Geostatistics and Machine Learning
Support Vector Machines

- Support Vector Machines
- SVM Demonstration in Python

Michael Pyrcz, The University of Texas at Austin

Geostatistics and Machine Learning
Support Vector Machines

- Support Vector Machines

Introduction

Data Analytics

Inferential Methods

Predictive Methods

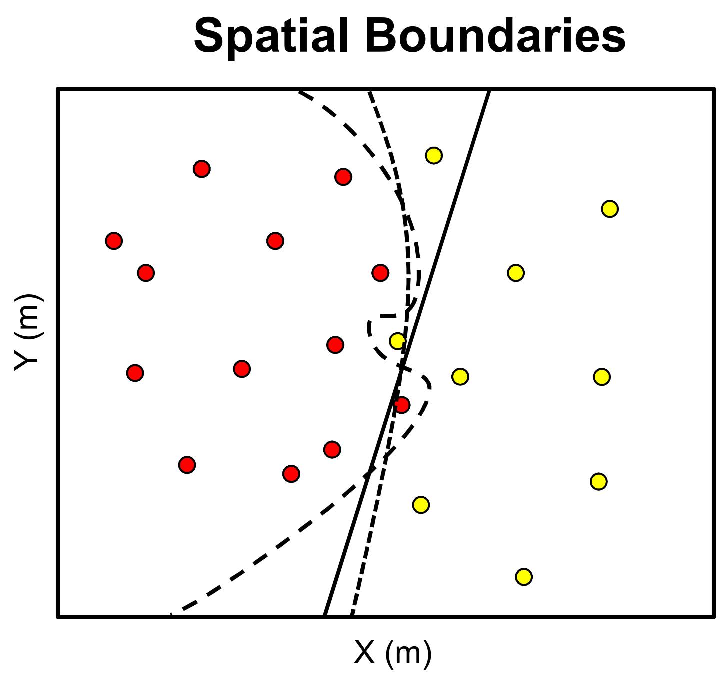
Advanced Methods

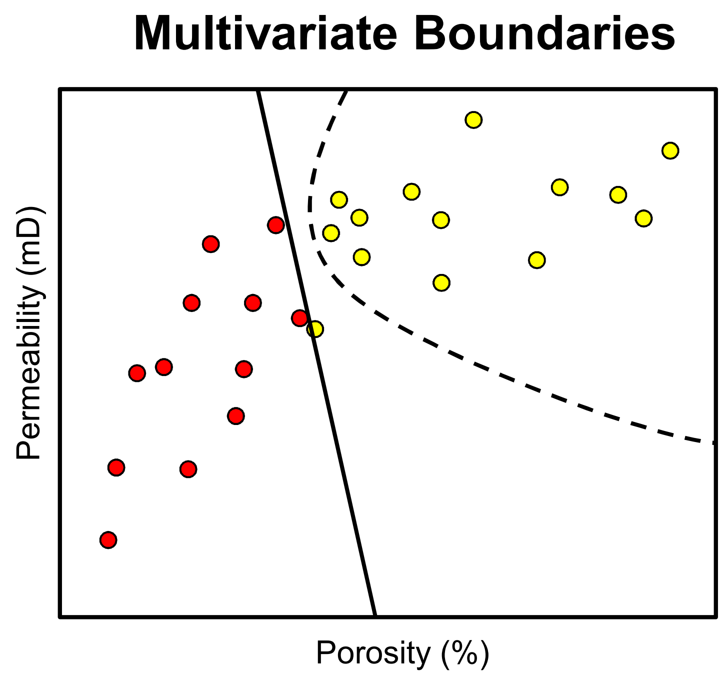
Conclusions

Michael Pyrcz, The University of Texas at Austin

Support Vector Machines

- A powerful supervised training, machine learning method for segmentation
- For example, forming a rule to segment a multivariate dataset into multiple categories with a decision rule.
- E.g. Geoscience Kanevski et al., 2000, Bio Goovaerts et al., 2018





Support Vector Machines

General Comments on Support Vector Machines

- Generalization of linear decision boundaries for classification
- Applied for classification with optimal separating hyperplanes when the classes overlap.
- Nonlinearity is achieved by transforming the feature space (typically to a high dimensional space)
- Maximizes the margin, separation between cases in each category
- Training data well within the decision boundary have no influence

Support Vector Machines

1. Form a boundary with the largest possible **margin** between the different cases.
2. Data within the margin or misclassified update the model, they are called **support vectors**.
3. Project into problem into a higher dimensional space to solve linearly, with a variety of **kernels**.
4. The **C** parameter controls to penalty of misclassification, high C will result in a more complicated model (lower model bias, higher model variance).

