Support Vector Machine Techniques for Nonlinear Equalization

Bhaskar Raj Upadhyay

Shamman Noor Shoudha

Contents

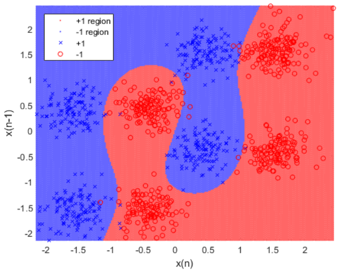
Equalization – Non Linear Equalization

- Equalization
  - Remove ISI and noise effects of the channel
  - Located at the receiver

- Severe channel effects, linear equalization methods suffer – Noise enhancement
  - Premise for Non-linear equalization

- Non-linear equalization challenges
  - Architectures maybe unmanageably complex
  - Loss of information – nonlinear system maybe non-invertible
  - Computationally intensive

Why not think of it as a classification problem?

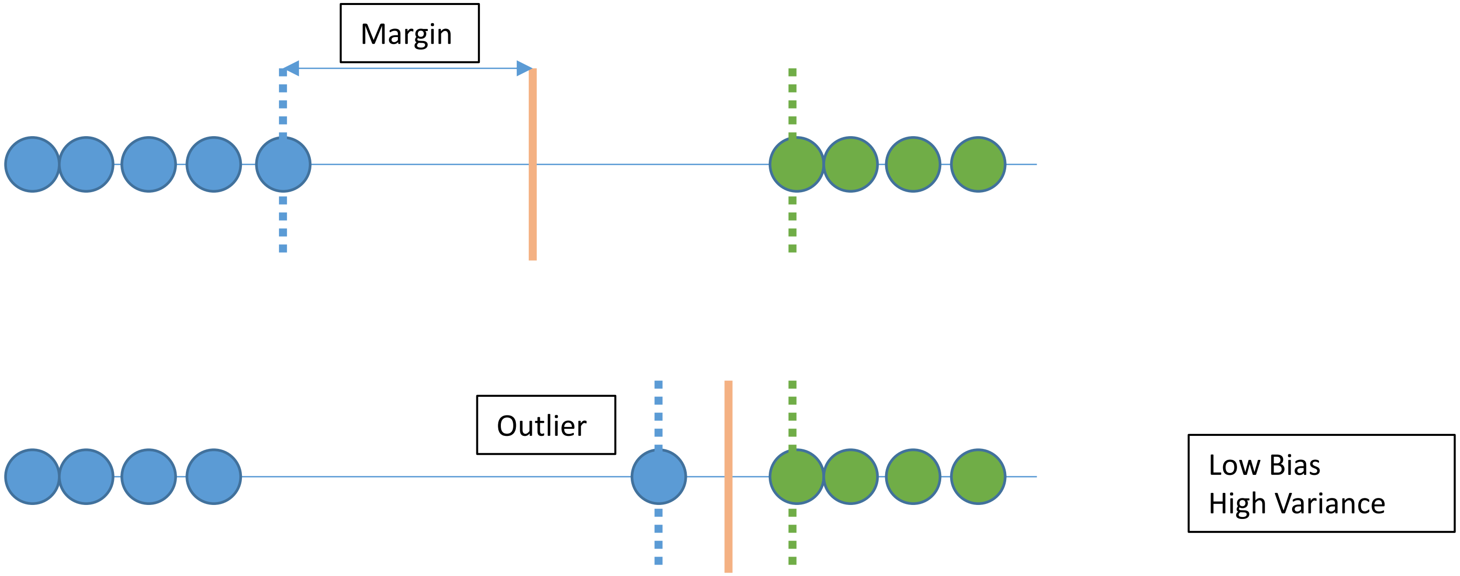


Why SVM

- Train with small amounts of data
- Training is straightforward
  - Less ad hoc input from designer
- Detection stage is efficient
- Results Comparable to Volterra filters and neural Networks
  - Volterra filters – dimension grows quickly
  - Neural networks – parameters of networks determined in an ad-hoc fashion

