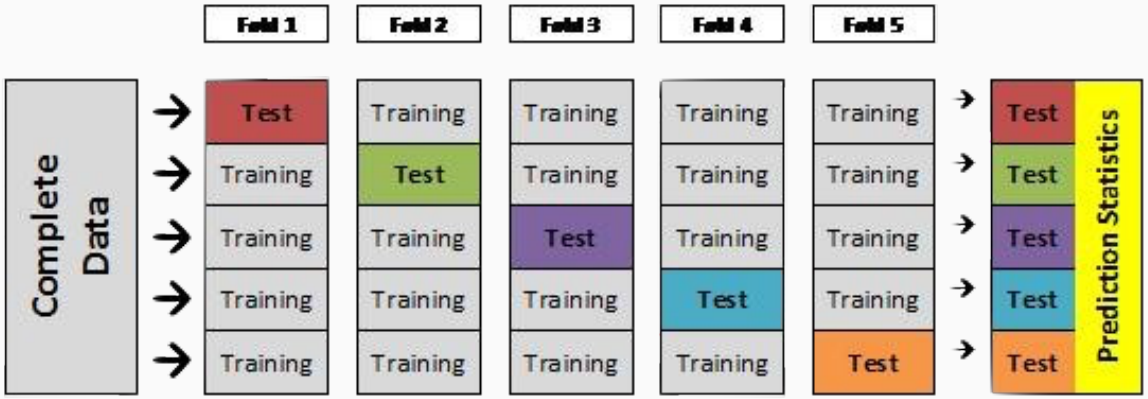


- **Train loss** continues to decrease as model complexity grows.
- **Test loss** "turns around" once our model gets too complex. Minimized around degree 3 – 4.

TRAIN-TEST PARADIGM

Typical train-test split: 70-90% / 10-30%. Trade-off between between optimization of model parameters and better estimate of model performance.

Cross-validation can offer a better trade off:



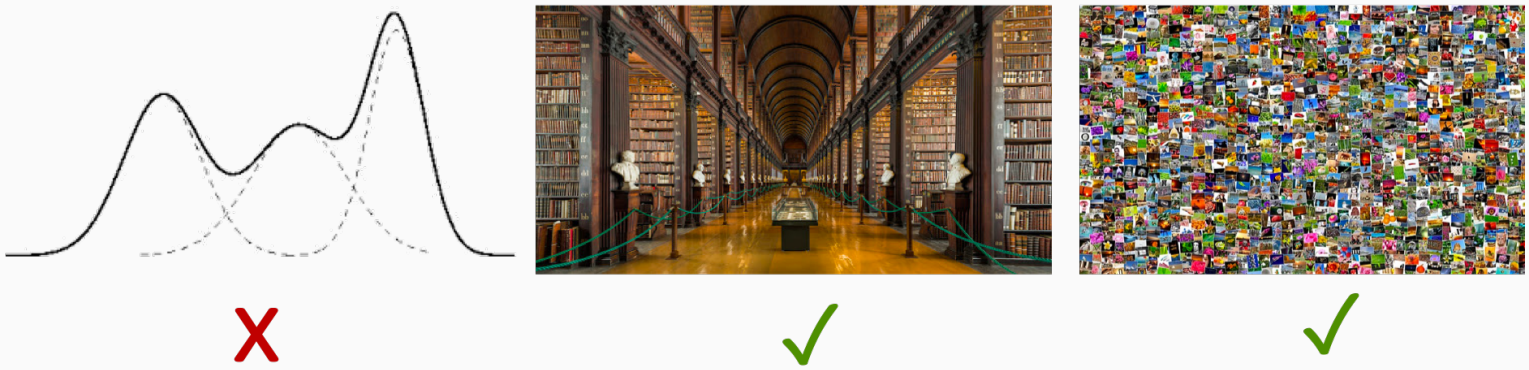
TRAIN-TEST INTUITION

Intuition: Models which perform better on the test set will generalize better to future data.

Goal: Introduce a little bit of formalism to better understand what this means. What is “future” data?

Statistical Learning Model:

- Assume each data example is randomly drawn from some distribution $(\mathbf{x}, y) \sim \mathcal{D}$.



This is not a simplifying assumptions! The distribution could be arbitrarily complicated.