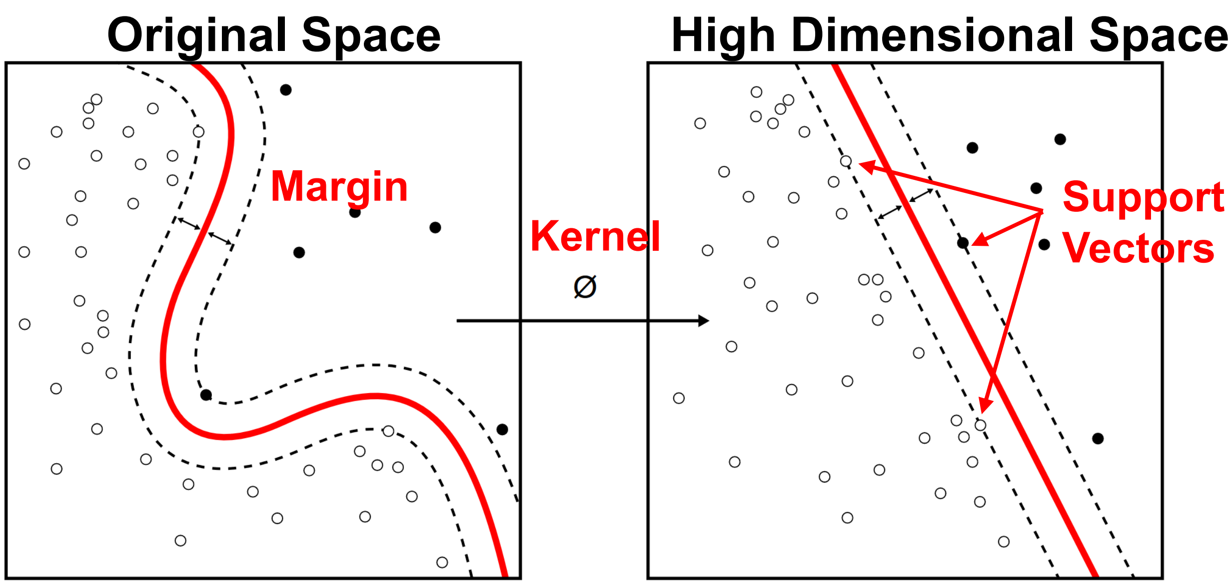


Image by Zirguezi, available at https://en.wikipedia.org/wiki/Support-vector_machine#/media/File:Kernel_Machine.svg

Support Vector Machines

The Support Vector Machine model:

Solve for the hyperplane:

$$f(x) = x^T\beta + \beta_0$$

$f(x)$ is the signed distance from the boundary, $-$ one side and $+$ the other.

$$G(x) = sign\big(f(x)\big)$$

where x is an vector, $x_j,\ \ j = 1, \ldots, m$ predictor features

Therefore the constraint all data must be on the correct side of the boundary would be represented by:

$$y_i\left(x_i^T\beta + \beta_0\right) \geq 0$$

since y_i is the response feature with categories -1 or $+1$.

Support Vector Machines

The Margin Concept

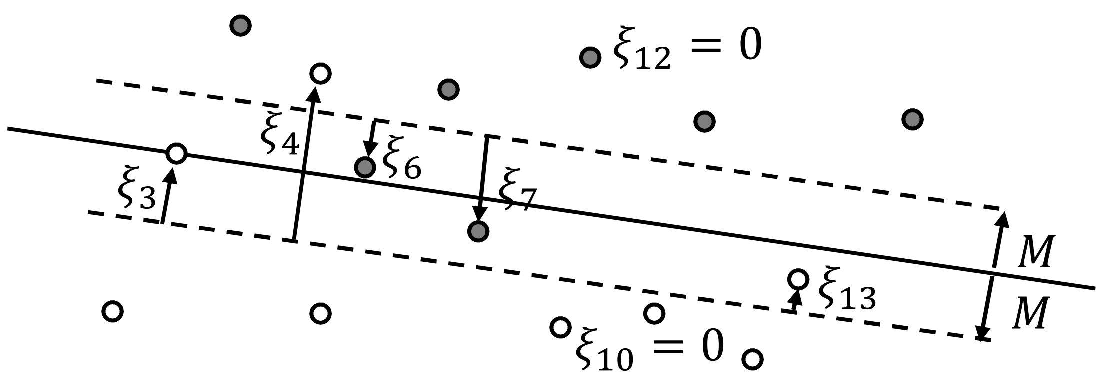
When the training data are overlapping it would not be possible nor desirable to develop a decision boundary that perfectly separates the categories for which this condition would hold:

$$y_i\left(x_i^T\beta + \beta_0\right) \geq 0$$

We need a model that allows for some misclassification.

$$y_i\left(x_i^T\beta + \beta_0\right) \geq M - \xi_i$$

We introduce the concept of a margin and a distance from the margin.



