

# CV\_M8\_L01

# Training Robust Models

**CS/DSA 5970: Machine Learning Practice**

# Model Fitting Challenges

We have already seen a case where a model can dramatically over-fit the available training data

- The first thing to try (if it is an option): add more data
- The next thing: add bias to the learning process to prefer certain types of solutions
  - In our examples, Ridge, Lasso and Elastic Net added a bias that preferred small coefficients

# Hyper-Parameters

Regularization introduced model hyper-parameter(s):

- Our regularization parameter expressed a trade-off between explaining the training data (e.g., by reducing MSE) and making the models simple (or smooth)
- For the Elastic Net, we had a 2<sup>nd</sup> hyper-parameter that expressed the balance between the L1 and L2 coefficient norms

# Hyper-Parameters

Other Types of Hyper-Parameters:

- Degree of polynomial used during feature preprocessing
- Maximum depth of a decision tree
- Minimum entropy in a decision tree leaf
- Number of layers and size of each layer in a deep network

# Training Robust Models

When we are faced with a new modeling problem, we fundamentally want to answer:

- What is the best model type to use (form and algorithm)
- How do we select the hyper-parameters?

  

- We want to be confident in our choice moving forward
- In particular, we want to make a statistically-sound decision

# The Model Bake-Off

A possible approach to comparing model form:

- Use Cross-Validation to make hyper-parameter choices for each model type
- Use the same CV approach to then compare the model types
- As presented, Cross-Validation can cause us to over-fit the hyper-parameters

# Outline

- The Charlatan Problem
- Choosing hyper-parameters
  - Grid search
- Holistic Cross-Validation
- Statistically comparing model types



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L02

# The Charlatan Problem

**CS/DSA 5970: Machine Learning Practice**

# The Charlatan Problem

What is a good algorithm for empirically choosing a stock broker?

# The Charlatan Problem

What is a good algorithm for empirically choosing a stock broker? One possibility:

- We ask the broker to make a judgement on a set of stocks as to whether they will go up or down in value in the next week (each is a binary question)
- At the end of the week, we ask how many the broker got right (also binary questions)

# The Charlatan Problem

How do we evaluate this statistically?

# The Charlatan Problem

How do we evaluate this statistically?

- Null hypothesis: the broker is a charlatan & doesn't really know how to choose
- Assume that choices are just made with a coin flip ( $p = 0.5$ )
- We will assume that the true probability of going up is also  $p = 0.5$
- How well do we expect the broker to do in this case?

# Guessing for a Single Stock

Under the null hypothesis, the charlatan will be correct for any one guess 50% of the time

- How about with  $N$  stocks?

# Guessing the Outcome of N Stocks

We expect half of the guesses will be correct

- Are there other possible outcomes?

# Guessing the Outcome of N Stocks

Are there other possible outcomes?

- Yes!  $N/2-1$  and  $N/2+1$  are equally likely (with each other)
- The probability drops off as we get further away from  $N/2$
- As  $N$  gets large, what does this distribution look like?

# Guessing the Outcome of N Stocks

- As  $N$  gets large, the distribution tends toward a Gaussian with mean  $N/2$
- Central Limit Theorem: the sum (or mean) of  $N$  samples from any distribution tends towards a Gaussian distribution as  $N$  gets large
- $N=30$  is a good place to be ( $N=20$  is still very close)

# Guessing the Outcome of N Stocks

How do we decide whether to hire the stock broker after we have done this experiment?

# Guessing the Outcome of N Stocks

How do we decide whether to hire the stock broker after we have done this experiment?

- If the number correct is large enough, then the probability of guessing correctly is a small probability under the null hypothesis assumption

## IPAD\_M9\_L02b

- Probability distribution under the null hypothesis
- Statistical value: a particular observation
- Integral of likelihood above the critical value: the probability of observing that or greater value

# Guessing the Outcome of N Stocks

Some terms:

- p-value: the estimated probability of incorrectly rejecting the null hypothesis
- alpha-value: the largest acceptable probability of incorrectly rejecting the null hypothesis
- critical value: the value of the statistic that corresponds to the alpha-value
  - In our case, this is the number of correctly selected stocks at which we accept that the stock broker is not a charlatan

# Choosing Alpha

Choosing alpha depends on context and how well we need to trust the result

- Typical: 5%
- Stretch: 10% (but some will argue)
- Life and death: 1% - .1%

# This is just hypothesis testing ...

- Assume that there is no difference in the way two models perform
- Ask how likely it is that we observe samples of performance from each of the two models under this assumption
- If the probability is too small, then we reject the assumption (which is what we want)

# Implications

- A model, even if it is not better than a competitor, can look good with some probability
- But, we can control the probability of making a mistake if we present the model with enough tests and have a high enough criterion



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L03

# The Many Charlatans Problem

**CS/DSA 5970: Machine Learning Practice**

# The Multiple Charlatans Problem

- We have already discussed a test that will reveal whether a stock broker can be hired. This test only makes a mistake with probability alpha
- How do we make the search process for a broker more efficient?

