PGE 383
k-Nearest Neighbour

- Mapping in the Feature Space
- k-Nearest Neighbour
- k-Nearest Neighbour Example
- k-Nearest Neighbour Hands-on

Michael Pyrcz, The University of Texas at Austin

k-Nearest Neighbour Regression

Motivation to Cover this Method

- simple and interpretable
- linkage to variance-bias trade-off
- introduce our first hyperparameter
- very flexible, performs well in many situations

PGE 383

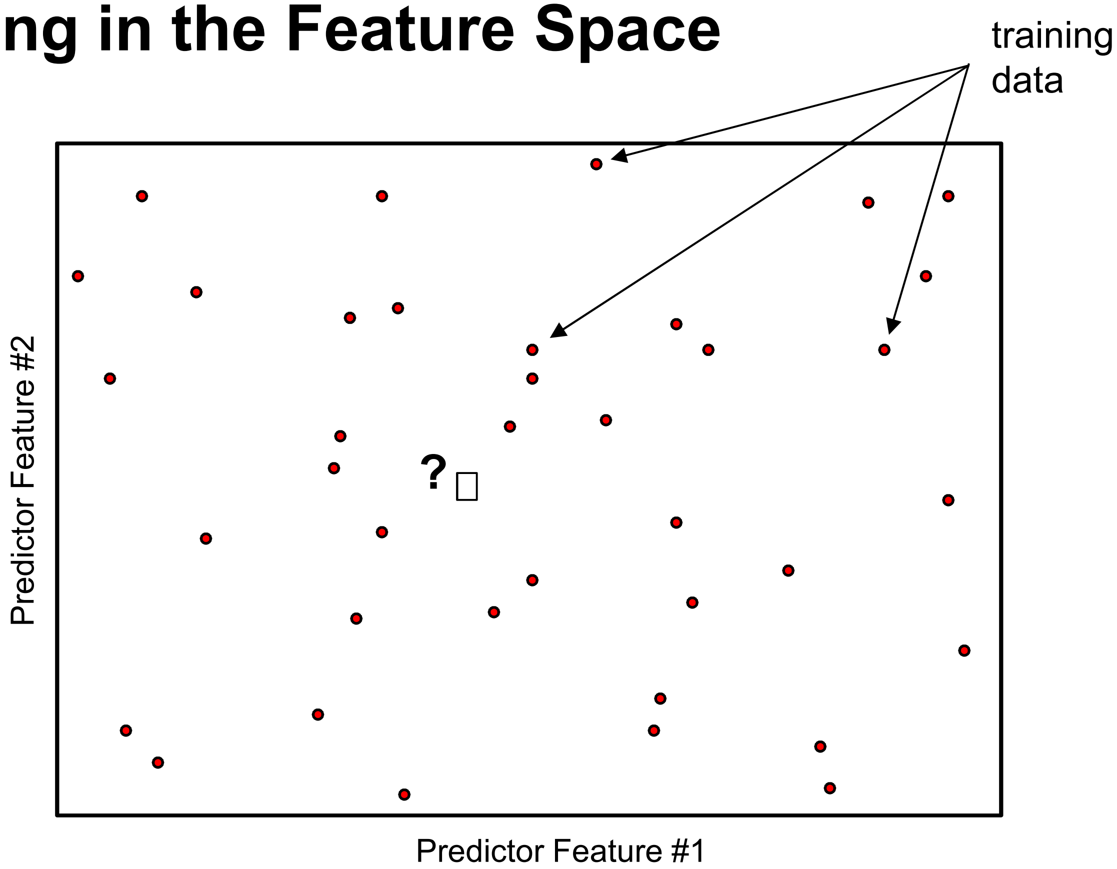
k-Nearest Neighbour

- **Mapping in the Feature Space**

Michael Pyrcz, The University of Texas at Austin

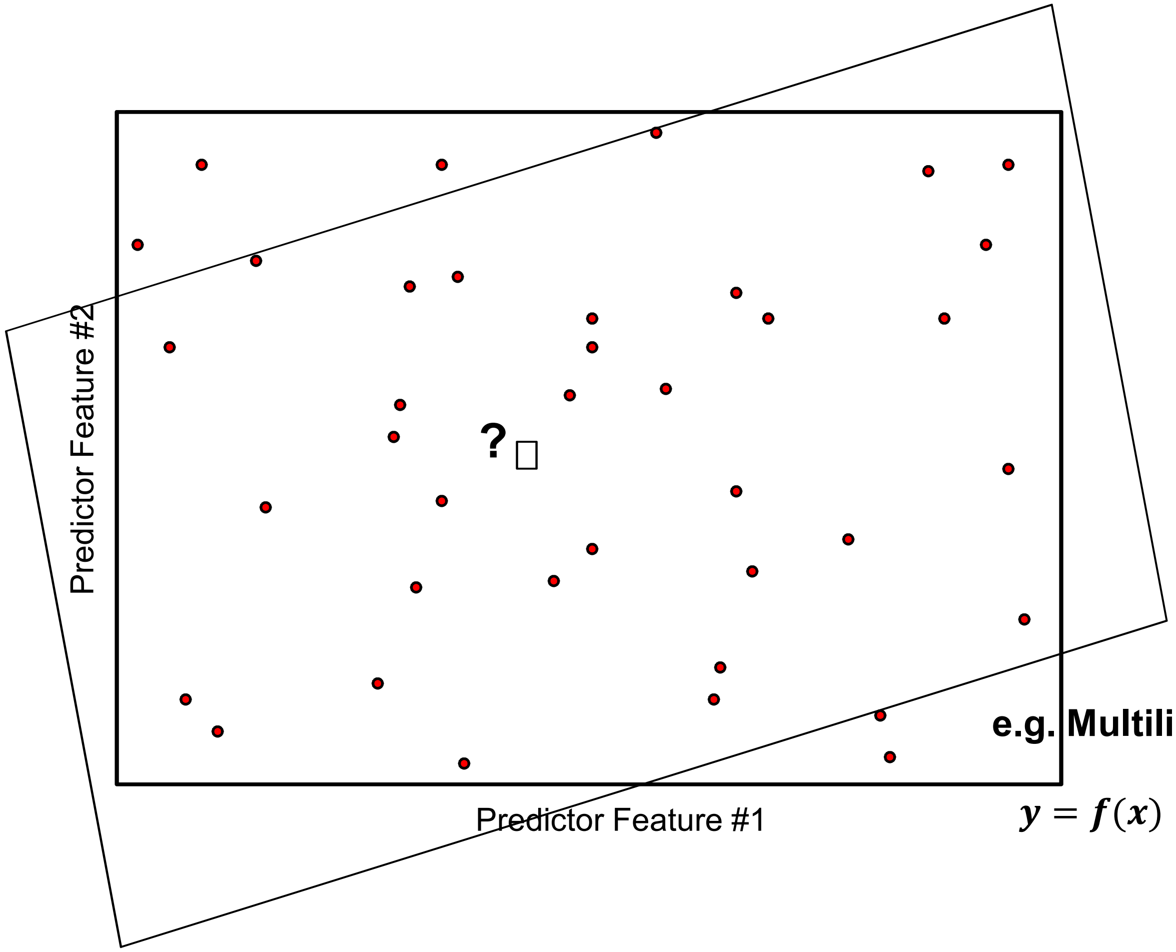
Mapping the Response in the Predictor Feature Space

- **Mapping in the Feature Space**



- We want to make predictions away from training data.