Support Vector Machines

Solving Support Vector Machines

This may be solved as a linear quadratic optimization:

define $M = 1/\|\beta\|$ then we can describe our system as:

$$min\|\beta\| \text{ subject to } \begin{cases} y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i \quad \forall\, i \\ \xi_i \geq 0, \sum \xi_i \leq constant \end{cases}$$

Observations:

- Training data well on the correct side of the boundary have no influence
- Training data within the margin or on the incorrect side of the boundary influence the boundary and are know as support vector machines.

Support Vector Machines

Solving Support Vector Machines

Quadratic with linear inequality constraint – convex optimization problem

We re-express the previous relationship as

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i$$

subject to $\xi_i \geq 0, \;\; y_i\left(x_i^T \beta + \beta_0\right) \geq 1 - \xi_i$

Recall $M = 1/\|\beta\|$, therefore we are:

- maximizing the margin, M and the sum of the distances from the margin $\sum_{i=1}^{N} \xi_i$ weight by a cost parameter, C.

Support Vector Machines

Basis Expansion

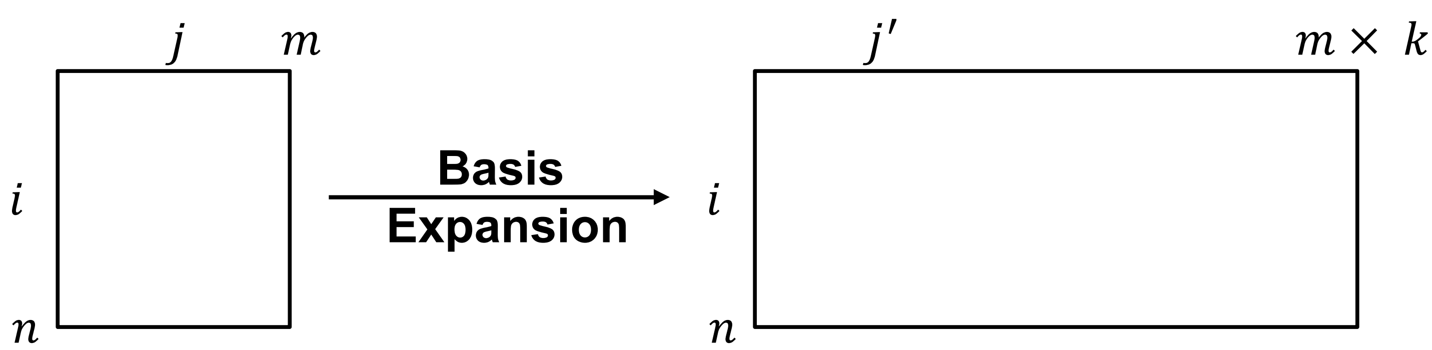
- To add flexibility, capture non-linearity in our model
- May be applied in regression, classification etc.

Basis Expansion: transform to a linearly independent set of objects (functions)

$$h(x_i) = \left(h_1(x_i),\, h_2(x_i), \ldots,\, h_M(x_i)\right)$$

Here's an example for polygonal expansion

$$h_{i,1}(x_i) = x_i, \quad h_{i,2}(x_i) = x_i^2, \quad h_{i,3}(x_i) = x_i^3, \; h_{i,4}(x_i) = x_i^4, \ldots, \; h_{i,k}(x_i) = x_i^k$$



Support Vector Machines

The Kernel Trick

We can incorporate our basis expansion in our method without actually every needing to transform to:

$$h(x)$$

We only need the inner projects:

$$h(x)h(x')^T = \langle h(x), h(x') \rangle$$

Instead of the actual values in the transformed space, we just need the ‘similarity’ between all available training values.

Support Vector Machines

The Kernel Trick

For example, let's assume a 2 features and 2nd order polynomial basis function

$$h: x \rightarrow h(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

we have expanded our linear system to include the 2nd order terms x_1^2, x_2^2 and the product term $\sqrt{2}x_1x_2$ over our 2 predictor features for a more complicated, nonlinear boundary.

But it can be shown that we don't need to use $h(x)$, just the kernel as:

$$h(x)h(x')^T = k(x, x') = (x^T x' + 1)^d$$

A function of base features, x, only.

