PGE 383

Tuning Hyperparameters

- **Training and Testing**
- **Model Goodness Metrics**
- **Cross Validation Workflows**

Michael Pyrcz, The University of Texas at Austin

Motivation

Let’s formalize concepts and define terms for hyperparameter tuning.

How do we judge a model as good?

- What is a good match with the training data as part of a loss function?
  - Has impact on model performance, e.g., robustness, sparse outputs, multiple solutions, differentiable for analytical solution
- What is a good match with the withheld testing data?
  - Selection of the best level of model complexity

PGE 383
Ridge Regression

- **Training and Testing**

Michael Pyrcz, The University of Texas at Austin

Model Parameters Definition

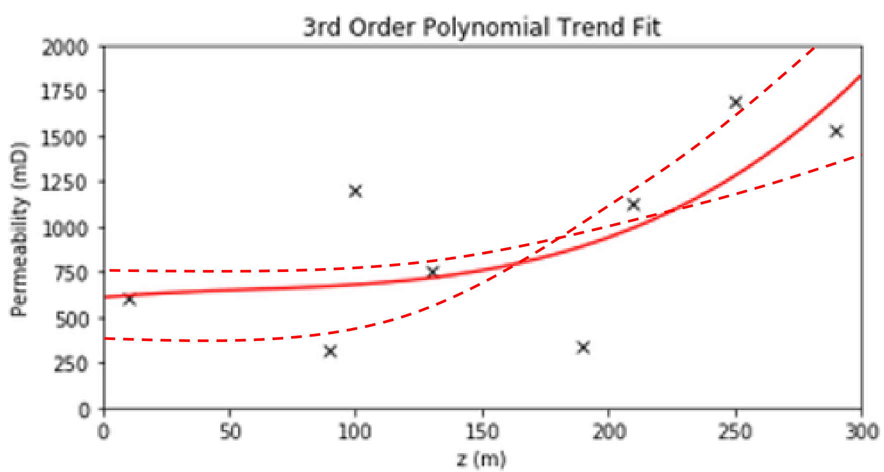
Model Parameters

Derived during training phase to fit the model to the training data

Parameters

$$k = b_3 z^3 + b_2 z^2 + b_1 z + c$$

b_3, b_2, b_1 and c



Model Hyperparameter Definition

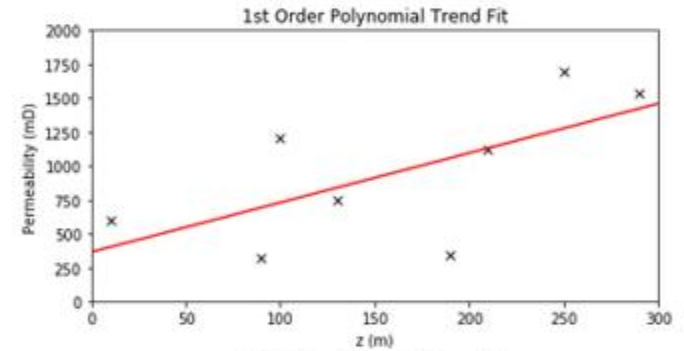
Model Hyperparameters

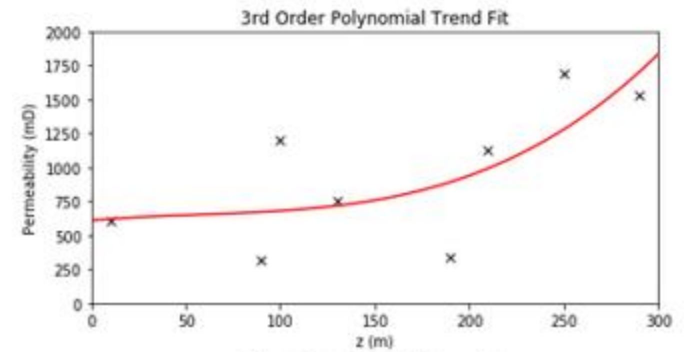
Set prior to learning from the data. Impact the form of the model and often the complexity.

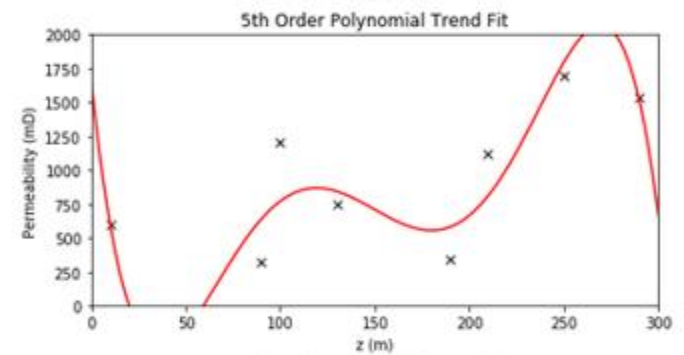
3rd Order: $$k = b_3 z^3 + b_2 z^2 + b_1 z + c$$