Mapping the Response in the Predictor Feature Space



e.g. Multilinear Regression

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \sum_{\alpha=1}^{m} \boldsymbol{b}_{\alpha} \boldsymbol{x}_{\alpha} + \boldsymbol{b}_0$$

- We could form a parametric model for $\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}(\boldsymbol{x})$.

Mapping the Response in the Predictor Feature Space

Recall: Parametric Methods

- make an assumption about the functional form, shape
- we gain simplicity and advantage of only a few parameters
- the model is generally compact and portable
- for example, here is a linear model

$$Y = f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m$$

- there is a risk that $\hat{f}$ is quite different than f, then we get a poor model!

Mapping the Response in the Predictor Feature Space

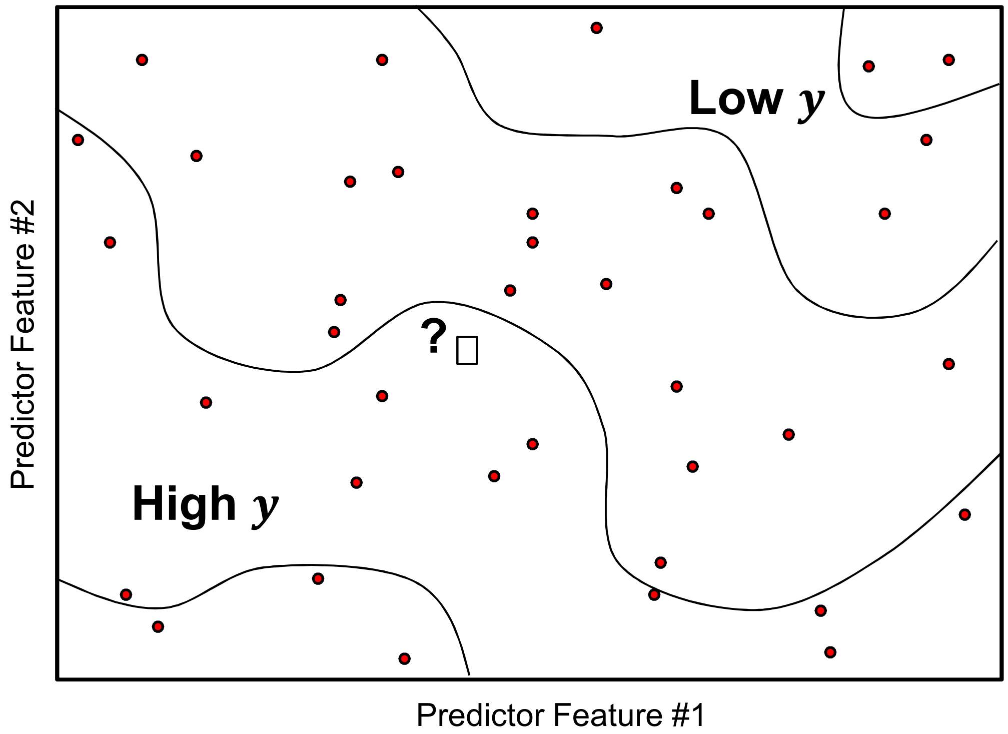
Recall: Nonparametric Methods

– make no assumption about the functional form, shape
– more flexibility to fit a variety of shapes for f
– less risk that $\hat{f}$ is a poor fit for f

– typically need a lot more data for an accurate estimate of f

‘Nonparametric is actually parametric rich!’

Mapping the Response in the Predictor Feature Space



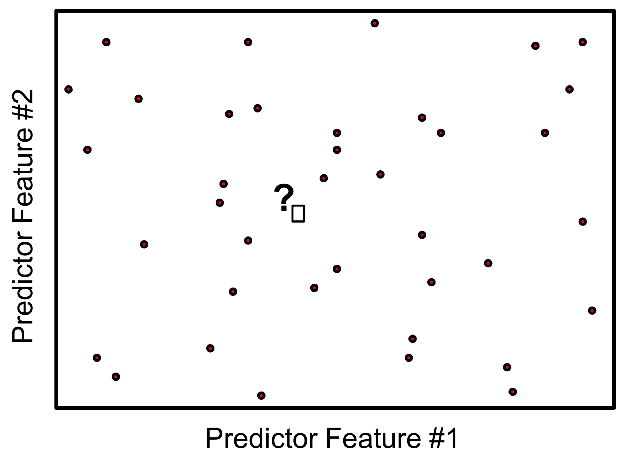
- We could interpolate the response feature in the predictor feature space!

PGE 383
k-Nearest Neighbour

- **k-Nearest Neighbour**

Michael Pyrcz, The University of Texas at Austin

Mapping the Response in the Predictor Feature Space



Possible methods for this interpolation:

- geostatistical, kriging
- inverse distance weighting
- moving window average / convolution

This is used for k-nearest neighbour

Convolution

Integral product of two functions

One interpretation, weighting

- weighting function, $f(\Delta)$, is applied to calculate the
- weighted average of function, $g(\mathrm{x}+\Delta)$

$$(\boldsymbol{f} * \boldsymbol{g})(\boldsymbol{x}) = \int_{-\infty}^{\infty} \boldsymbol{f}(\Delta)\mathrm{g}(\mathrm{x}+\Delta)\mathrm{d}\Delta$$

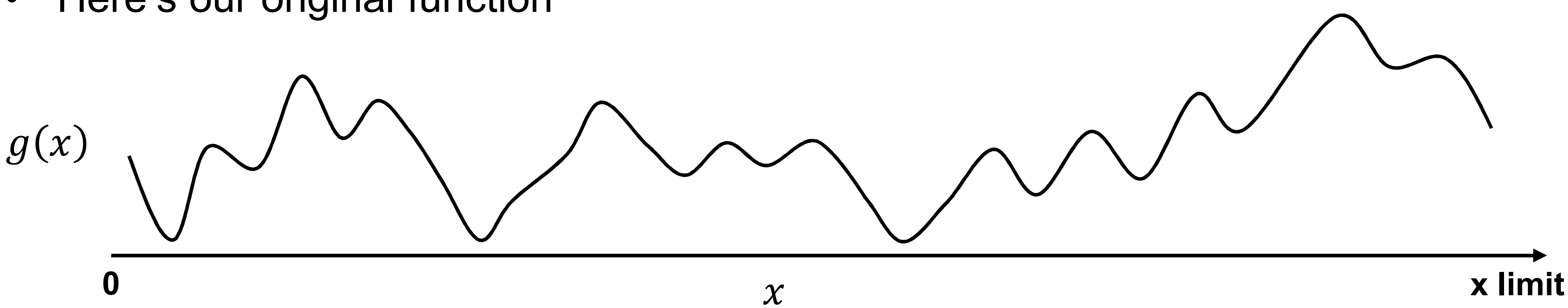
- this easily extends into multidimensional

$$(\boldsymbol{f} * \boldsymbol{g})(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{f}(\Delta_x, \Delta_y, \Delta_z)\mathrm{g}\left(\mathrm{x}+\Delta_x, \mathrm{y}+\Delta_y, \mathrm{z}+\Delta_z\right)\mathrm{d}\Delta_x \mathrm{d}\Delta_y \mathrm{d}\Delta_z$$