Support Vector Machines

The Kernel Trick

Let's return to our model in terms of support vectors, y_n, and the Kernel, $K()$.

$$G(x) = sign(h(x)^T\beta + \beta_0)$$

Given:

$$\beta = \sum \alpha_n y_n h(x)$$

We have:

$$G(x) = sign(\alpha_n y_n h(x)^T h(x) + \beta_0) \rightarrow sign\left(\sum \alpha_n y_n K(x_n, x) + \beta_0\right)$$

We can solve for β_0 as:

$$\beta_0 = y_m - \sum \alpha_n y_n K(x_n, x_m)$$

Support Vector Machines

Other Kernel Transforms

We can substitute any kernel to map our problem to a higher dimensional space. We could in fact design our own under these conditions.

1. $k(x, x')$ is symmetric
2. k_{matrix} is positive semi-definite

$$k_{matrix} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_1) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}$$

Of course, there are various kernel ready for us to work with:

- Linear: $k(x_m, x_n) = x_m^T x_n$
- Radial Basis Function: $k(x_m, x_n) = e^{-\frac{\|x_m - x_n\|^2}{2\sigma^2}} = e^{-\gamma\|x_m - x_n\|^2}$
  - a $h(x)$ with infinite dimensionality
  - where $\|x_m - x_n\|^2$ is the squared Euclidean distance between feature vectors.
- New kernel, new support vector machine!

Support Vector Machines

Alternative Interpretation of Our Support Vector Machine

Our classifier is the linear weighted sum of a ‘similarity’ measure between the support vectors, x_n, and the new location, x

$$G(x) = sign\left(\sum_{\alpha_n > 0} \alpha_n y_n K(x_n, x) + \beta\right)$$

Geostatistics and Machine Learning
Support Vector Machines

- **SVM Demonstration in Python**

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Michael Pyrcz, The University of Texas at Austin

Support Vector Machines Demonstration

Demonstration workflow with support vector machines to form a decision rule for segmentation.

Subsurface Data Analytics

Support Vector Machine for Multivariate Segmentation of Facies in Python

Wendi Liu, Michael Pyrcz, University of Texas at Austin

Workflow Goals

Learn the basics of support vector machine in python to segment facies given petrophysical properties. This includes:

- Loading and visualizing sample data
- Trying out support vector machine with different kernels (linear, polynomial, radial basis function)
- Tuning the SVM model parameters and results evaluation

Objective

I want to provide hands-on experience with building subsurface modeling workflows. Python provides an excellent vehicle to accomplish this.

The objective is to remove the hurdles of subsurface modeling workflow construction by providing building blocks and sufficient examples. This is not a coding class per se, but we need the ability to 'script' workflows working with numerical methods.

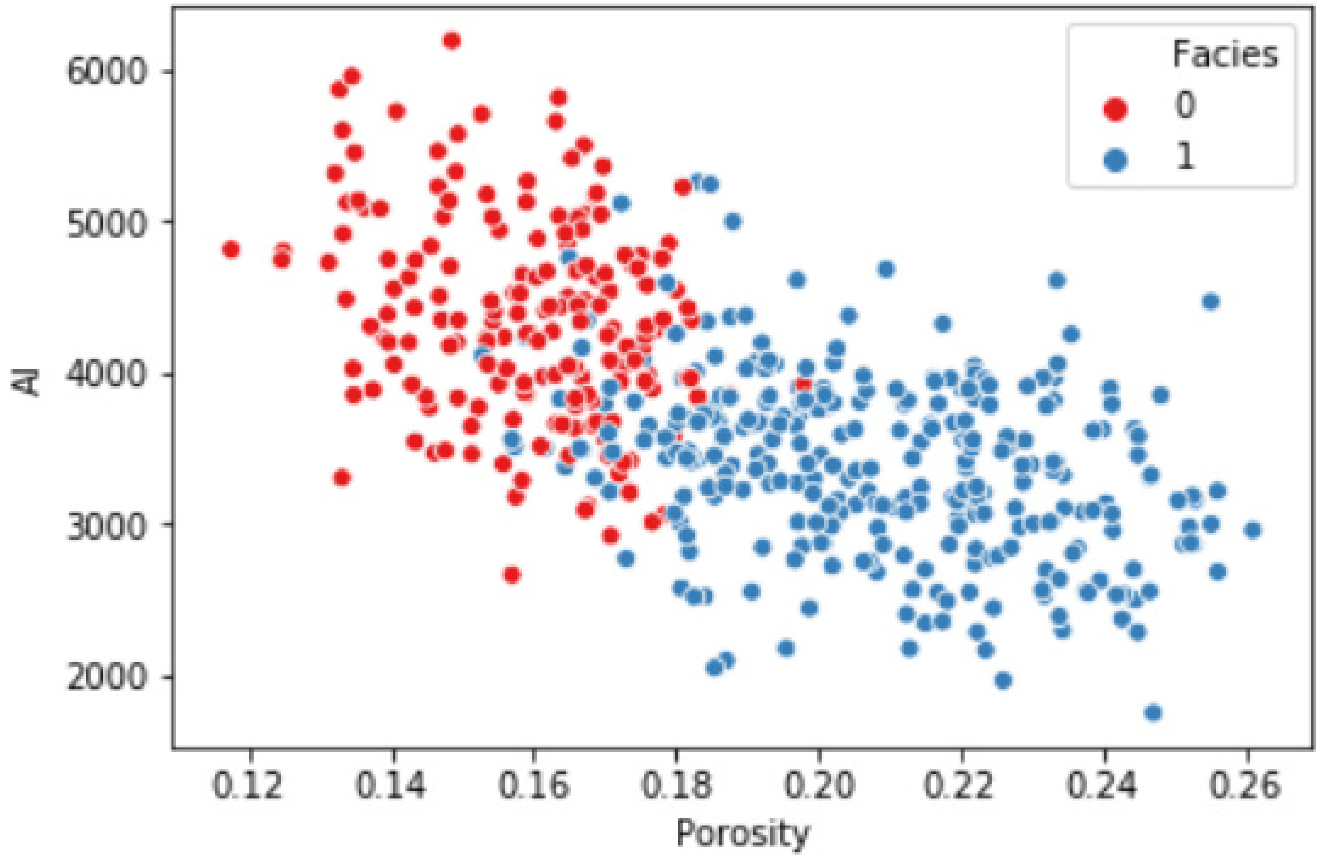
Load the required libraries and functions