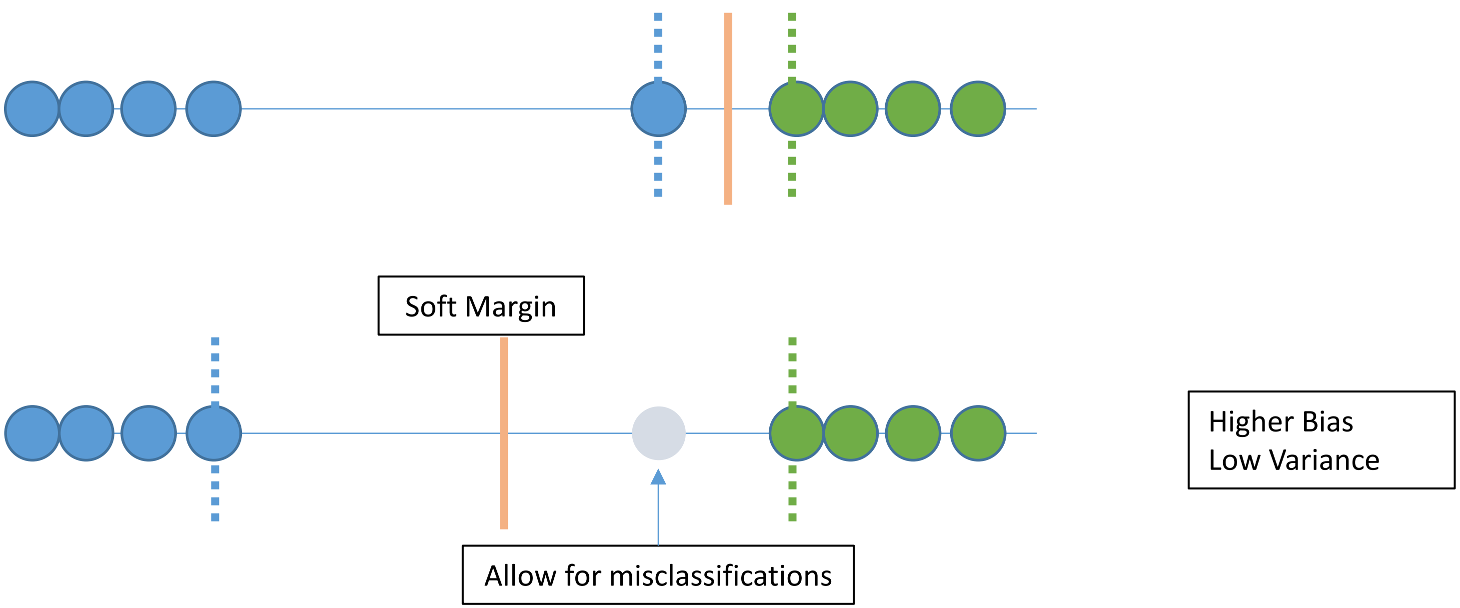
Intro to SVM

- Separate clouds of data using an optimal hyperplane
- Maximum margin classifiers don’t work well with outliers