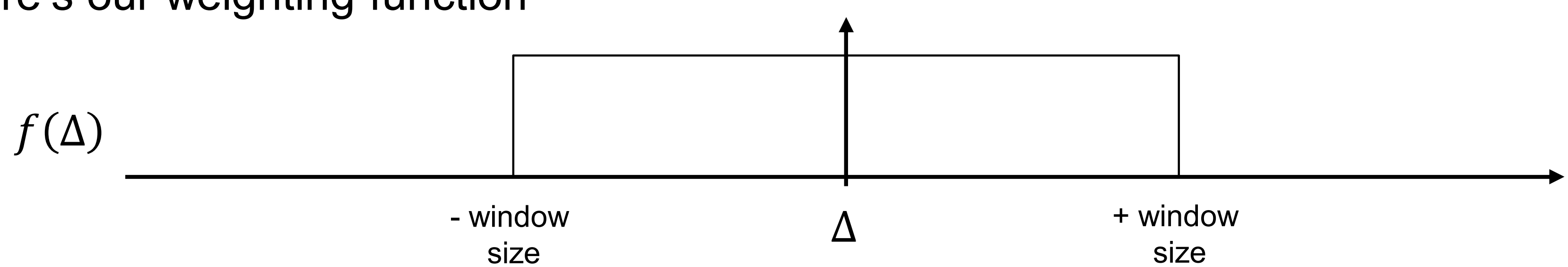
Convolution

Convolution explained graphically

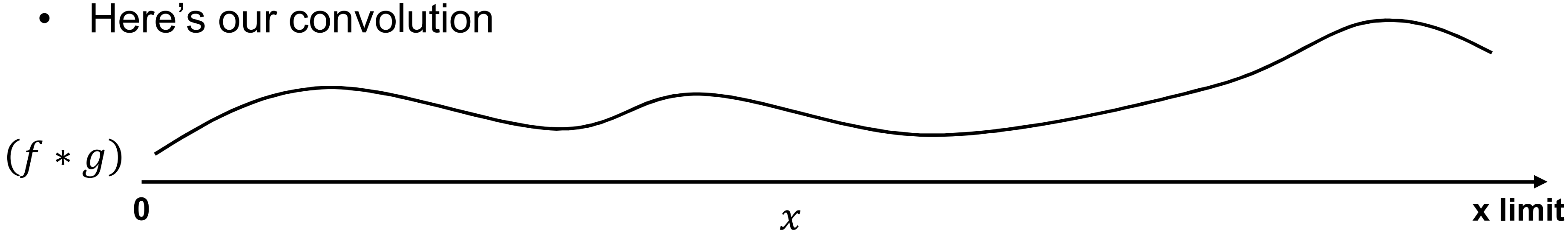
- Here's our original function



- Here's our weighting function



- Here's our convolution

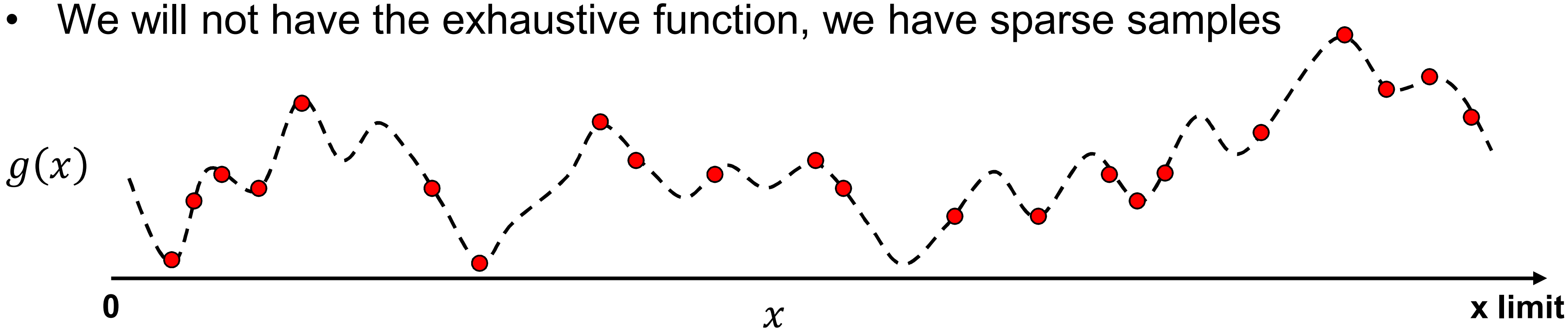


Uniform weighted moving window, window size 3m, 6m and 20m (left to right).

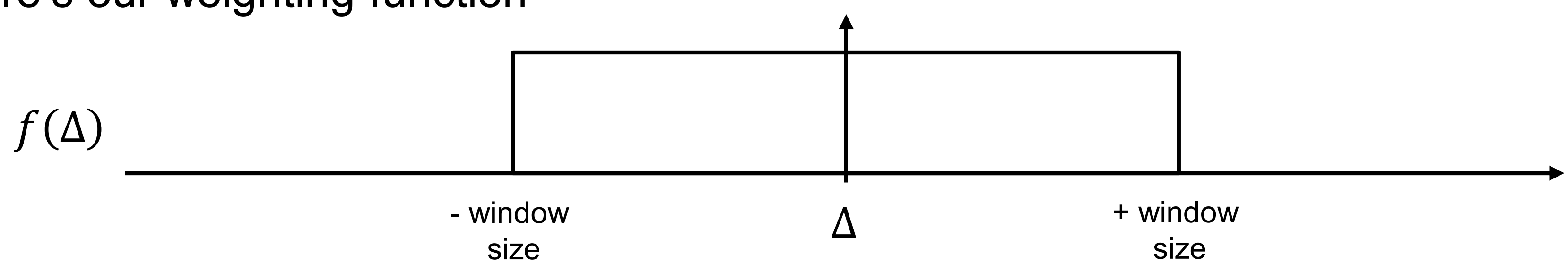
Convolution

Convolution explained graphically

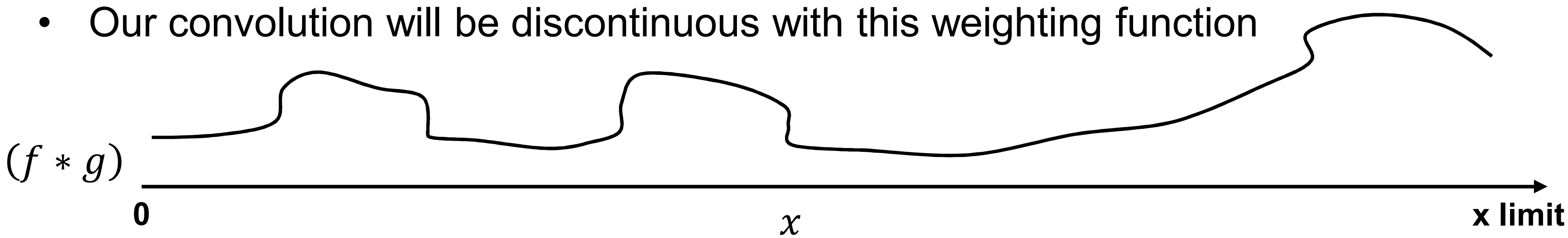
- We will not have the exhaustive function, we have sparse samples



