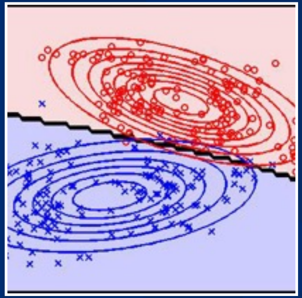
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

k Nearest Neighbors
Parzen Window

Mei-Ching Chen, PhD
Ben Rodriguez, PhD

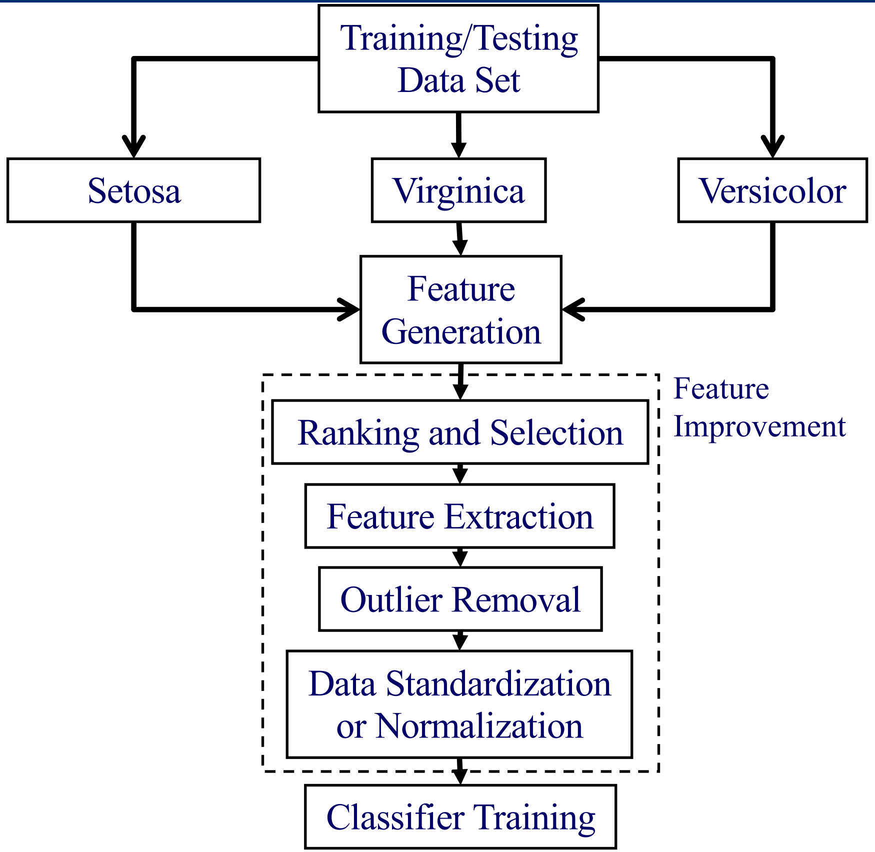


Overview

- ML Detection System
- Classifier Training
- Simple Visual Classification Examples
- K Nearest Neighbor
- Parzen Window Example
- Kernel Functions
- Transforming Data

Machine Learning Detection System

- Iris Data Set
  - Flower Type
- Feature Generation
  - Petal length, Petal width, sepal length, sepal width
- Feature Improvement
  - Feature Ranking and Selection
  - Feature Extraction
  - Outlier Removal
  - Data Standardization and Normalization
- Classifiers
  - Bayes with Gaussian Mixture Models
  - K Nearest Neighbor
  - Parzen window