The following code **imports** the required libraries. After we excute this code we can use 'os', 'np', 'pd' to access functionality in each of these libraries.

Image by Zirguezi, available at https://en.wikipedia.org/wiki/Support-vector_machine#/media/File:Kernel_Machine.svg

Support Vector Machines Demonstration

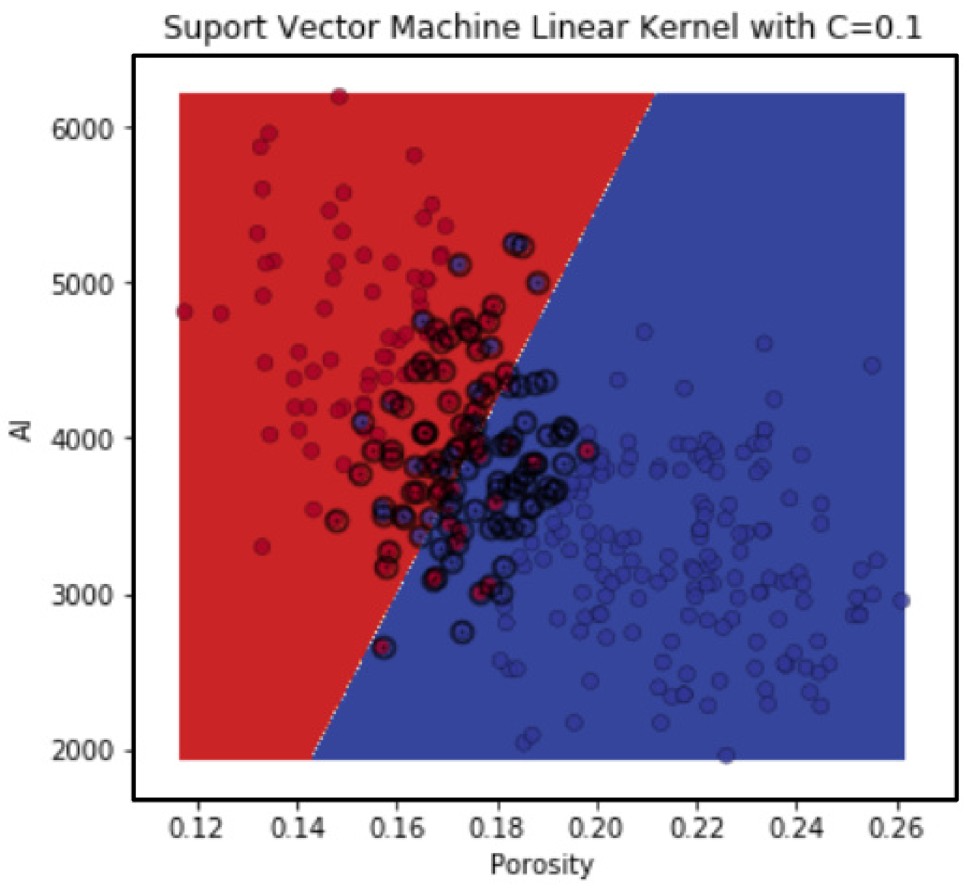
Demonstration workflow with support vector machines to form a decision rule for segmentation. Here's the training data.



Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines Demonstration

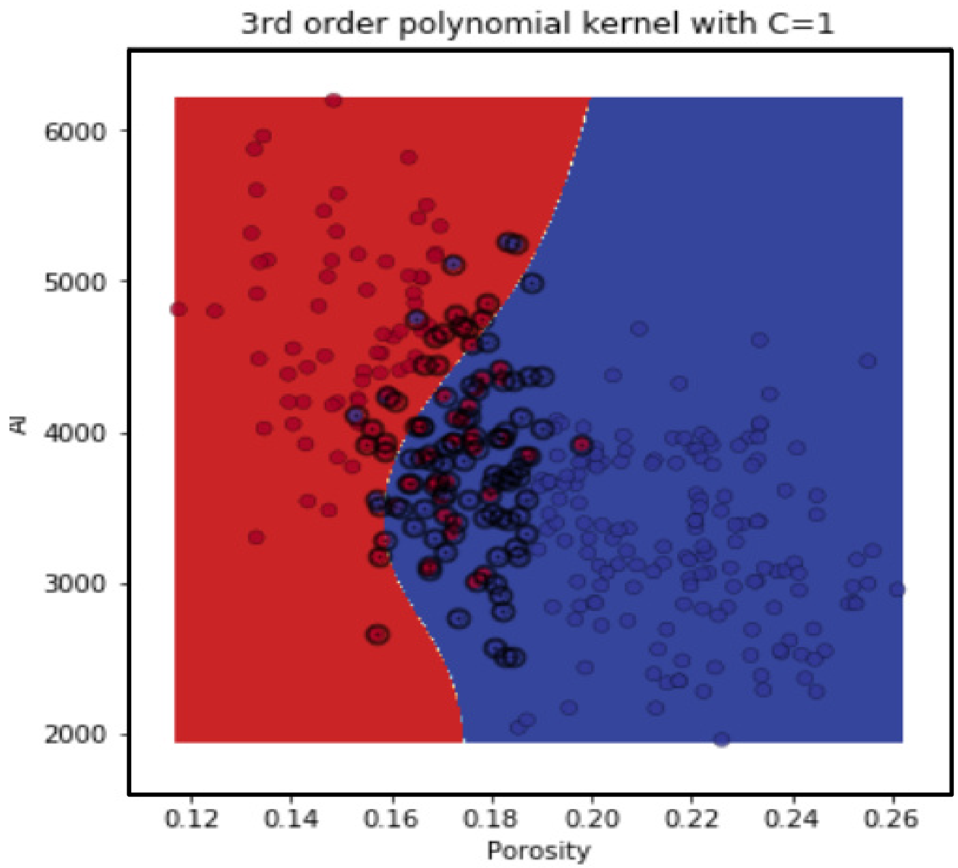
Linear kernel – decision boundary and support vectors highlighted.



Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines Demonstration

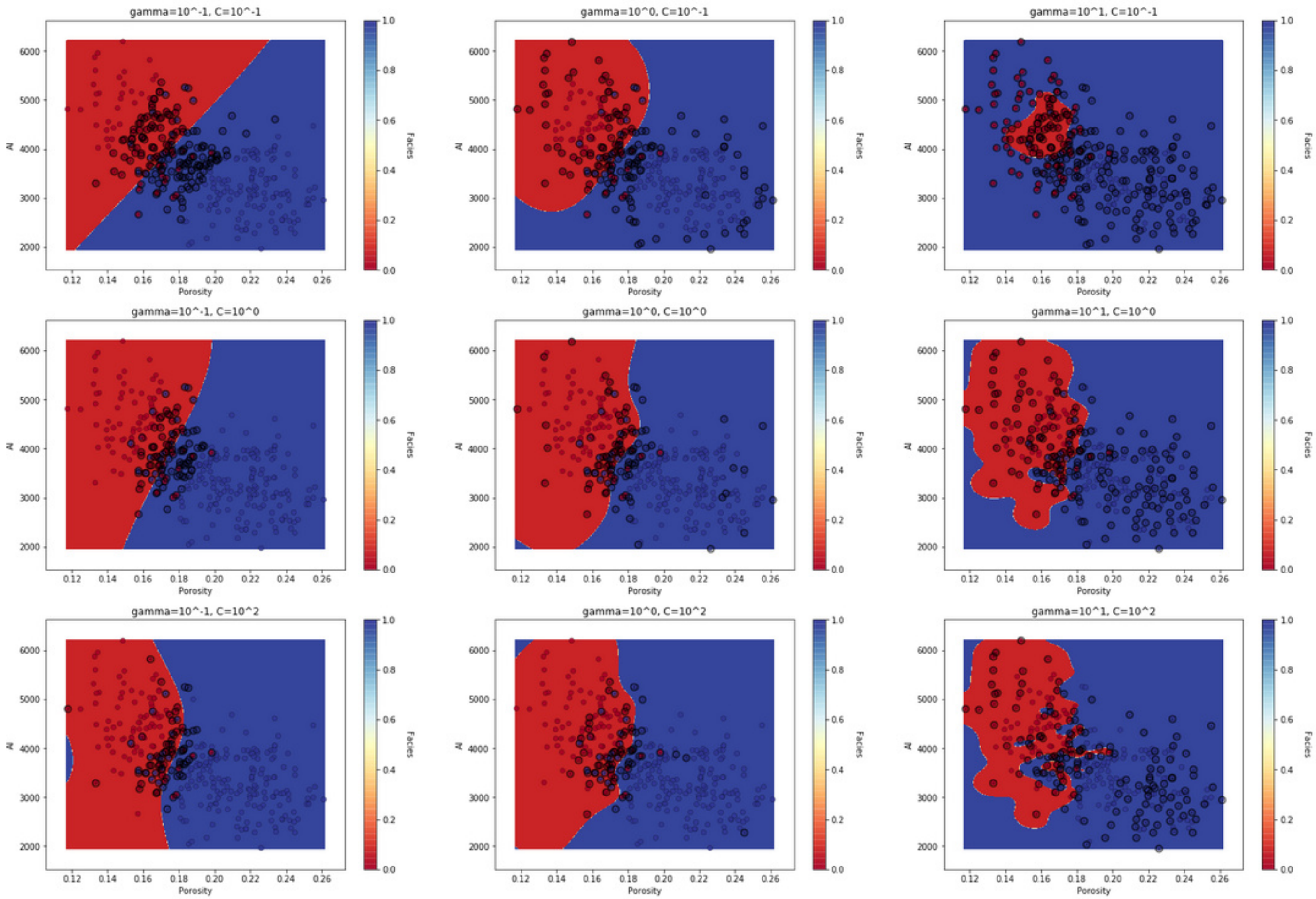
3rd Order Polynomial Kernel – decision boundary and support vectors highlighted.



Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines Demonstration

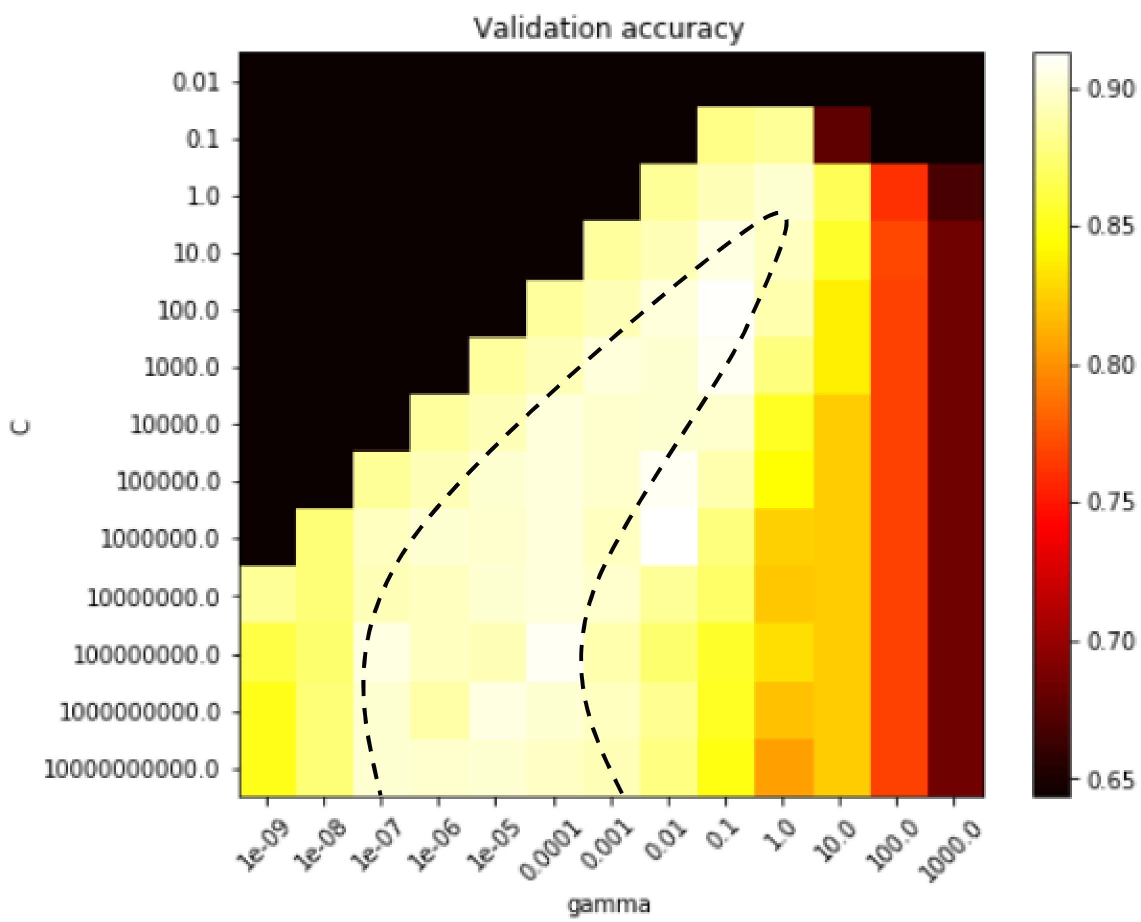
Radial Kernel – control of C and curvature parameter, γ.



Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines Demonstration

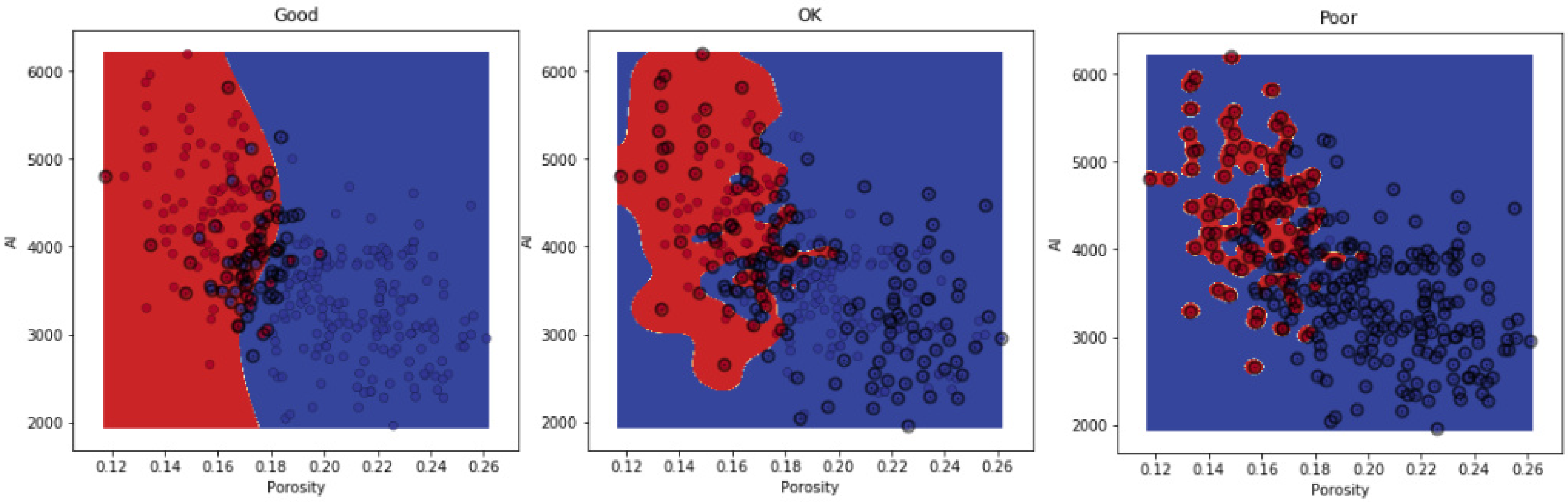
Performance with testing for a wide variety of parameters.



Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines Demonstration

Examples of models that perform well, average and poorly in testing.



- A clear case of overfit.

Workflow developed by Wendi Liu, PhD student at The University of Texas at Austin.

Support Vector Machines

- Generalization of linear decision boundaries for classification
- $Y = f(X_1, \ldots, X_m) + \epsilon$
- Optimal separating hyperplanes when the classes overlap.
- Nonlinearity is achieved by transforming the feature space (typically to a high dimensional space)
- Maximizes the margin, separation between cases in each category

Geostatistics and Machine Learning
Support Vector Machines

- Support Vector Machines
- SVM Demonstration in Python

Michael Pyrcz, The University of Texas at Austin