Intro to SVM
Soft Margins and Outliers

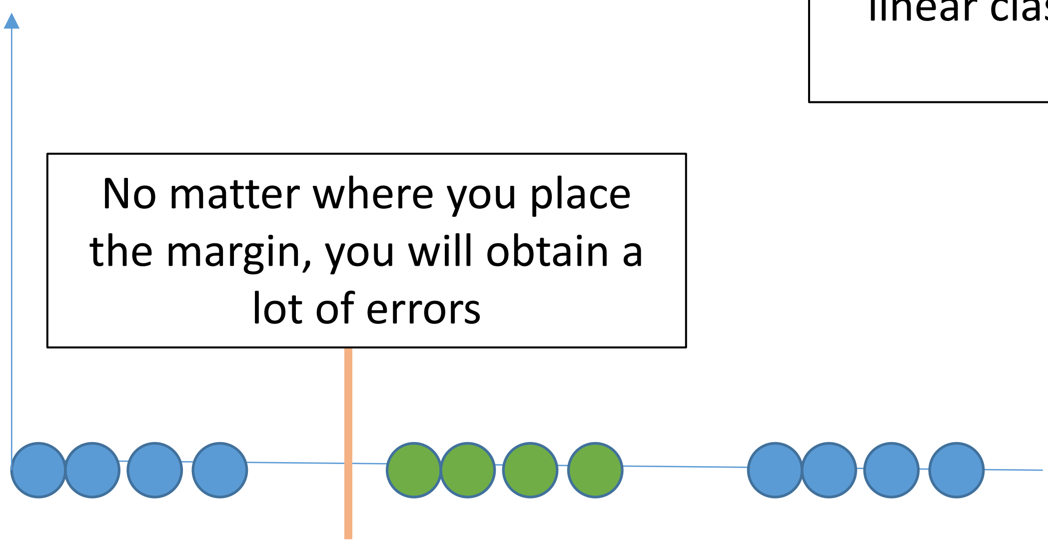
- Separate clouds of data using an optimal hyperplane
- Maximum margin classifiers don’t work well with outliers – Support vector classifiers do



Intro to SVM
Linear Classifier Limited

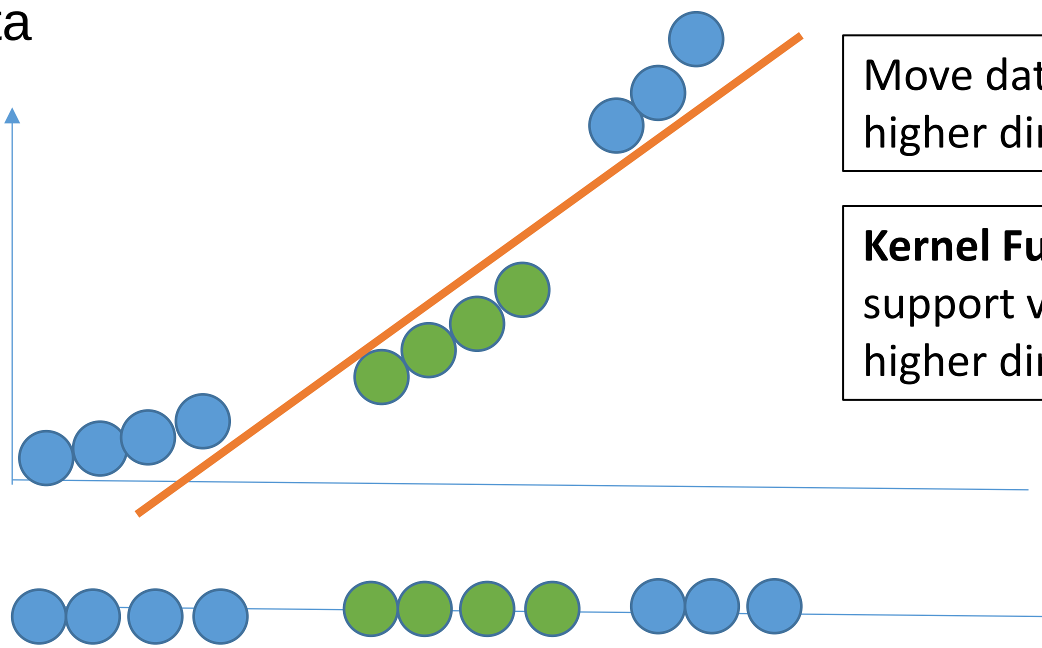
- Separate clouds of data using an optimal hyperplane
- In 2-Dimensions, the support vector classifier is a line
- **Support vector machines** – Deal with data with high amounts of overlaps

2 categories, but no obvious linear classifier to separate them.