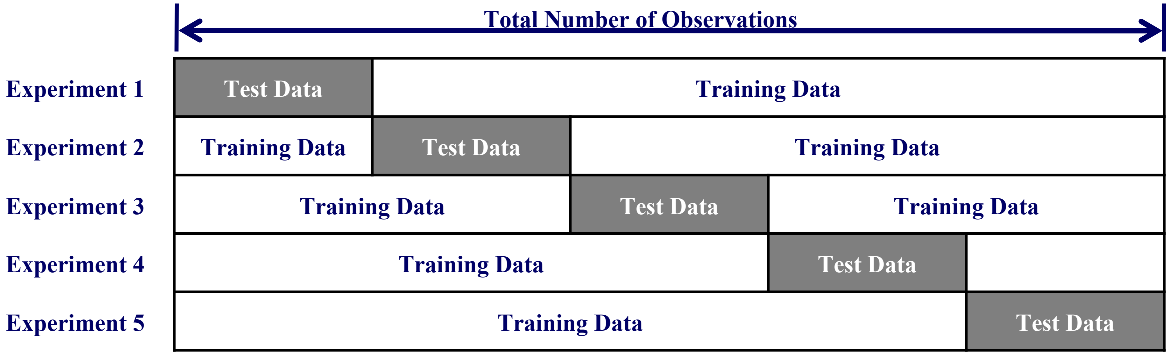
Classifier Training - Training and Testing Data

Data 150 observations

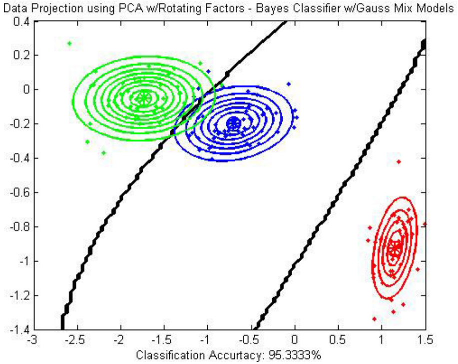
- 50 Setosa
- 50 Versicolor
- 50 Viginica

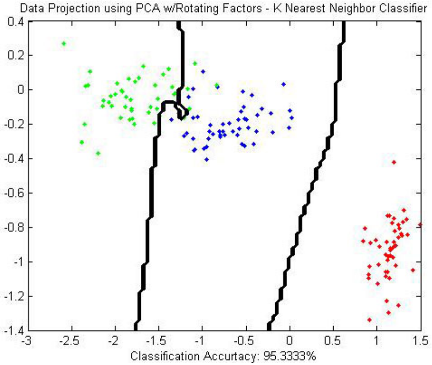
Training with 5-fold Cross Validation

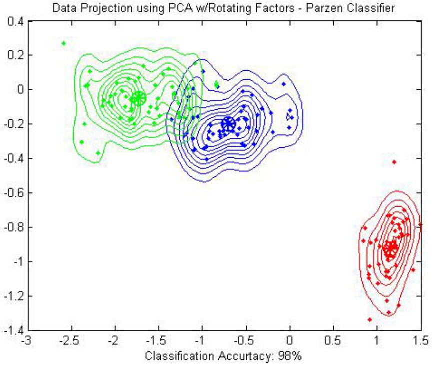
- Training (Total 120 observations)
  - 40 Setosa
  - 40 Versicolor
  - 40 Virginica
- Testing (Total 30 observations)
  - 10 Setosa
  - 10 Versicolor
  - 10 Viginica



Simple Visual Classifiers Examples







Bayes Classifier w/Gaussian Mixture Models

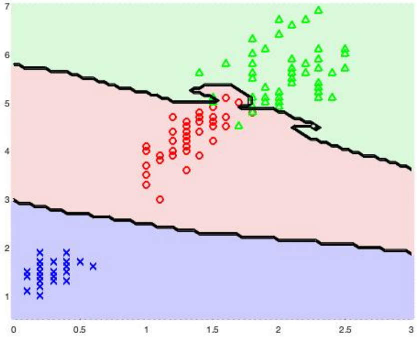
k Nearest Neighbor

Parzen Window

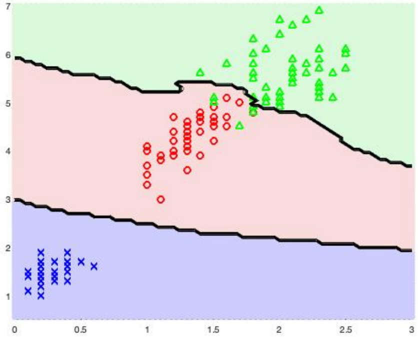
- Iris data reduced from 4 features to 2 feature with PCA