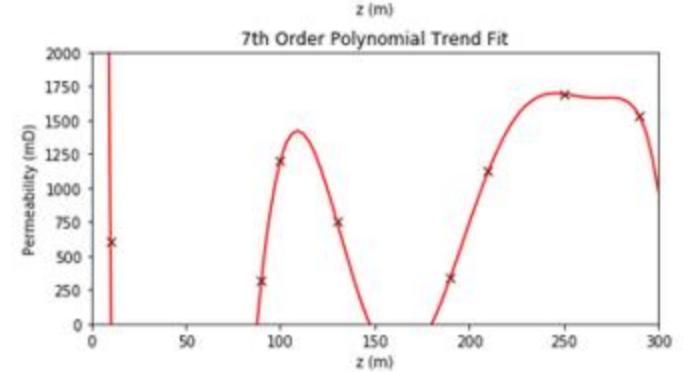
2nd Order: $$k = b_2 z^2 + b_1 z + c$$

1st Order: $$k = b_1 z + c$$





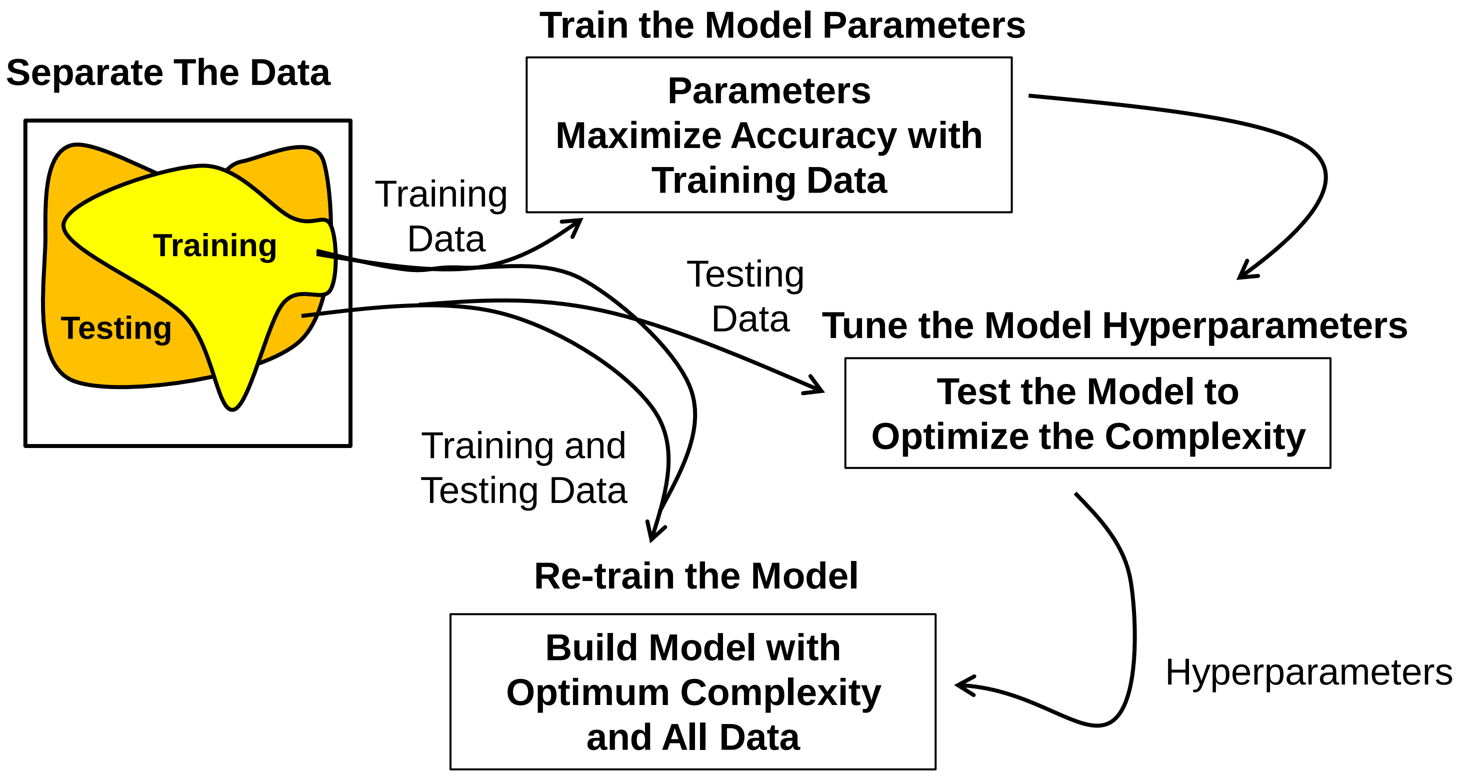




Training and Testing

The Training and Testing Workflow

- establish a subset of the data for fair testing of the model

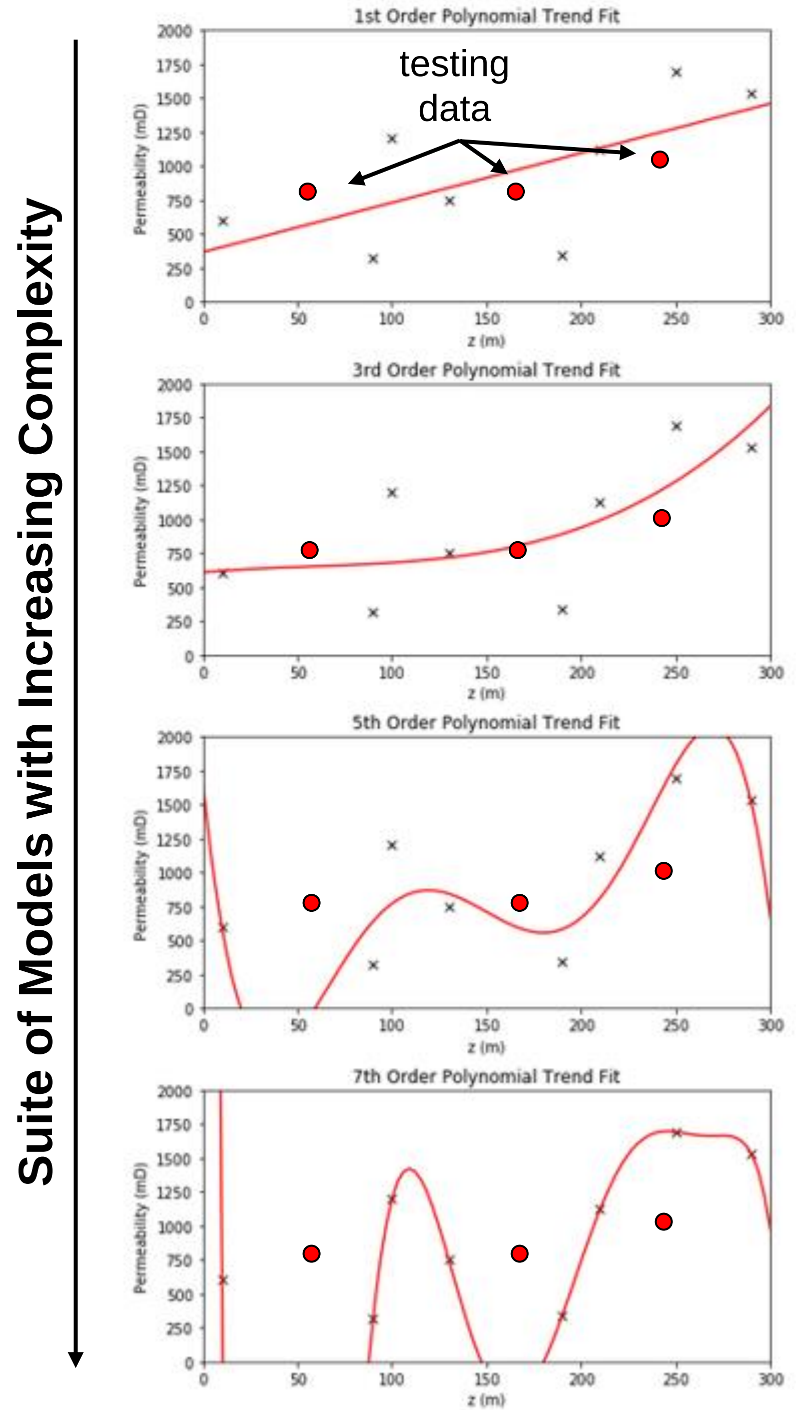


We avoid the overfit problem.

Hyperparameter Tuning

What do we have?

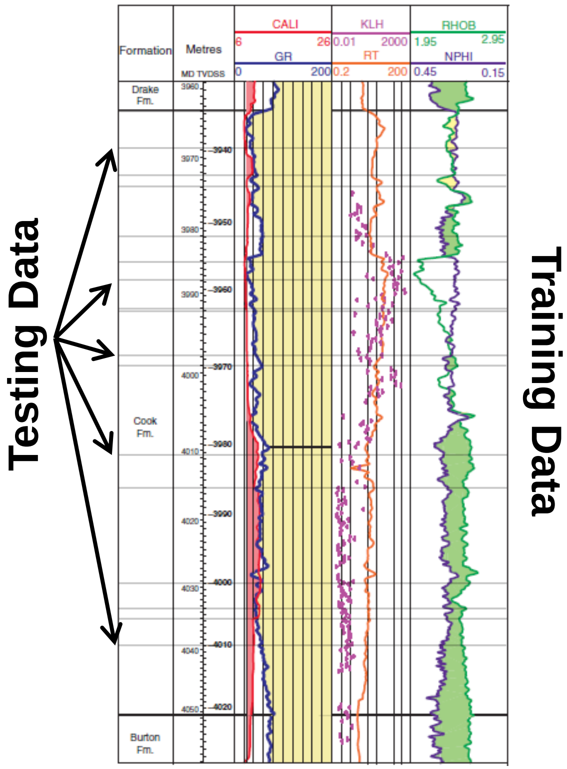
- A suite of models of variable level of complexity, and other decisions informed by a range of hyperparameters.