Intro to SVM
Non Linear Classifier

- Separate clouds of data using an optimal hyperplane
- In 2-Dimensions, the support vector classifier is a line
- **Support vector machines** – Deal with data with high amounts of overlaps – non linear mapping from the pattern space to higher dimensional feature space to create linearly separable clouds of data



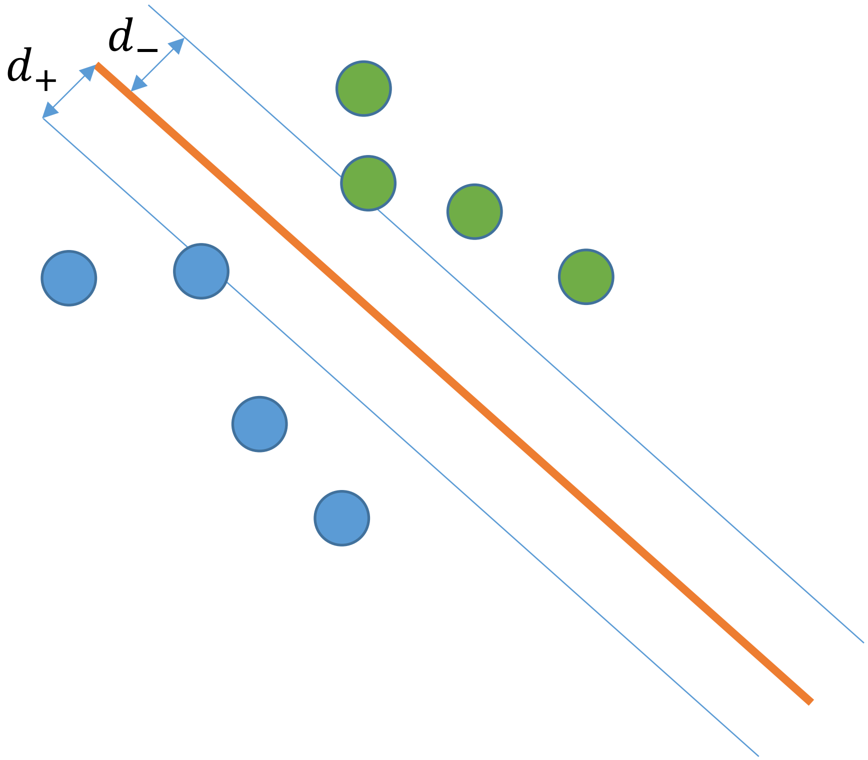
Move data to higher dimension

Kernel Functions – Find support vector classifiers in higher dimensions

Support Vector Machines
Hyperplanes and decision criter...

- Objective – Find the weights (**w**) and bias (b) to define a hyperplane:

$$\mathbf{w}^T\mathbf{x} + b = 0$$



Optimal Hyperplane – A hyperplane for which the margin of separation is maximized

Margin of separation is maximum when norm of the weight is minimized.

Support Vector Machines
Lagrangian Optimization Problem

Primal Problem

$$\min L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} a_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} a_i$$

Dual Optimization problem

$$\max L_d(a_i) = \sum a_i - \frac{1}{2}\sum a_i a_j y_i y_j K(x_i, x_j)$$

Under Constraints

$\sum_{i=1}^{l} a_i y_i = 0$ and $0 \leq a_i \leq C$

$K(\cdot,\cdot)$: Kernel

Polynomial

$$K(x, y) = (x \cdot y + 1)^p$$

Radial Basis Function

$$K(x, y) = \exp\left\{\frac{\{-||x-y||^2\}}{2\sigma^2}\right\}$$

Sigmoid Function

$$K(x, y) = \tanh(\kappa x \cdot y - \delta)$$

Why the dual?