Statistical Learning Model:

- Assume each data example is randomly drawn from some distribution $(\mathbf{x}, y) \sim \mathcal{D}$.
- Define the Risk of a model/parameters:

$$R(f, \boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[L \left(f(\mathbf{x}, \boldsymbol{\theta}) - y \right) \right]$$

here L is some loss function (e.g. $L(z) = |z|$ or $L(z) = z^2$).

Goal: Find model $f \in \{f^{(1)}, \ldots, f^{(q)}\}$ and parameter vector $\boldsymbol{\theta}$ to minimize the $R(f, \boldsymbol{\theta})$.

RISK

- **(Population) Risk**:

$$R(f, \boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[L \left(f(\mathbf{x}, \boldsymbol{\theta}) - y \right) \right]$$

- **Empirical Risk**: Draw $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \sim \mathcal{D}$

$$R_E(f, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L \left(f(\mathbf{x}, \boldsymbol{\theta}) - y \right)$$

Minimizing training loss is the same as minimizing the empirical risk of the training data.

Often called empirical risk minimization.

For any fixed model f and parameters $\boldsymbol{\theta}$,

$$\mathbb{E}\left[R_E(f, \boldsymbol{\theta})\right] = R(f, \boldsymbol{\theta}).$$

Only true if f and $\boldsymbol{\theta}$ are chosen *without looking at the data used to compute the empirical risk.*

- Train q models $(f^{(1)}, \boldsymbol{\theta}_1^*), \ldots, (f^{(q)}, \boldsymbol{\theta}_q^*)$.
- For each model, compute empirical risk $R_E(f^{(i)}, \boldsymbol{\theta}_i^*)$ using test data.
- Since we assume our original dataset was drawn independently from $\mathcal{D}$, so is the random test subset.

No matter how our models were trained or how complex they are, $R_E(f^{(i)}, \boldsymbol{\theta}_i^*)$ is an unbiased estimate of the true risk $R(f^{(i)}, \boldsymbol{\theta}_i^*)$ for every i. Can use it to distinguish between models.

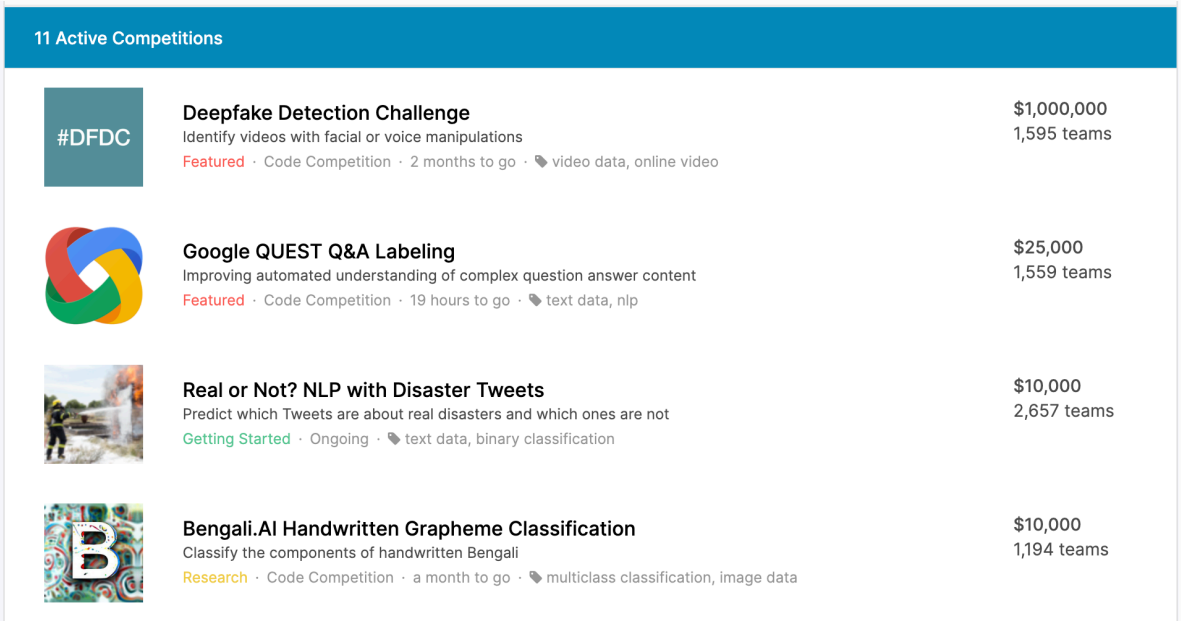
This is typically not how machine learning or scientific discover works in practice!