- We have a set of testing data withheld from the training of the model parameters.
- A measure of performance, e.g., an accuracy norm, like mean square error for each model etc.



Training and Testing Split

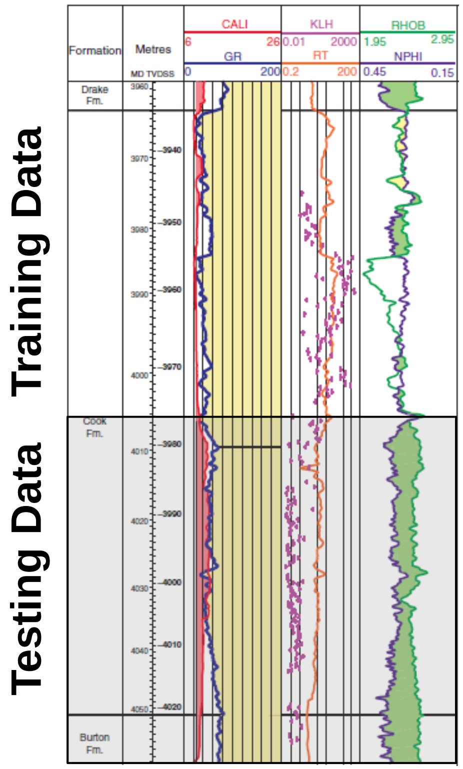
Fair Testing in Spatial / Temporal Settings

Too Easy



Predictions only at ½ ft offsets

Too Hard



Predictions in a different rock.

The Train and Test Split Should Be Fair

The prediction difficulty (interpolation, extrapolation) should be similar to the planned real-world use of the prediction model.