Let's us solve the problem by computing just the inner products

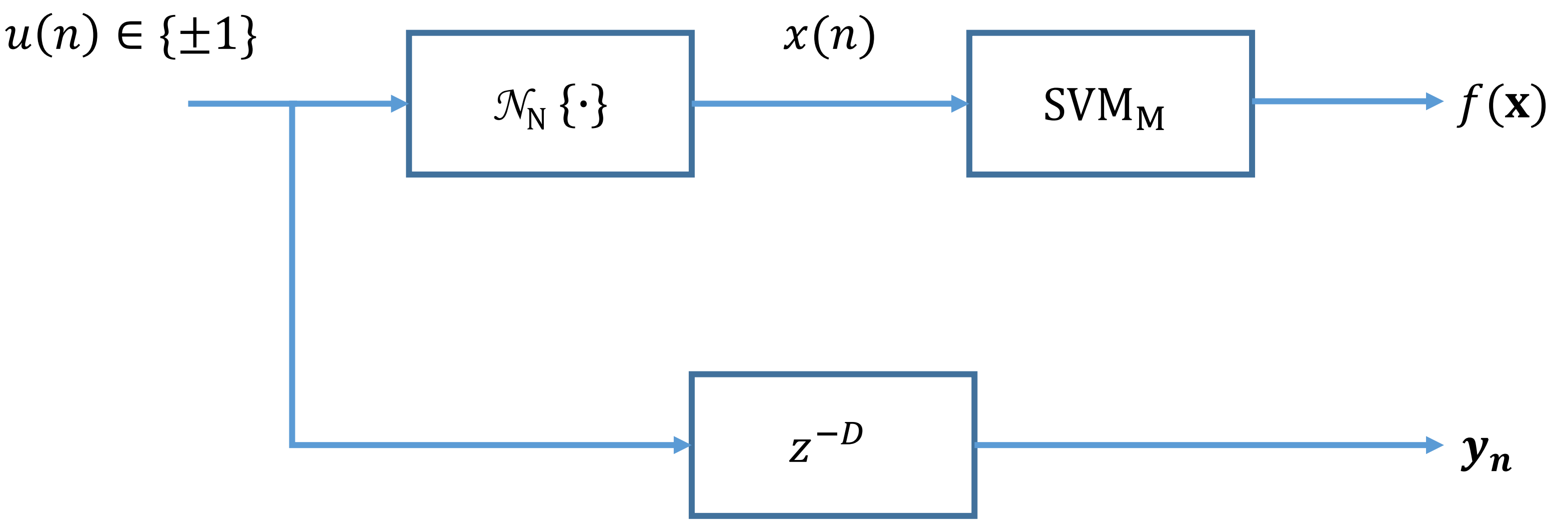
SVM Classification Equalization

$$\hat{y} = sign\{f(\mathbf{x})\}$$

- $\hat{y}$: Estimate to the classification
- $f(\mathbf{x}) = \sum_{\mathrm{i}\in\mathrm{S}} \alpha_\mathrm{i}\, \mathrm{y_i} \Phi(\mathrm{x_i}) \cdot \Phi(\mathrm{x}) + \mathrm{b} = \sum_{\mathrm{i}\in\mathrm{S}} \alpha_\mathrm{i}\, \mathrm{y_i} K(\mathrm{x_i}, \mathrm{x}) + \mathrm{b}$
  - $\{\alpha_\mathrm{i}\}$ – Lagrange Multipliers
  - S – Set of indices for which x_i is a support vector
  - $K(\cdot,\cdot)$ - Kernel satisfying conditions of Mercer's theorem
  - b – Affine offset