# The Multiple Charlatans Problem

How do we make the search process for a broker more efficient?

- Let's test K brokers in parallel!
- Each broker gets the same stocks to judge
- Choose the broker with the highest accuracy
  - Must also have a low enough p-value

# The Multiple Charlatans Problem

The broker with the highest accuracy is selected and this accuracy is above a critical threshold

- What is the probability that we have made a mistake?
- I.E.: assume that all are charlatans. What is the probability that we still accept someone?

## IPAD\_M8\_L03b

- Math for computing aggregate alpha assuming alphas of 0.05
- Math for computing individual alphas given an aggregate alpha of 0.5
- Corrections options: Bonferroni vs Sidek

# Solving Multiple Comparisons Problem

Comparing one sampling against more than one other sampling dilutes the power of the individual comparisons.

Options for addressing:

- Correct the alpha
  - This is a very conservative approach, but is effective
- Once we have selected the best, we take new samples to do the final comparisons

# Selecting Model Types and Hyper-Parameters

This is really a multi-level question

- For a given model type, we first need to know what the best hyper-parameter set is. This can involve *\*a lot\** of comparisons
- Then, we can begin to compare model types
- Typical approach: use different data sets for these two levels

# Selecting Model Types and Hyper-Parameters

Typical approach: use independent data for these two levels

- Validation data: use for selection of hyper-parameters
- Test data: use for comparing models



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L04

# Hyper-Parameter Selection

**CS/DSA 5970: Machine Learning Practice**

# Hyper-Parameters

- Every problem will require different choices for hyper-parameters
- We typically formulate this as a process of search
- With experience, you will achieve some intuition as to where to start this search

# Hyper-Parameters

Some of the algorithms that we have covered to date have a single hyper-parameter

- The problem becomes a search along a number line
- Typical to establish a range for this search
- Then select, we select a spacing:
  - Exponential
  - Regular

## IPAD\_M8\_L04b

- Illustrate exponential vs regular spacing
- For each choice: train model, measure performance on training and validation data
- Talk about edge-effects
- Keep recording going

# Single Hyper-Parameter

- Exponential spacing:
  - Cover a wide range
  - Good for quickly narrowing down a region for further focus
  - Factors of 10 vs factors of 2
- Regular spacing
  - Cover a narrow range
  - Allows us to achieve a careful tuning of the parameters

# Multiple Hyper-Parameters

- More common case
- Hyper-parameters are generally not independent
- Cannot conduct a search as if they are independent

# Grid Search

Approach:

- Each hyper-parameter has its own set of values that we would like to test
  - Can use exponential or regular spacing
- Then, we consider the Cartesian product of these choices

## IPAD (still in prior recording)

- Show grid of 2 parameters
- For each grid cell: learn model, compute performance wrt training and validation sets
- Draw grid of 3 parameters

# Grid Search

- When the number of hyper-parameters starts to get large (where large  $> 3$  or  $4$ ), we really have too many cases to consider
- Alternative approach:
  - Fix all but 2 or 3 of the hyper-parameters
  - Perform a grid search across these 2-3 hyper-parameters
  - Pick the parameters with the best performance with respect to the validation set
  - Repeat with another subset of hyper-parameters

# Grid Search

- Even when we have a grid of hyper-parameters, we can unroll this grid into a line
- We will make use of this abstraction as we take the next step

# Grid Search

Scikit-Learn provides a facility for doing grid search automatically: GridSearchCV

- Takes as input:
  - An instance of a model object
  - A list of hyper-parameters to vary
  - For each hyper-parameter: a list of values to try
- Fits a model for every combination of the hyper-parameters and each cross-validation fold

# Grid Search

## GridSearchCV

- For each hyper-parameter set, the model is trained  $N$  times ( $N$ -fold cross-validation)
- So, we get  $N$  performance measures for each hyper-parameter set
- Gives the best set of hyper-parameters with respect to this distribution

# GridSearchCV Limitations

- The training and validation process consumes all of the data
- This does not leave any independent data for comparing across the types of models
- The Scikit-Learn answer: hold data out from this process so that it can be used to independently measure performance of winning hyper-parameter set
  - This does not give us a way to ask a statistical question when comparing model types