Training and Testing Split

How Much Data Should be Used in Testing?

The proportion in testing is recommended by various sources from 30% - 15% of the total dataset.

- Data withheld for testing reduces the data available for training; therefore, reduces the accuracy of the model.

- Data withheld for testing improves the accuracy of the assessment of the model performance.

- Various authors have experimented on a variety of training and testing ratios and have recommended splits for their applications:
  - The optimum ratio of training and testing split depends on problem setting
  - Could consider the difficulty in training (e.g., the number of parameters) and the difficulty in testing (e.g., number of hyperparameters).

Alternative Testing Workflows

Training, Validation and Testing

There is a more complete workflow commonly applied.

Note: to avoid confusion in our class we will use the train and test approach only.

- **Train with training data**. Models sees and learns from this data to train the model parameters.
- **Validate with validation data**. Unbiased evaluation of model fit to tune the model hyperparameters (testing data in train and test workflow).
- **Test with testing data.** Data withheld until the model is complete to provide a final evaluation. Commonly applied to compare multiple competing models. This data had no role in building the model.

The train, validate and test data splits.

Image and concepts modified from https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

PGE 383
Ridge Regression

- **Model Goodness Metrics**

Michael Pyrcz, The University of Texas at Austin

Training and Testing Metrics

Training and Testing Metrics

To evaluate model performance, we need to access the goodness of the suite of models with respect to the training and testing data.

- An accuracy norm to summarize the vector of errors.
- There are a variety of metrics that are applied for this assessment.
- They depend primarily on classification vs. regression.
- We will cover regression metrics first.
- Here we are not talking about loss functions, etc., so we refer to testing only.

Regression Testing Metrics

Regression Testing Metrics

Mean Square Error (MSE)

- L^2 norm – sensitive to large errors

$$Test\ MSE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2 = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (\Delta y_i)^2$$

Mean Absolute Error (MAE)

- L^1 norm – less sensitive to large errors

$$Test\ MAE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} |y_i - \hat{y}_i| = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} |\Delta y_i|$$

Regression Testing Metrics

Regression Testing Metrics

Variance Explained – note, **we only use it for linear models**

- Proportion of variance of the response feature captured by the model
- Takes advantage of the additivity of variance
  - Total Variance = Variance Explained + Variance Not Explained

$$\boldsymbol{\sigma^2_{explained}} = \frac{\mathbf{1}}{n_{test}} \sum_{\boldsymbol{i=1}}^{n_{test}} (\boldsymbol{\hat{y}_i} - \boldsymbol{\bar{y}})^{\mathbf{2}} \qquad \boldsymbol{\sigma^2_{not\ explained}} = \frac{\mathbf{1}}{n_{test}} \sum_{\boldsymbol{i=1}}^{n_{test}} (\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}$$

$$r^2 = \frac{\sigma^2_{explained}}{\sigma^2_{explained} + \sigma^2_{not\ explained}} = \frac{\sigma^2_{explained}}{\sigma^2_{total}}$$

Issues

- recall $r^2 = (\rho)^2$, suffers the same issues of correlation coefficients, linearity, do not account for redundancy, sensitive to outliers.
- e.g., recall impact of adding outliers and 2 populations on correlation!

Regression Testing Metrics

Regression Testing Metrics

Inlier Ratio, proportion of testing data within a margin, ϵ, of the model $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$.

$$\boldsymbol{IR} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} I(y_i, \hat{y}_i)$$

Given the indicator transform:

$$I(y_i, \hat{y}_i) = \begin{cases} 1, & \text{if } |y_i - \widehat{y_i}| \leq \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Testing data, model with margin and inliers identified, image from https://upload.wikimedia.org/wikipedia/commons/b/b7/RANSAC_Inliers_and_Outliers.png

Classification Testing Metrics

Classification Testing Metrics

Confusion Matrix

- Matrix with the categorical truth vs. predicted
- Visualize and diagnose all the combinations of correct and misclassification with the classification model.

Model says category 3, Data value is category 1.

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

Predicted

- Perfect accuracy is number of each class in the training data on the diagonal.

Truth			
	n_1	0	0
	0	n_2	0
	0	0	n_3

Predicted

Classification Testing Metrics

Precision – for group k, the ratio of true positive over all positives.

$$precision_k = \frac{n_{k\ true\ positive}}{n_{k\ true\ positive} + n_{k\ false\ positive}} = \frac{true\ positive}{all\ positives}$$

$$precision_k = Probability(k\ is\ happenning \mid the\ model\ says\ k\ is\ happenning)$$

$\boldsymbol{k = 1}$

true positive

false positive

Truth	$\widehat{C}_{k=1}$	$\widehat{C}_{k=2}$	$\widehat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{15}{15+(5+7)} = \frac{15}{27} = 0.56$$

$\boldsymbol{k = 2}$

true positive

false positive

Truth	$\widehat{C}_{k=1}$	$\widehat{C}_{k=2}$	$\widehat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{22}{22+(15+15)} = \frac{22}{52} = 0.42$$

$\boldsymbol{k = 3}$

true positive

false positive

Truth	$\widehat{C}_{k=1}$	$\widehat{C}_{k=2}$	$\widehat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{4}{4+(2+9)} = \frac{4}{15} = 0.27$$

Classification Testing Metrics

Recall – for group k, the ratio of true positive over all cases of k.

$$Recall_k = \frac{n_{k\ true\ positive}}{n_k}$$

How many of group k did we catch? Does not account for false positives.