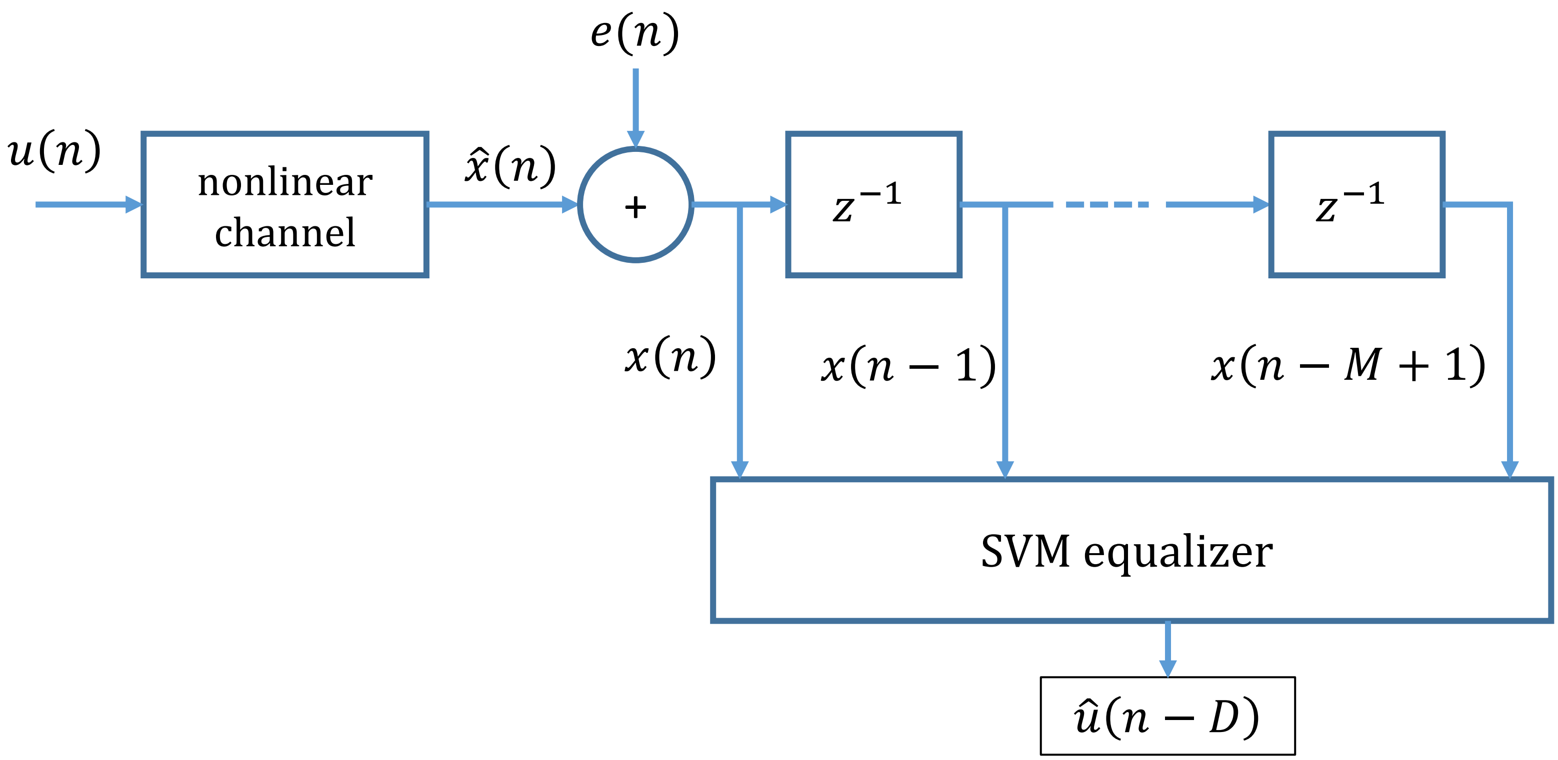
- Training set consists of
- $\mathbf{x_i} \in \mathbf{R^M}$
- $\mathbf{y}_\mathrm{i} \in \{-1,1\}$, $\mathrm{i} = 1, \dots, \mathrm{L}$

System Model



- $\mathcal{N}_{\mathrm{N}}\{\cdot\}$ – Nonlinear system
- $x(n)$ – Nonlinear system output
- $u(n)$ – Training sequence
- y_n – Desired output (delayed version of training sequence)

Nonlinear Transmission Systen PAM

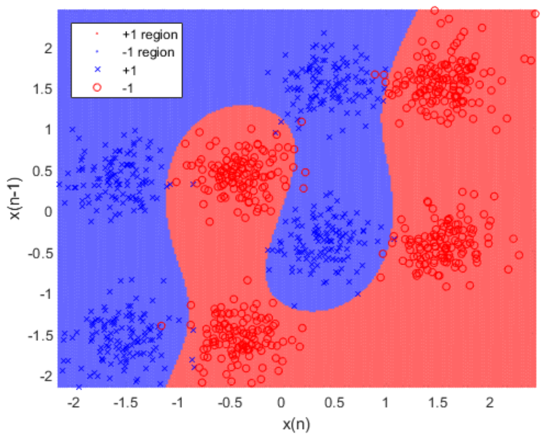


- $\hat{x}(n)$ – Nonlinear channel output
- $e(n)$ – Additive Noise
- $(M-1)$ – Feed Forward Delay (No of past channel outputs utilized)
- $\hat{u}(n-D)$ – Equalizer detection output (goal to mimic $u(n-D)$)