- Here’s our weighting function



- Our convolution will be discontinuous with this weighting function

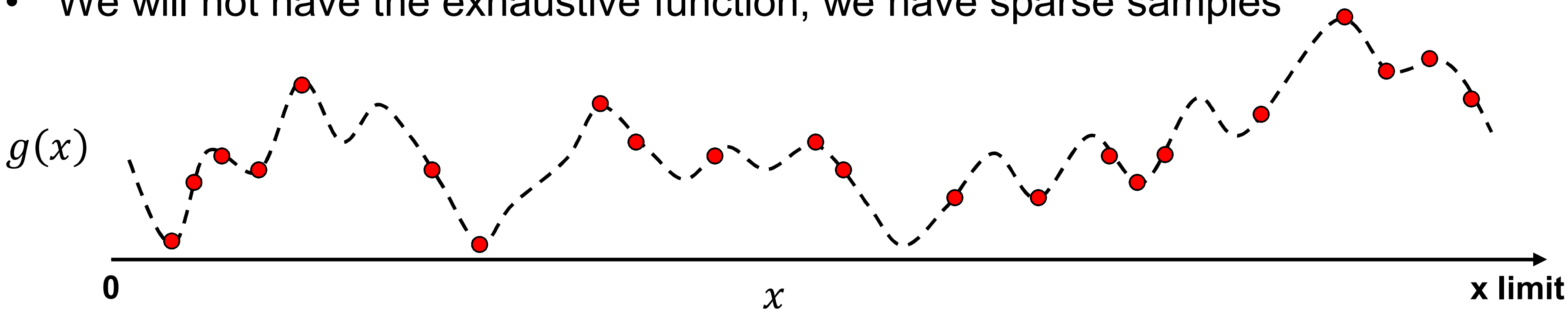


Sparse sampling convolution with uniform weighted moving window.

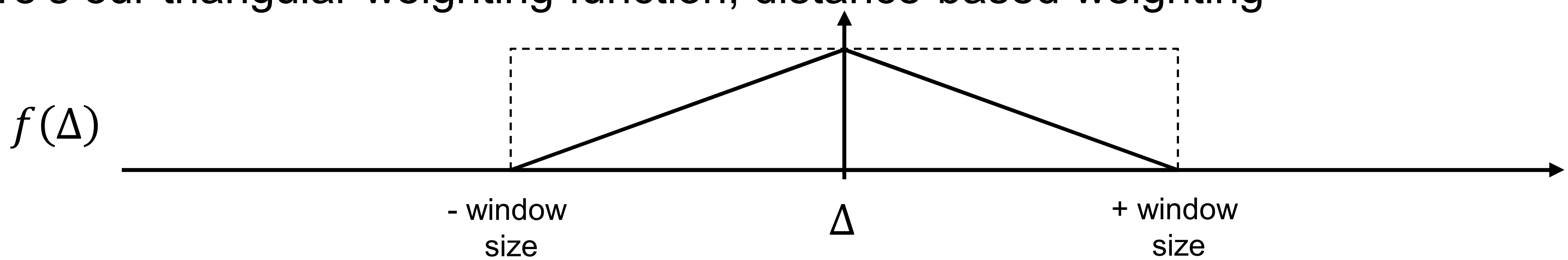
Convolution

Convolution explained graphically

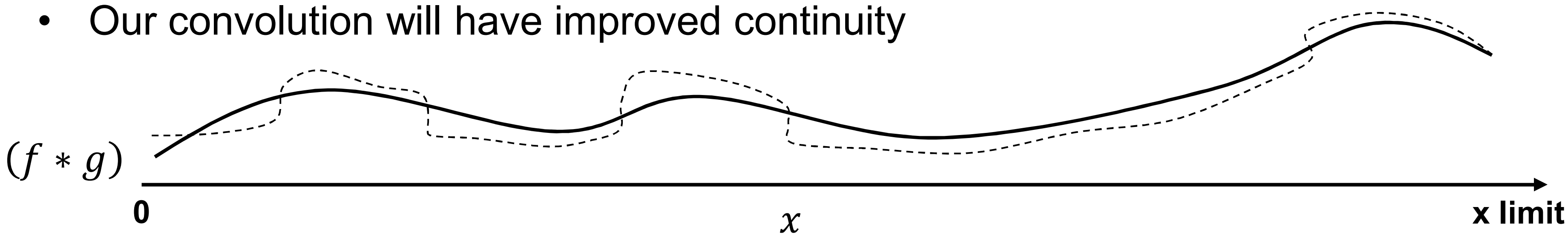
- We will not have the exhaustive function, we have sparse samples



- Here's our triangular weighting function, distance-based weighting



- Our convolution will have improved continuity



Sparse sample convolution with uniform and triangular weighted moving window.

Nearest Neighbours

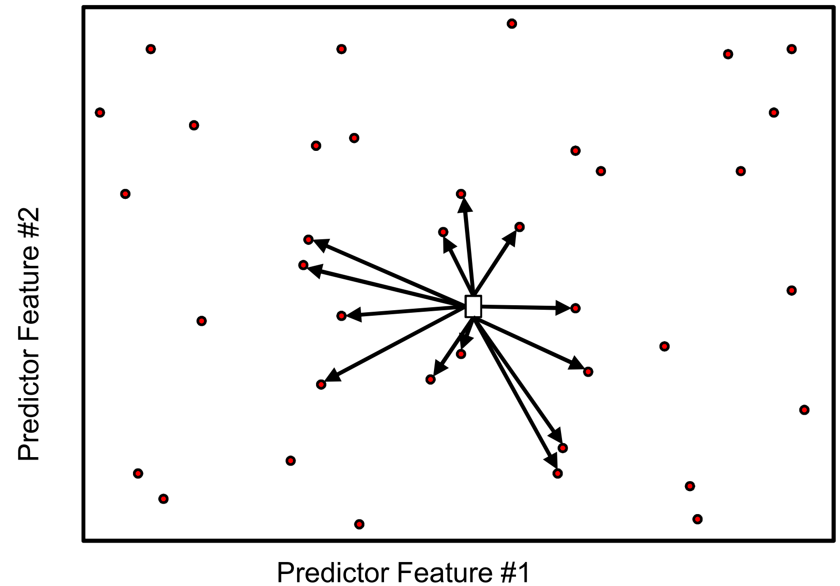
What are the nearest training samples?