$\boldsymbol{k = 1}$

true positive

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{15}{15+(15+2)} = \frac{15}{32} = 0.47$$

$\boldsymbol{k = 2}$

true positive

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{22}{22+(5+9)} = \frac{22}{36} = 0.61$$

$\boldsymbol{k = 3}$

true positive

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{4}{4+(7+15)} = \frac{4}{26} = 0.15$$

Classification Testing Metrics

Sensitivity (same as Recall) and Specificity (medical and biology)

$$Sensitivity_k = Recall_k = \frac{n_{k\ true\ positive}}{n_k} \qquad Specificity_k = \frac{n_{k\ true\ negative}}{n_{\neq k}}$$

How many of not group k did we catch? Does not account for true positives!

$k = 1$

true negative

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{22+9+15+4}{(22+9+15+4)+(5+7)} = \frac{50}{62} = 0.81$$

$k = 2$

true negative

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{15+2+7+4}{(15+2+7+4)+(15+15)} = \frac{28}{58} = 0.48$$

$k = 3$

true negative

Truth	$\hat{C}_{k=1}$	$\hat{C}_{k=2}$	$\hat{C}_{k=3}$
$C_{k=1}$	15	15	2
$C_{k=2}$	5	22	9
$C_{k=3}$	7	15	4

$$\frac{15+15+5+22}{(15+15+5+22)+(2+9)} = \frac{57}{68} = 0.84$$

Classification Testing Metrics

	Precision	Recall	f1-score
$k = 1$	0.56	0.47	0.51
$k = 2$	0.42	0.61	0.49
$k = 3$	0.27	0.15	0.19

Precision and Recall measure 2 components of categorical accuracy. Let's combine them into one measure.

$$f1 - score_k = \frac{2}{\frac{1}{Precision_k} + \frac{1}{Recall_k}}$$

$f1 - score$ is the Harmonic mean of precision and recall for k.

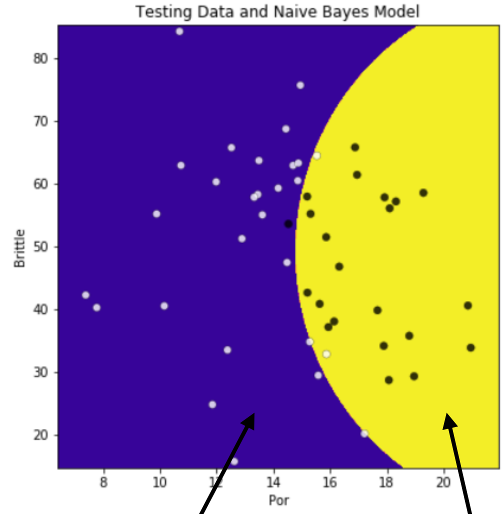
- Sensitive the to lowest score, good performance in one score cannot make up for bad performance in the other!

Classification Testing Metrics

true positive
k=low

$$
\begin{array}{c|cc}
 & \multicolumn{2}{c}{\text{Predicted}} \\
\text{Truth} & \text{Low} & \text{High} \\
\hline
\text{Low} & \boxed{26} & 2 \\
\text{High} & \boxed{1} & 21
\end{array}
$$

false positive
k=low



Low Production

High Production

Classification Testing Metrics

Another example from the naïve Bayes workflow.

- $Precision_{k=Low} = \frac{n_{k\ true\ positive}}{n_{k\ true\ positive} + n_{k\ false\ positive}} = \frac{26}{26+1} = 0.96$

	precision	recall	f1-score	support
Low	0.96	0.93	0.95	28
High	0.91	0.95	0.93	22
avg / total	0.94	0.94	0.94	50

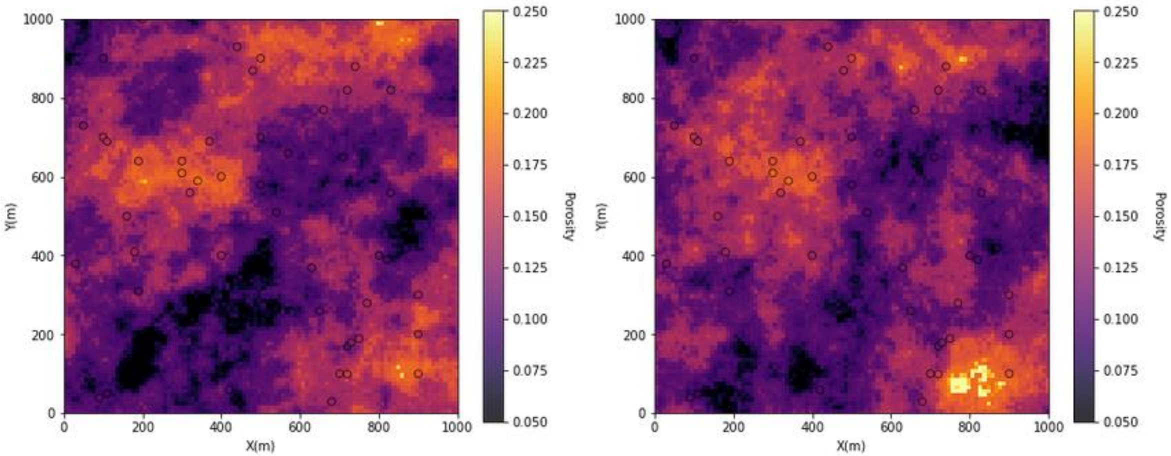
- $Recall_{k=Low} = \frac{n_{k\ true\ positive}}{n_k} = \frac{26}{26+2} = 0.93$

- $f1-score_{k=Low} = \frac{2}{\frac{1}{Precision_{k=Low}} + \frac{1}{Recall_{k=Low}}} = \frac{2}{\frac{1}{0.96} + \frac{1}{0.93}} = 0.95$

Example from workflow: SubsurfaceDataAnalytics_NaiveBayes.ipynb

Image Testing Metrics

What if we want to compare images?