Typical workflow:

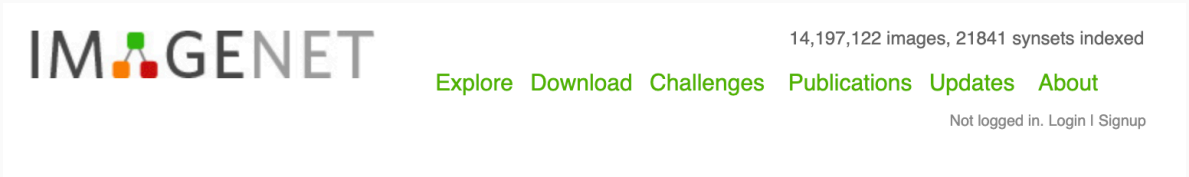
- Train a class of models.
- Test.
- Adjust class of models.
- Test.
- Adjust class of models.
- Cont...

Final model implicitly depends on test set because performance on the test set guided how we changed our model.

Popularity of ML benchmarks and competitions leads to adaptivity at a massive scale.



Kaggle (various competitions)



Imagenet (image classification and categorization)

Is adaptivity a problem? Does it lead to over-fitting? How much? How can we prevent it? All current research.

REPORT

The reusable holdout: Preserving validity in adaptive data analysis

Cynthia Dwork[1,*], Vitaly Feldman[2,*], Moritz Hardt[3,*], Toniann Pitassi[4,*], Omer Reingold[5,*], Aaron Roth[6,*]

+ See all authors and affiliations

Science 07 Aug 2015:
Vol. 349, Issue 6248, pp. 636-638
DOI: 10.1126/science.aaa9375

Do ImageNet Classifiers Generalize to ImageNet?

Benjamin Recht* UC Berkeley
Rebecca Roelofs UC Berkeley
Ludwig Schmidt UC Berkeley
Vaishaal Shankar UC Berkeley

12 Jun 2019

Abstract

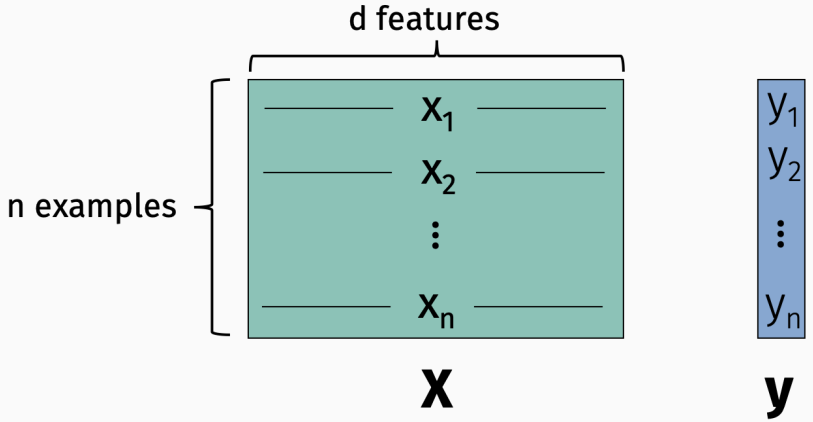
We build new test sets for the CIFAR-10 and ImageNet datasets. Both benchmarks have been the focus of intense research for almost a decade, raising the danger of overfitting to excessively re-used test sets. By closely following the original dataset creation processes, we test to what extent current classification models generalize to new data. We evaluate a broad range of models and find accuracy drops of 3% – 15% on CIFAR-10 and 11% – 14% on ImageNet. However, accuracy gains on the original test sets translate to larger gains on the new test sets. Our results suggest that the accuracy drops are not caused by adaptivity, but by the models' inability to generalize to slightly "harder" images than those found in the original test sets.