k Nearest Neighbor

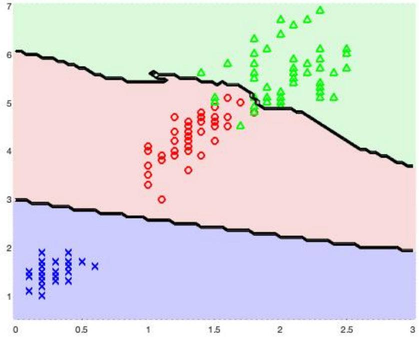
- How does k influence the algorithm



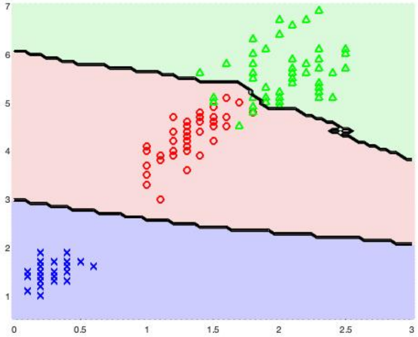
k = 2



k = 3



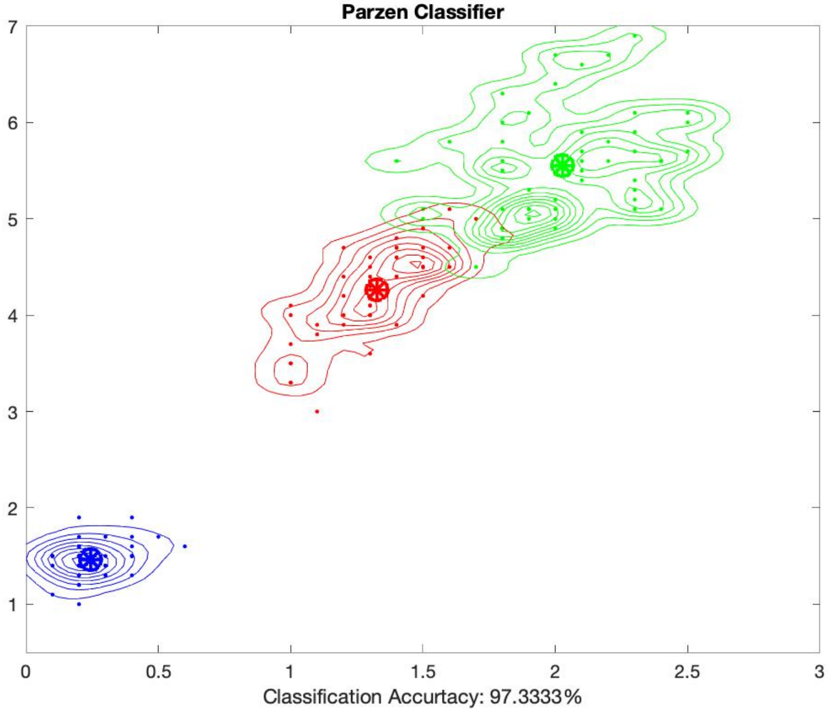
k = 4

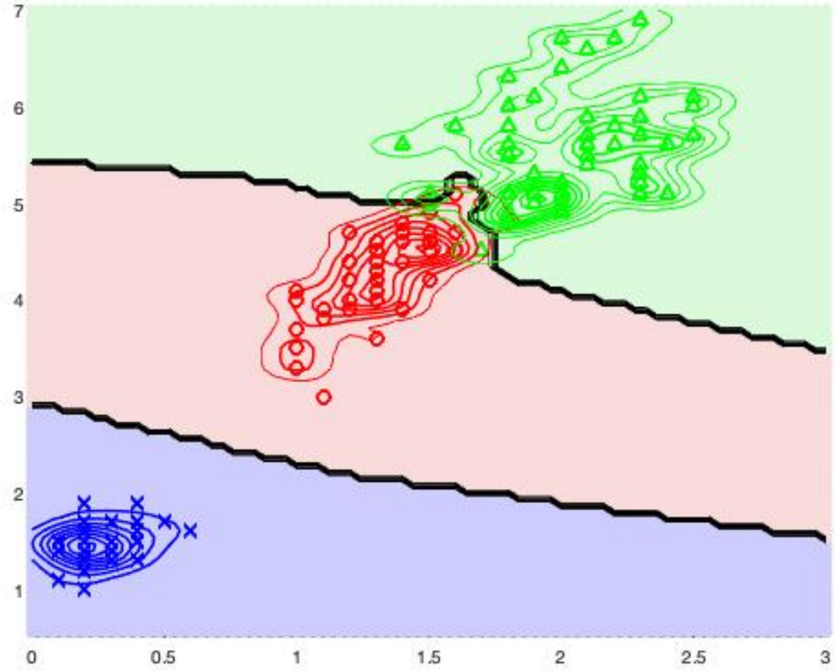


k = 5

Parzen Window

The Parzen window classifier is defendant on a kernel methods also defined as a weighting function by Parzen in 1962

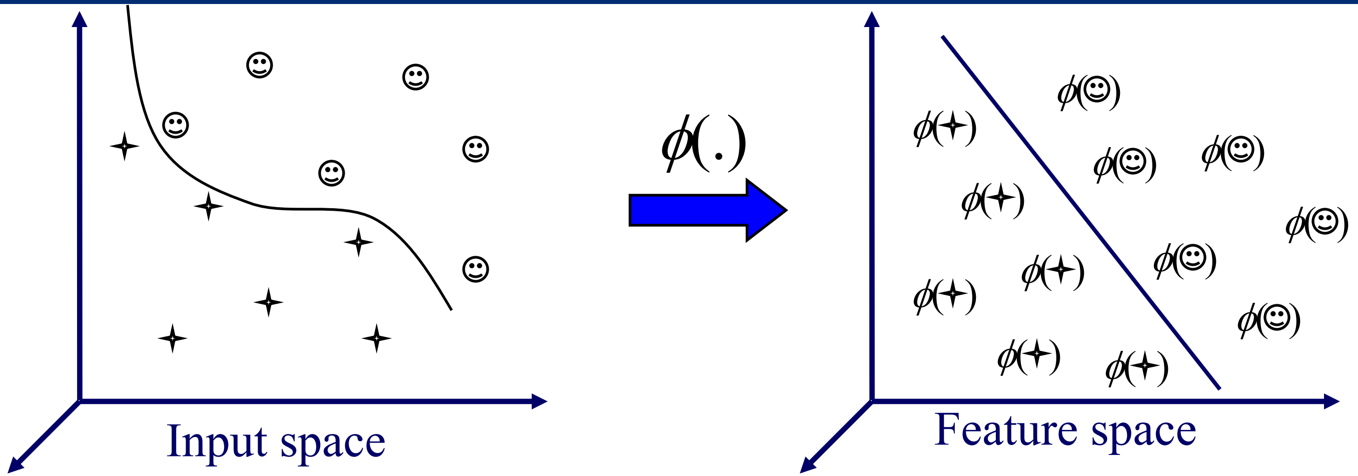




Kernel Functions

- To make the data linearly separable we could:
  - Project the data from the input space to a new space called “feature space”
  - This feature space having more dimensions than the input space we could separate the data THERE…
  - Using the Parzen method with a kernel function will allow us to separate the data in a linear fashion

Transforming (Projecting) the Data



Note: feature space is of higher dimension than the input space in practice

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is typically infinite-dimensional!
- The "kernel trick" comes to rescue

Kernel Functions

- So we could define a kernel function as follows: It is the function that represents the inner product of some space in ANOTHER space.

- Some spaces are known only by their kernel function
  (i.e., their projection is UNKNOWN)

- Some kernel functions are as follows:

  - Linear kernel: $K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle = x^T z$

  - Polynomial kernel: $K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle = \left(\gamma \langle \mathbf{x} \cdot \mathbf{z} \rangle + b\right)^d, \gamma > 0$

  - Gaussian RBF kernel: $K(\mathbf{x}, z) = e^{\|x_i - z\|^2 / 2\sigma^2}$
    - Closely related to radial basis function neural networks
    - The feature space is infinite-dimensional

  - Sigmoid kernel: $K(\mathbf{x},\mathbf{z}) = \tanh\left(\gamma \langle \mathbf{x} \cdot \mathbf{z} \rangle + b\right)$