Training image (left) and generated image (right).

Sometimes our model predictions are entire images, we may want to compare our generated images to original training images.

- We will cover methods such as convolutional neural networks that make predictions from images, auto encoders that project images to lower dimensional representations and generative adversarial networks that make new images.

Image Testing Metrics

Pixel-by-pixel comparisons

MSE pixel-by-pixel

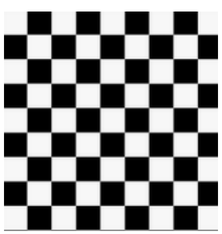
$$Test\ MSE = \frac{1}{n_x \cdot n_y} \sum_{iy=1}^{n_y} \sum_{ix=1}^{n_x} \left(y(\mathbf{u}_{ix,iy}) - \hat{y}(\mathbf{u}_{ix,iy})\right)^2$$

Correlation pixel-by-pixel

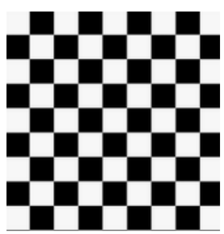
$$\rho_{y,\hat{y}} = \frac{1}{n_x \cdot n_y \cdot \sigma_x \cdot \sigma_x} \sum_{iy=1}^{n_y} \sum_{ix=1}^{n_x} \left(y\left(\mathbf{u}_{ix,iy}\right) - \bar{y}\right) \cdot \left(\hat{y}\left(\mathbf{u}_{ix,iy}\right) - \bar{\hat{y}}\right)$$

- The pixel-by-pixel methods are very sensitive to local exactness.

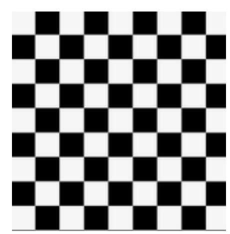
Perfectly Same



$=$



Perfectly Different



$\neq$

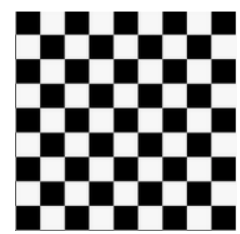


Image Testing Metrics

Generalized image comparison with structural similarity

Based on combination of 3 image comparisons:

1. Luminance (l) – similar average intensity
2. Contrast (c) – similar variance in intensity
3. Structures (s) – correlation between collocated pixels

$$I(a,b) = \frac{2\mu_a\mu_b + c_1}{\mu_a + \mu_a + c_1}$$

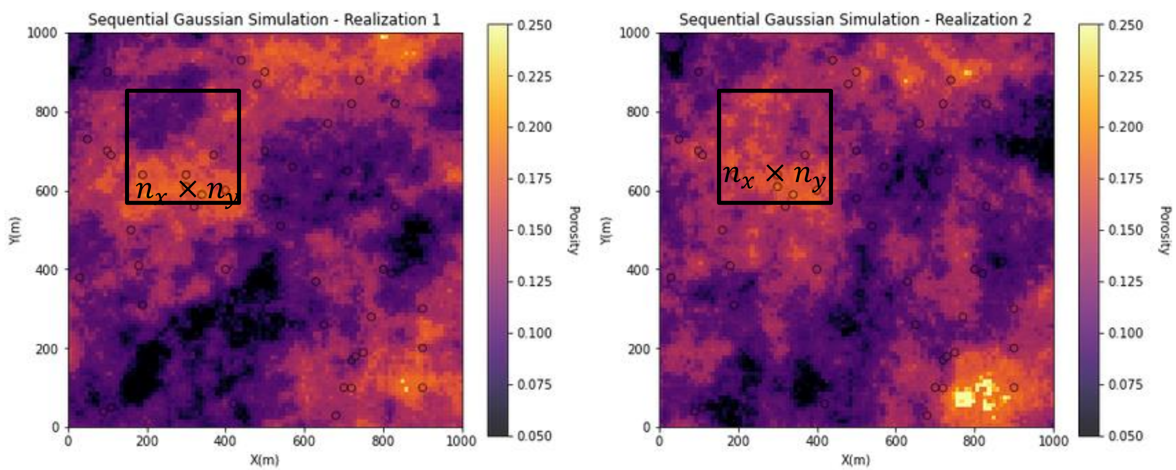
$$c(a,b) = \frac{2\sigma_a\sigma_b + c_2}{\sigma_a^2 + \sigma_b^2 + c_2}$$

$$s(a,b) = \frac{2\sigma_{a,b} + c_3}{\sigma_a\sigma_b + c_3}$$

$$\boldsymbol{SSIM(a,b)} = I(a,b)^{\boldsymbol{\alpha}} \cdot c(a,b)^{\boldsymbol{\beta}} \cdot s(a,b)^{\boldsymbol{\gamma}}$$

where α, β, γ are weights.

The calculation is aggregated our over multiple windows of the images:



Images for comparison (left = a and right = b).