REGULARIZATION

OVER-PARAMETERIZED MODELS

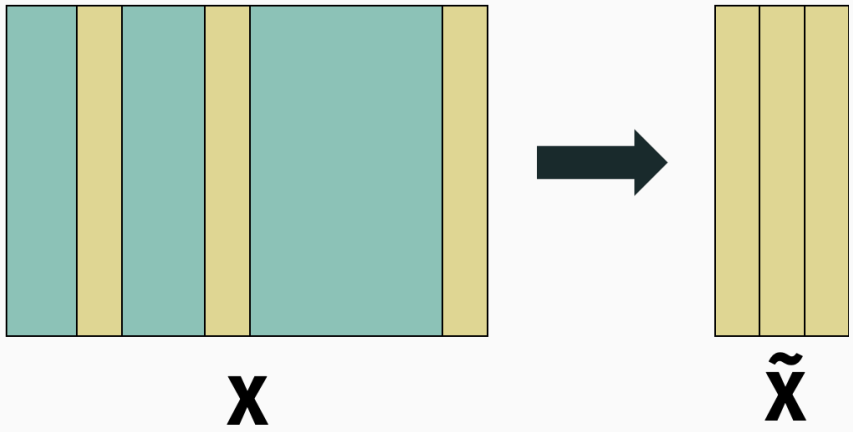
In all the model selection examples we've discussed we had full control over the complexity of the model: could range from underfitting to overfitting.

In practice, you often don't have this freedom. Even the most basic model will overfit.



Example: Linear regression model where $d \geq n$. Can always find β so that $\mathbf{X}\beta = \mathbf{y}$ exactly.

Select some subset of features to use in model:



Filter method: Compute some metric for each feature, and select features with highest score.

- Example: compute loss/R^2 value when each feature in $\mathbf{X}$ is used in single variate regression.

Any potential limitations of this approach?

FEATURE SELECTION

Exhaustive approach: Pick best subset of q features.

Faster approach: Greedily select q features.

Stepwise Regression:

- **Forward:** Step 1: pick single feature that gives lowest loss. Step k: pick feature that when combined with previous $k-1$ chosen features gives lowest loss.
- **Backward:** Start with all of the features. Greedily eliminate those which have least impact on model performance.

Feature selection deserves more than two slides, but we won't go into too much more detail!

Regularization: Explicitly discourage overfitting by adding a regularization penalty to the loss minimization problem.

$$\min_{\boldsymbol{\theta}} \left[L(\boldsymbol{\theta}) + R(\boldsymbol{\theta}) \right].$$

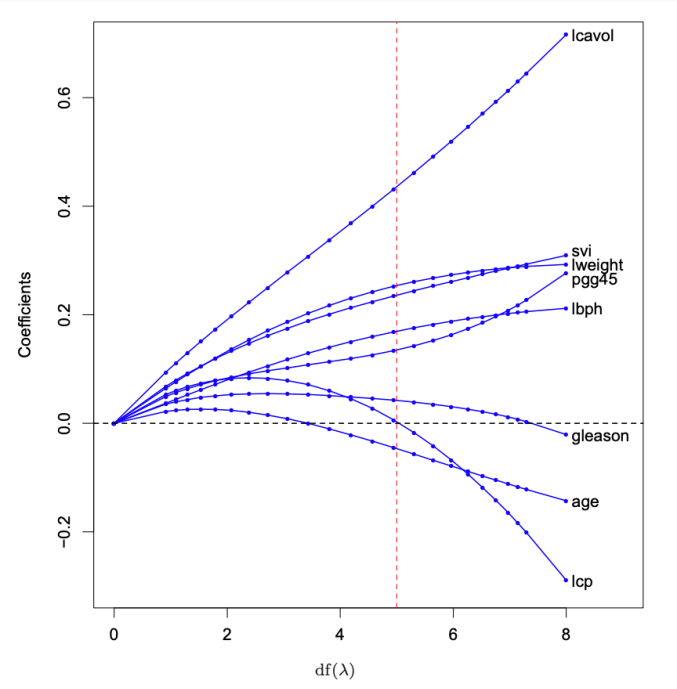
Example: Least squares regression. $L(\boldsymbol{\beta}) = \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$.