We need to rank samples by proximity in feature space!

$$d_\alpha = \sqrt{\sum_{\alpha=1}^{m} (\Delta X_\alpha)^2}$$

note: there is an assumption that all the features have the same range of possible outcomes

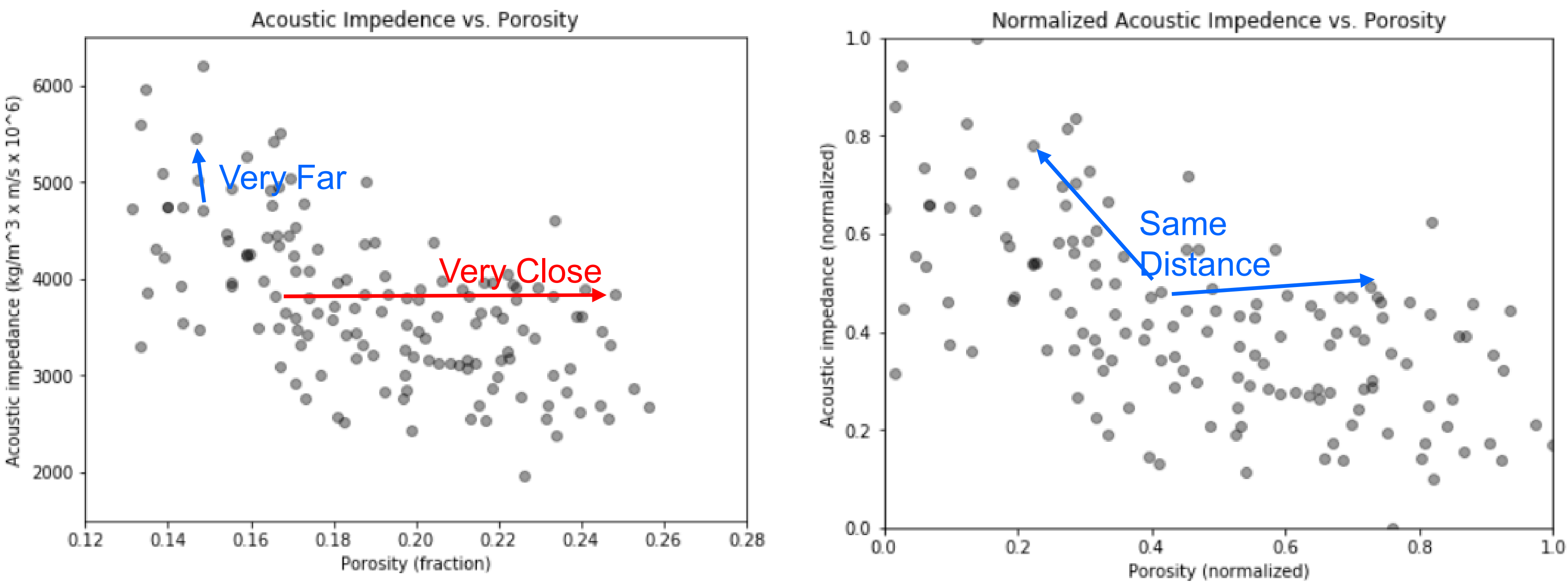
if not the case, features with lower ranges will have increased weight



Nearest Neighbours

We generally require some form of:

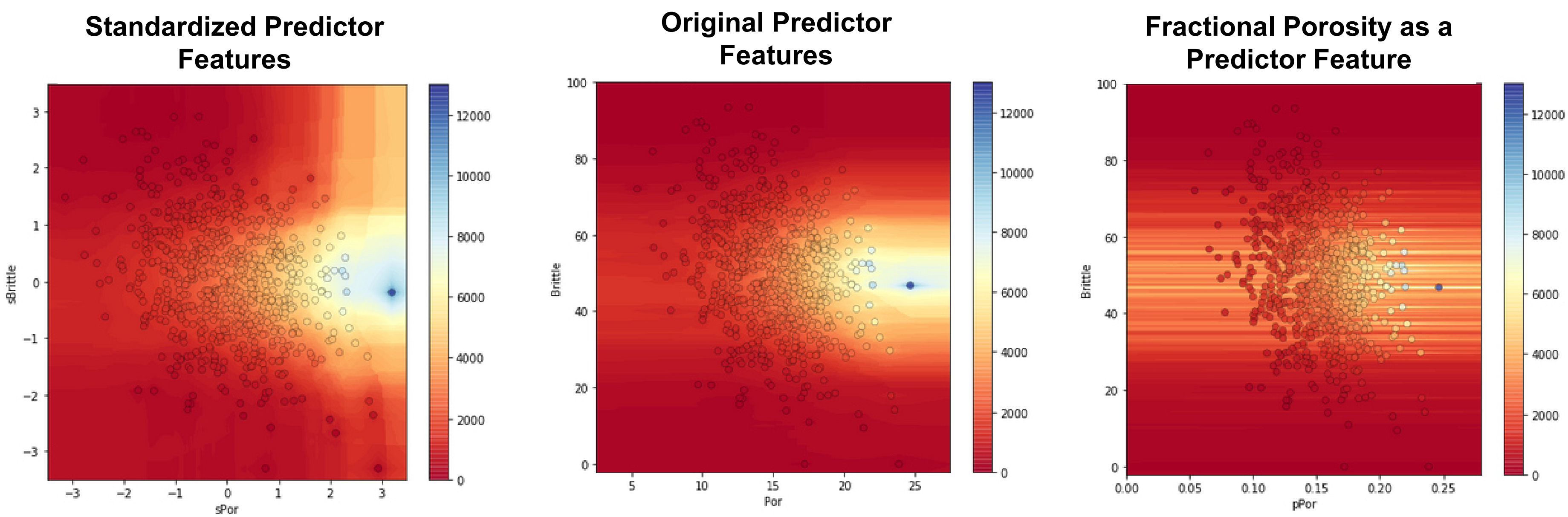
- **normalization** – constrain range [0,1] $$x_n = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$
- **standardization** – constrain the mean and variance $$x_s = \left(\frac{\sigma_{x_s}}{\sigma_x}\right)(x - \overline{x}) + \overline{x_s}$$



Nearest Neighbours

What are the nearest training samples?

Here's three examples of k-nearest neighbour prediction models for production.