Probability Distribution-based Metrics

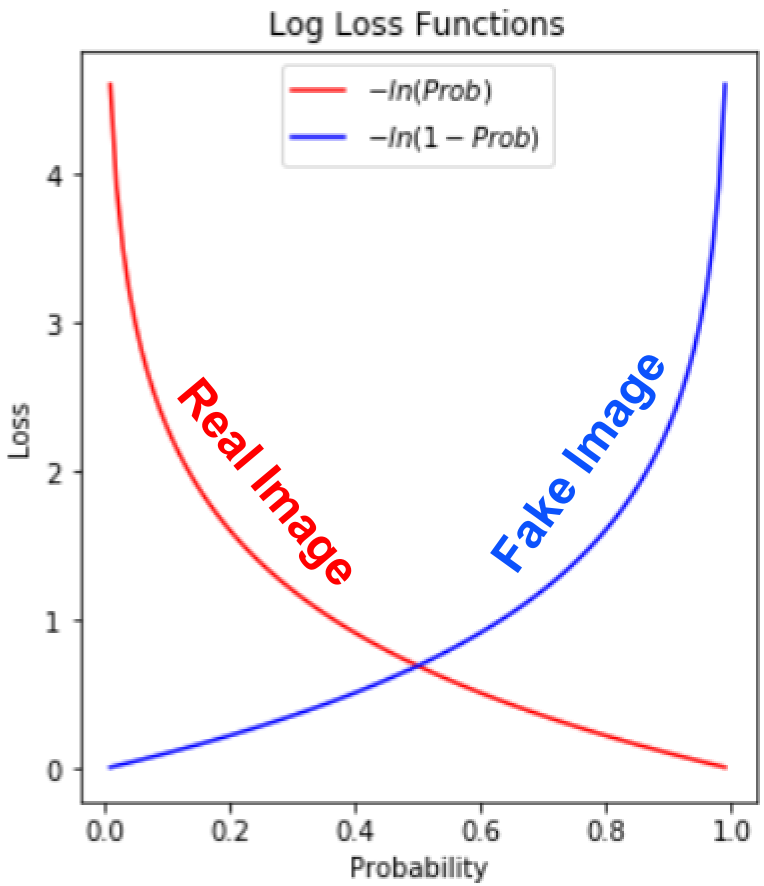
What if we have calculated a probability with our model?

For example, in generative adversarial models, the discriminator has a goal to predict a probability of real image accurate.

- Probability = 0.0 if image is fake
- Probability = 1.0 if image is real

We use negative log-loss to convert from probability to loss for each case.

- By minimizing this we are actually minimizing the cross-entropy or Kullback-Leibler Divergence (more later).



Negative log loss functions to map probability to loss.

Note: Kullback-Leibler Divergence (relative entropy) is statistical distance between distributions.

PGE 383

Tuning Hyperparameters

- **Cross Validation Workflows**

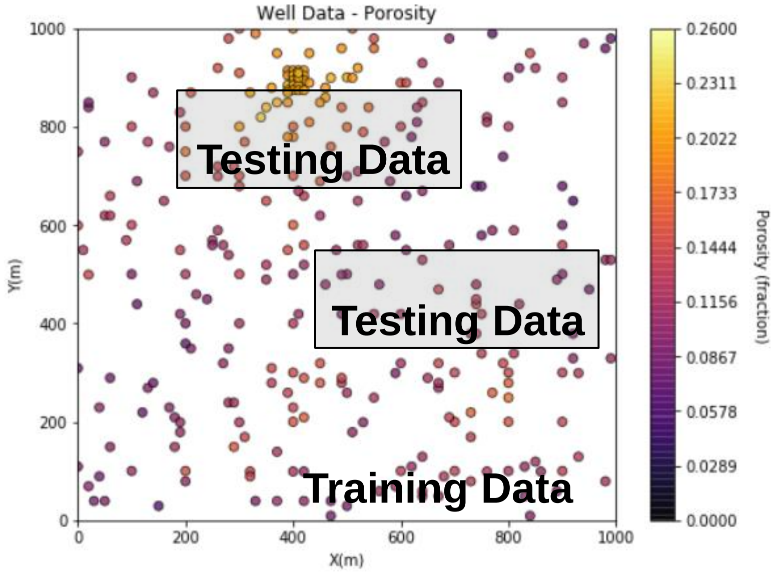
Michael Pyrcz, The University of Texas at Austin

Model Cross Validation Workflows

Cross Validation – Hold Out Method - The Standard Train – Test Workflow

- Withhold subset of the data during model training
- Then testing the trained model with withheld subset dataset
- Sensitive to the selection of testing, must make sure cross validation is fair
- Training data set (used for training), testing data set (withheld for testing)

Predictions in a distinctly different range of reservoir values.