- Ridge regression (ℓ_2): $R(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_2^2$
- LASSO (least absolute shrinkage and selection operator) (ℓ_1): $R(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_1$
- Elastic net: $R(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$

RIDGE REGULARIZATION

Ridge regression: $\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$.

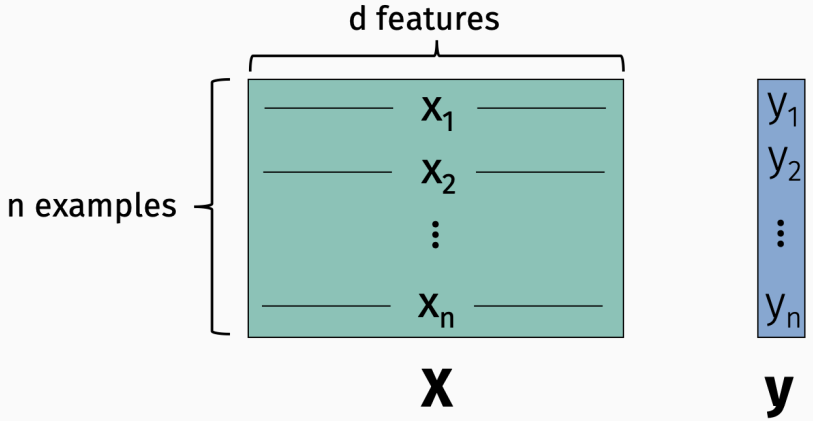
- As $\lambda \to \infty$, we expect $\|\boldsymbol{\beta}\|_2^2 \to 0$ and $\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \to \|\mathbf{y}\|_2^2$.
- Feature selection methods attempt to set many coordinates in $\boldsymbol{\beta}$ to 0. Ridge regularizations encourages coordinates to be small.



RIDGE REGULARIZATION

Ridge regression: $\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$.

- Can be viewed as shrinking the size of our model class. Relaxed version of $\min_{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_2^2 < C} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$. Which won't have a solution at zero for all $\mathbf{y}$, even when over-parameterized.

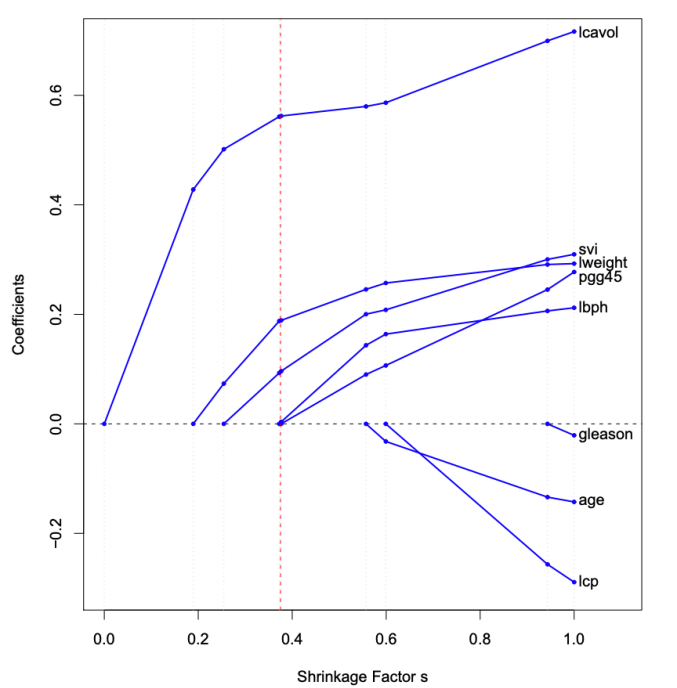


- Method is not invariant to data scaling. Typically when using regularization we mean center and scale columns to have unit variance.

LASSO REGULARIZATION

Ridge regression: $\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$.

- As $\lambda \to \infty$, we expect $\|\boldsymbol{\beta}\|_1 \to 0$ and $\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \to \|\mathbf{y}\|_2^2$.
- Typically encourages subset of $\boldsymbol{\beta}_i$ to go to zero, in contrast to ridge regularization.



LASSO REGULARIZATION

Pros:

- Simpler, more interpretable model.
- More intuitive reduction in model order.

Cons:

- No closed form solution because $\|\beta\|_1$ is not differentiable.
- Can be solved with iterative methods, but generally not as quickly as ridge regression.