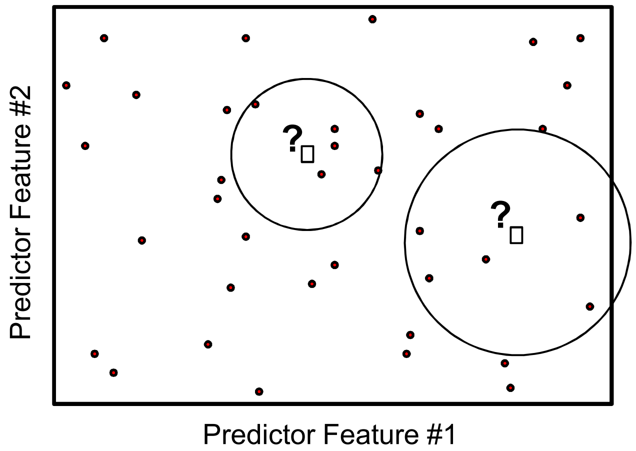
k-nearest neighbour prediction of unconventional well production from porosity and brittleness.

Mapping the Response in the Predictor Feature Space

k-nearest neighbor regression with moving window average / convolution

Hyperparameter, number of nearest data *k*

- size of the window / how many nearest data to include
- k-nearest neighbor is a locally adaptive search
- sparse sampled will require a larger window
- larger k results in smoother response prediction → underfit
- smaller k results in more detailed response prediction → overfit

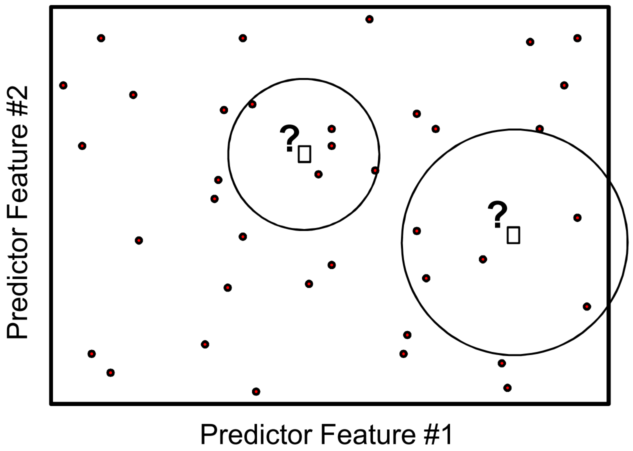


Mapping the Response in the Predictor Feature Space

k-nearest neighbor regression with moving window average / convolution

Hyperparameter, weighting function form

- there are generally 2 parametric forms available for the weighting function
- **uniform** – insensitive to distance of training data from estimated location
- **inverse distance weighting** with a power specified



PGE 383 Lecture xx
k-Nearest Neighbour

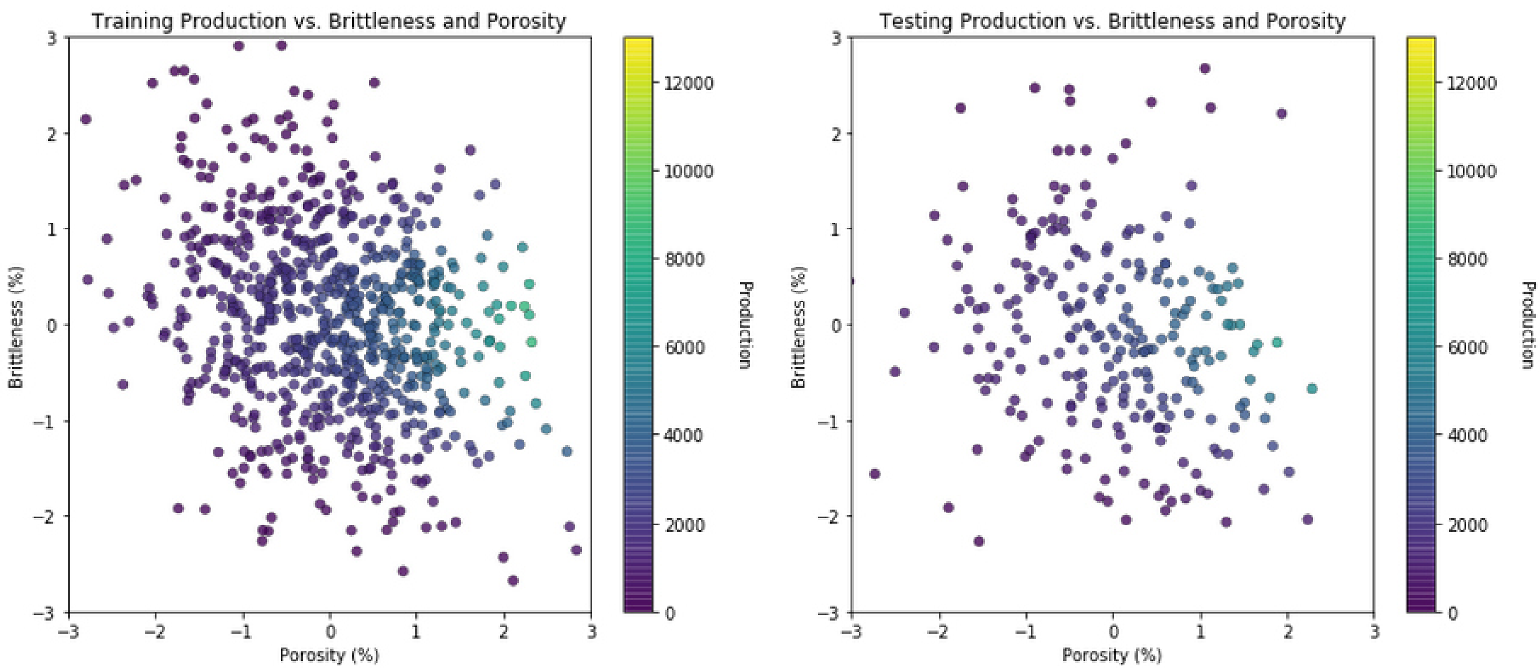
- **k-Nearest Neighbour Example**

Michael Pyrcz, The University of Texas at Austin

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:

$$prod = f(porosity, bittleness)$$

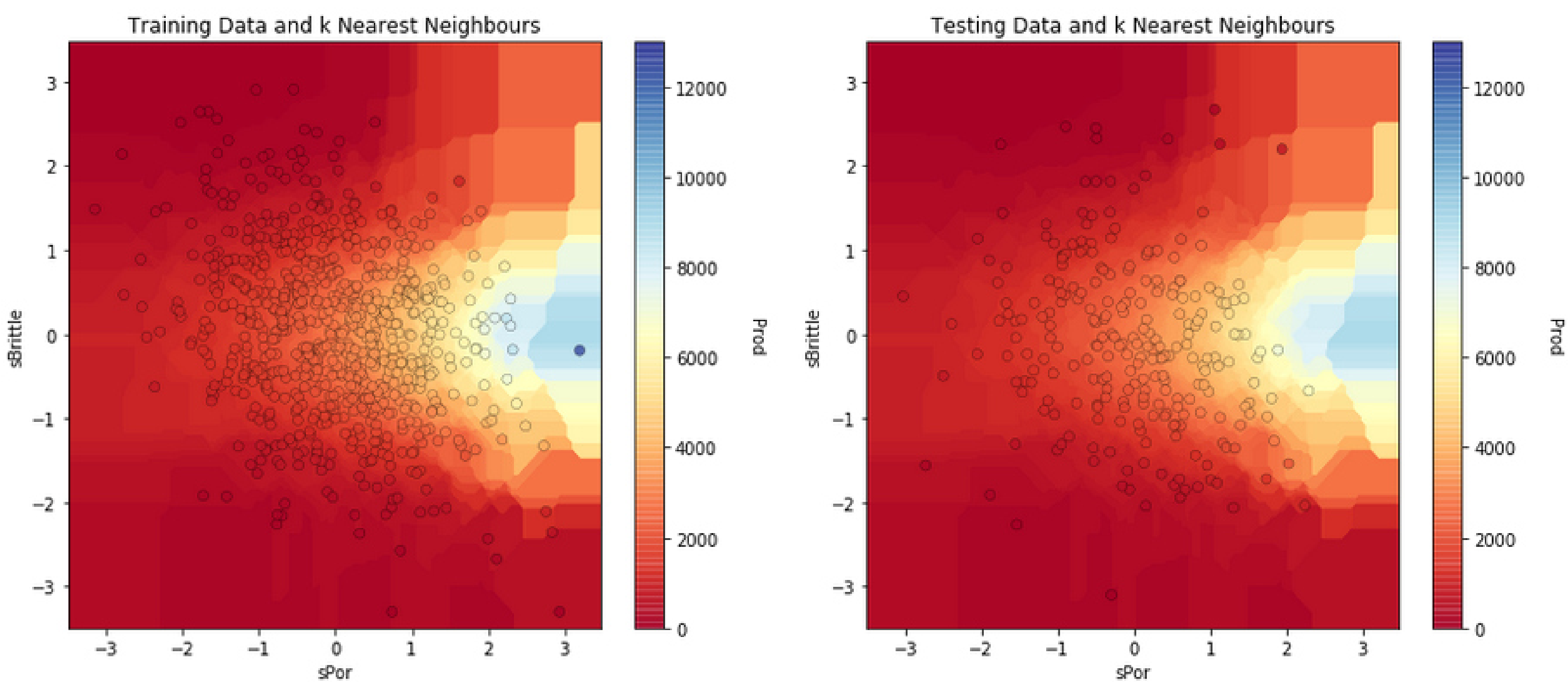


Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:

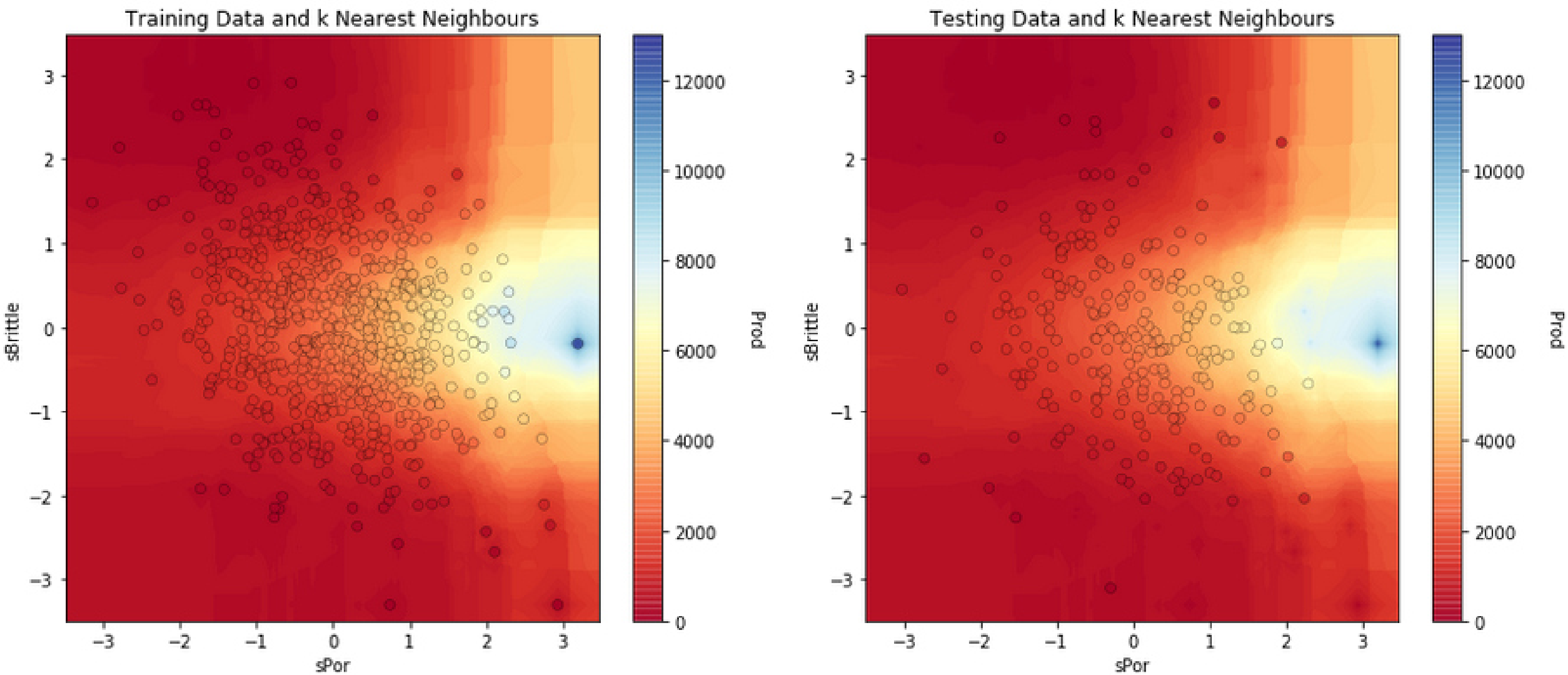
- uniform weights, 5 nearest neighbours, standardized predictor features



Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:
- distance weighted, 15 nearest neighbours, standardized predictor features