Model Cross Validation Workflows

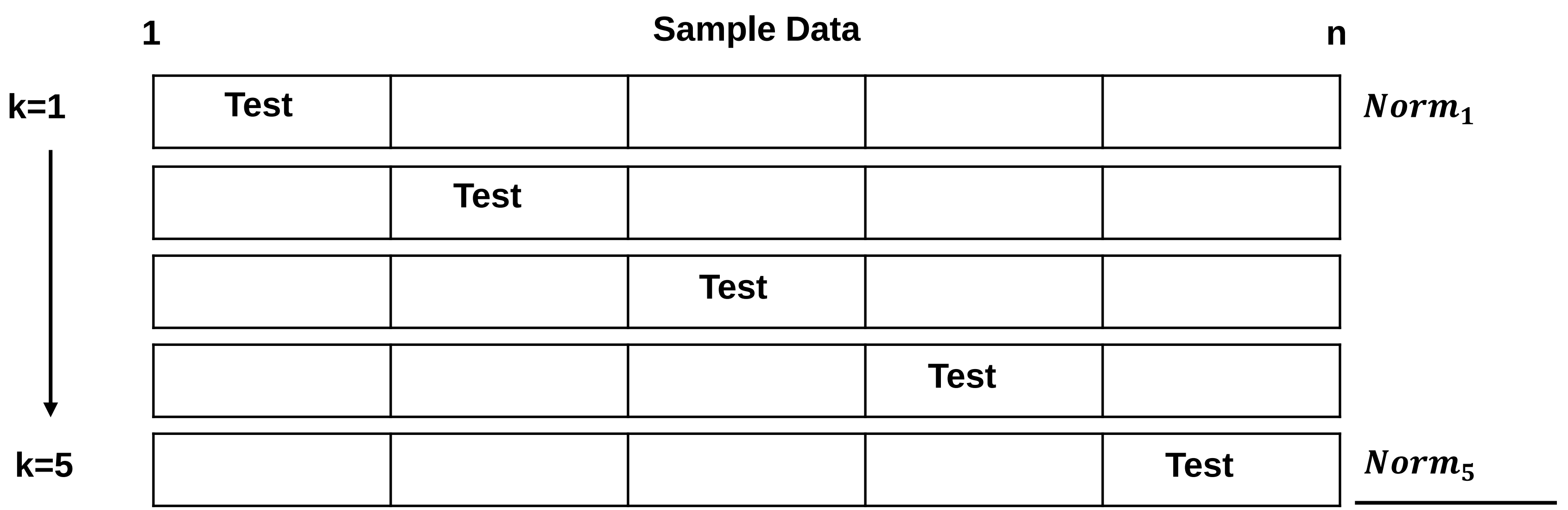
Leave-one-out Cross Validation (LOO CV)

- An exhaustive cross validation method.
- Combines training and testing into one step
- Loop over all data, withhold that data
  - Train on $n-1$ data and test on the withheld single data
  - Calculate model goodness metric
- Aggregate accuracy norm (single error) over all data, n
- Typically, too easy of an estimation problem
- K-fold is a more general and robust approach, this method assumes $\mathrm{K} = n$.

Model Cross Validation Workflows

k-fold Cross Validation (k-fold CV)

- A non-exhaustive cross validation approach.
- Select k, integer number of folds, impacts the size of testing set!
- Break data set into k subsets, equal size n/k
- Loop over k subsets:
  - use data outside the k subset to predict inside the k subset
  - calculate the model goodness metric
- Aggregate accuracy norms over all subsets, k



Cycle test over k folds, all other data are train.

$Norm\ Aggregation$

Model Cross Validation Workflows

Leave-p-out Cross Validation (LpO-CV)

- An exhaustive cross validation approach.
- For any p extract all possible p sized testing data sets.
- Train and calculate the accuracy norm with the withheld testing data
- Summarize over all possible p sized testing data sets.
- The possible number of datasets is $\binom{n}{p}$, n choose p. This is the binomial coefficient.

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}, \text{ where } n > p$$

- Aggregate accuracy norm over all combinations.

Model Cross Validation Workflows Limitations

Methods in the Python scikit-learn package:

sklearn.model_selection.train_test_split(X,y,train_size,random_state)

- random selection for training and testing
- specify train_size or test size (other is defaulted as compliment)
- use random_state for repeatability
- stratify will enforce whole sample statistics in the train and test sample subsets

sklearn.cross_val_score(model_object,X,y,cv,scoring)

- wrapper that performs k-fold cross validation with k random subsets
- k is the cv parameter
- scoring allows assignment of a custom model accuracy metric

Model Cross Validation Workflows Limitations

Methods in the Python scikit-learn package:

sklearn.cross_validate(model_object,X,y,cv,scoring)

- wrapper that performs k-fold cross validation with k random subsets
- k is the cv parameter
- scoring allows assignment of a custom model in training accuracy metric
- scoring parameter may be a list of model accuracy metrics
- outputs include fit and scoring times, estimates and all scores in a dictionary object

Model Cross Validation Workflows Limitations

Issues with Cross Validation

Peeking, information leakage – some information is transmitted from the withheld data into the model, some model decision(s) use all the data. Pipelines and wrappers help with this.

Fair Train and Test Split – many practitioners use random selection for train and test split (we use it, it is built into scikit-learn) and this may be too easy of a prediction problem

Black Swans / Stationarity – the model cannot be tested for data events not available in the data

This is also known as the '**No Free Lunch Theorem**' in machine learning

'even after the observation of the frequent or constant conjunction of objects, we have no reason to draw any inference concerning any object beyond those of which we have had experience' - Hume (1739–1740)

Model Cross Validation Workflows Limitations

Issues with Cross Validation

Subsurface Validation – it is not possible to validate open earth systems – Oreskes et al., 1994. Here's the abstract from their paper:

'Verification and validation of numerical models of natural systems is impossible. This is because natural systems are never closed and because model results are always nonunique. Models can be confirmed by the demonstration of agreement between observation and prediction, but confirmation is inherently partial. Complete confirmation is logically precluded by the fallacy of affirming the consequent and by incomplete access to natural phenomena. Models can only be evaluated in relative terms, and their predictive value is always open to question. The primary value of models is heuristic.'

Abstract from https://science.sciencemag.org/content/263/5147/641

Model Cross Validation Workflows Limitations

Issues with Cross Validation

‘All models are wrong, but some are useful’ – George Box

Parsimony – since all models are wrong, an economical description of the system. Occam’s Razor

Worrying Selectively – since all models are wrong, figure out what is most importantly wrong.

‘Be humble, the earth will surprise you!’ – me.

Concepts from https://en.wikipedia.org/wiki/All_models_are_wrong

PGE 383

Tuning Hyperparameters

- **Training and Testing**
- **Model Goodness Metrics**
- **Cross Validation Workflows**

Michael Pyrcz, The University of Texas at Austin