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L05

# Example: Grid Search

**CS/DSA 5970: Machine Learning Practice**

# Example: Grid Search

- Regularization parameter for BMI and Ridge Regression and Elastic Net



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L06

# Holistic Approach to Cross-Validation

**CS/DSA 5970: Machine Learning Practice**

# The Model Bake-Off

Two levels of search:

- Choose the best hyper-parameters for a given model type
- Compare model types

Challenge:

- If we use the same data for making both choices, we can over-fit the hyper-parameters
- The implication is that we may not perform well on future data

# The Model Bake-Off

We need to ask these questions with:

- Independent data
- Multiple samples of the performance metric

# Holistic Cross-Validation

One cross-validation split:

- Training data: fit the model
- Validation data: performance measure for hyper-parameter selection
- Test data: performance measure for model type choice

## IPAD\_M8\_L06b

- Cross-validation splits and rotation
- Grid of models
- Compute performance for hyper-parameter choice on validation data
- Select hyper-parameters
- Compare with other model types (high-level)
- Splits when varying training set size



*The* UNIVERSITY *of* OKLAHOMA

# CV\_M8\_L07

# Comparing Models

**CS/DSA 5970: Machine Learning Practice**

# Holistic Cross-Validation

- When we are selecting model parameters, we potentially have a large number of parameter sets
- Neighboring parameter sets should yield similar performance
  - Be suspicious if this is not the case
- We address the many charlatans problem by using independent data sets:
  - Validation data set: parameter selection
  - Test data set: only use at the very end when you are justifying your use of the model (including comparison to a small number of alternatives)

# Holistic Cross-Validation: Comparing Models

For each model type, we have:

- N performance measures (one for each model instance / test set combination)
- Because these performance measures are independent of the data used for hyper-parameter selection, we won't be fooled by hyper-parameter over-fit
- Have the opportunity to perform paired comparisons
  - Pairs are between models for a given test data set

## IPAD\_M8\_L07

- Hyper parameter selection: gives way to best choice for each model
- Paired samples
- Can simply pick the best model: note that this involves  $O(M)$  comparisons to find the max
  - Statistically comparing the best against the next best has to take these  $M$  comparisons into account
- Comparing  $M$  models against a baseline model: if we want to make a statistical argument that one of the  $M$  is better than the baseline, then we assume that we have  $M$  charlatans that we are comparing against the baseline

# Comparing Model Types

Compare two models statistically:

- Often see a student t-test (one tailed)
  - Nominally assumes an underlying normal distribution of the performance metric
- Sample based approaches: computationally involved, but are robust to any underlying distribution
  - For example: bootstrap resampling and bootstrap randomization
  - And, can be constructed to address lack of independence in the samples or even censored samples

# Comparing M Model Types

- If we have more than 2 model types in question, then we again have the multiple charlatans problem
- ANOVA:
  - Tells us that at least one model is different from the others (but not which one)
  - But, this gives us permission to perform pairwise tests

# Comparing M Model Types

- We can perform all  $M * M-1 / 2$  comparisons
  - But this is often overkill
- A couple common alternatives:
  - We want to pick the best of  $M$  & make a statistical argument that this is the best one
    - This involves  $O(M)$  comparisons
  - We want to show that at least one of  $M$  is statistically better than some baseline model
    - Exactly  $M$  comparisons are made here
- In any of these cases: we use Bonferroni or Sidèk correction to adjust the p-value cutoff

# Comparing M Model Types

- Remember that Bonferroni and Sidèk are very conservative adjustments to the p-value cutoff
- An alternative: use new data
- Cross-validation then involves 4 data sets:
  - Train, validation, test1, test2
  - Use test1 to pick the favorite model of M (and the next best one)
  - Use test2 to confirm that it is the best (a single statistical test!)

# Final Thoughts

- Hyper-parameters:
  - If you are doing tests that examine how much training data you need for a given model, this is also a hyper-parameter
  - Make this selection based on validation performance
- Data cutting:
  - Be sure that samples across the folds are independent
  - Time series data can have a lot of autocorrelation
    - Cutting a contiguous block of data (in time) can yield folds that exhibit some autocorrelation

# Final Thoughts

Cross-Validation (even Holistic) can be fooled

- The cutting of the data can give you “lucky” results
- Typical approach (not very common, though):
  - Re-cut the data info a new set of folds and repeat the procedure (multiple times)
  - We expect consistent results
  - Beyond our scope: formal methods for deciding how many times to repeat



*The* UNIVERSITY *of* OKLAHOMA



*The* UNIVERSITY *of* OKLAHOMA