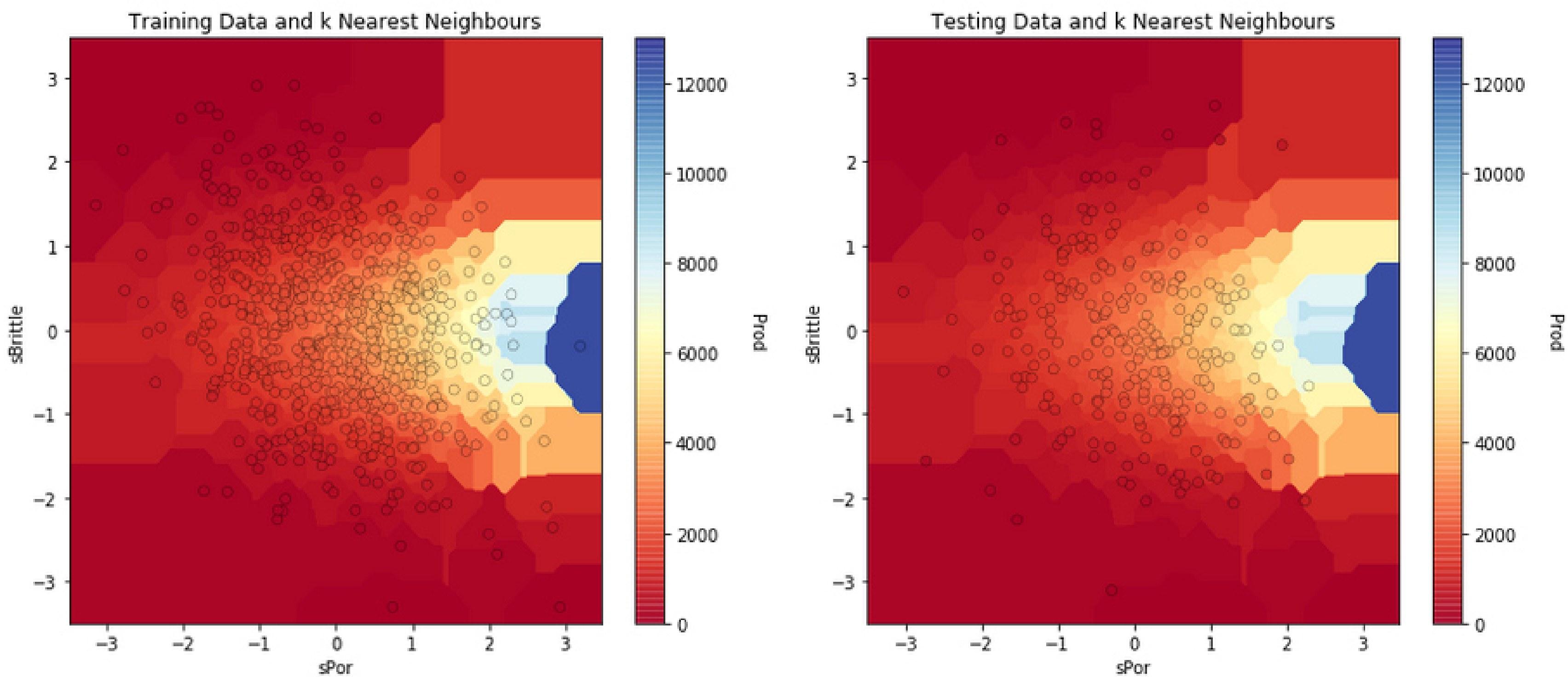


Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:

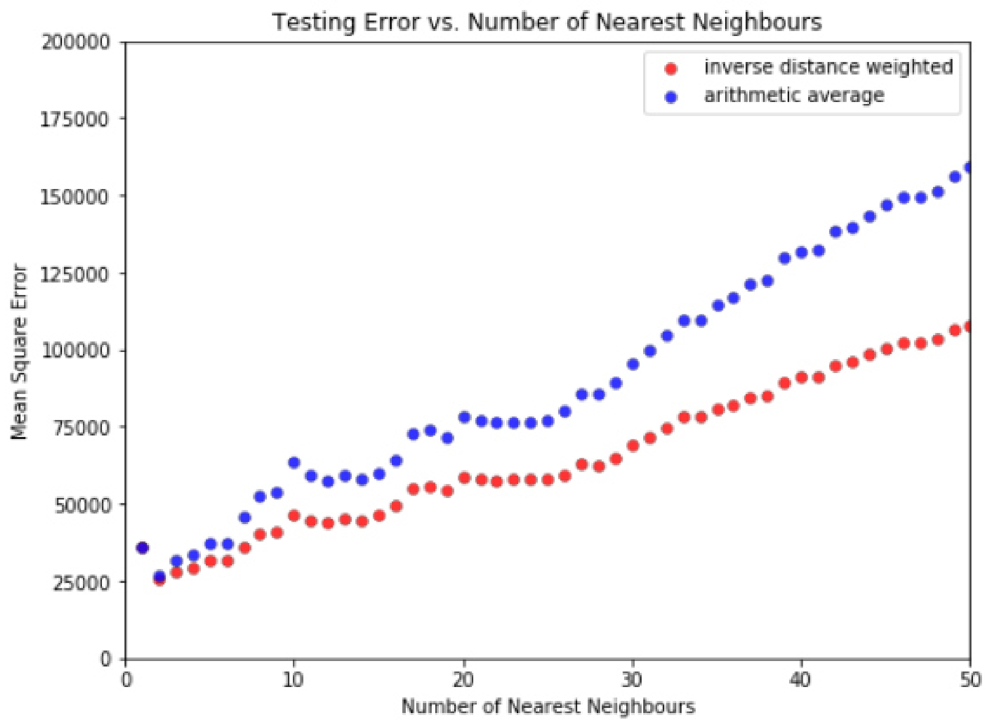
- uniform weights, 1 nearest neighbours, standardized predictor features



Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:
- hyperparameter tuning with Jackknife



Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

k Nearest Neighbour Example

Prediction of unconventional production rates (MCFPD) from:

- hyperparameter tuning with k-fold cross validation

Prediction problem, production rate (MCFPD) from porosity (standardized) and brittleness (standardized).

PGE 383 Lecture xx
k-Nearest Neighbour

- **k-Nearest Neighbour Hands-on**

Michael Pyrcz, The University of Texas at Austin

k-nearest Neighbour Demonstration

Demonstration workflow with k-nearest neighbour regression for prediction.

Subsurface Machine Learning with k Nearest Neighbours

k Nearest Neighbours for Multivariate Modeling for Subsurface Modeling in Python

Michael Pyrcz, Associate Professor, University of Texas at Austin

Twitter | GitHub | Website | GoogleScholar | Book | YouTube | LinkedIn | GeostatsPy

PGE 383 Exercise: k Nearest Neighbours for Subsurface Modeling in Python

Here's a simple workflow, demonstration of k nearest neighbours for subsurface modeling workflows. This should help you get started with building subsurface models that data analytics and machine learning. Here's some basic details about K nearest neighbours.

K Nearest Neighbours

Machine learning method for supervised learning for classification and regression analysis. Here are some key aspects of k nearest neighbours.

Prediction

- non-parametric method for regression and classification
- a function $\hat{f}$ of the nearest k training data in predictor feature space such that we predict a response feature Y from a set of predictor features $X_1, \ldots, X_m$.
- the prediction is of the form $\hat{Y} = \hat{f}(X_1, \ldots, X_m)$
- for classification the majority response category among the k nearest training data is selected as the prediction
- for regression the average (or other weighted average, like inverse distance weighted) of the response features among the k nearest training data is assigned as the prediction

File SubsurfaceDataAnalytics_kNearestNeighbour.ipynb at https://git.io/fjinq.

PGE 383 Lecture xx
k-Nearest Neighbour

- Mapping in the Feature Space
- k-Nearest Neighbour
- k-Nearest Neighbour Example
- k-Nearest Neighbour Hands-on

Michael Pyrcz, The University of Texas at Austin