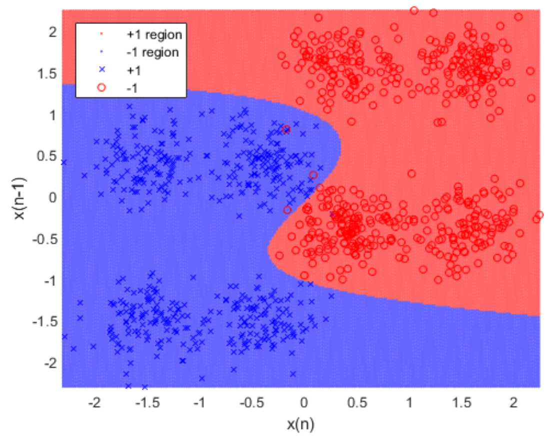
System Structure and Parameters

- $\hat{x}(n) = \tilde{x}(n) - 0.9\tilde{x}^3(n)$
- $\tilde{x}(n) = u(n) + 0.5u(n-1)$
- $e(n) \sim N(0, \sigma_e^2) \rightarrow N(0,0.2)$
- SVM Parameters
  - C = 5 (constraint)
  - d = 3 (equalizer kernel order)
  - M = 2 (equalizer dimension)
  - Kernel = Polynomial

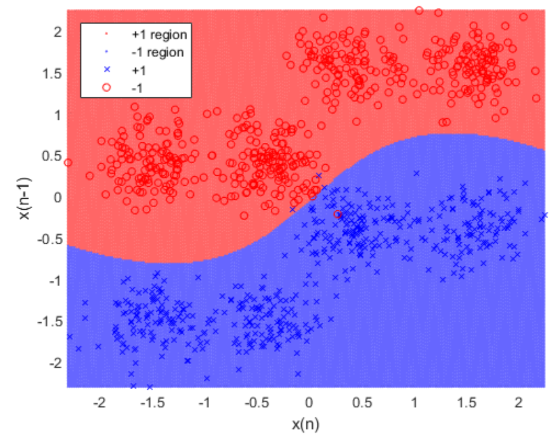
Simulation Results
Typical classification regions of an SVM



$D = 0$



$D = 1$



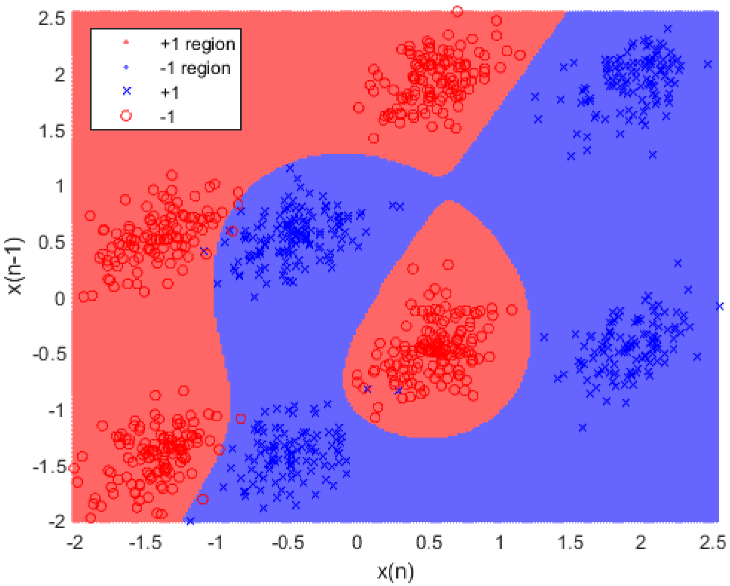
$D = 2$

- $\hat{x}(n) = \tilde{x}(n) - 0.9\tilde{x}^3(n)$
- $\tilde{x}(n) = u(n) + 0.5u(n-1)$
- $e(n) \sim N(0, \sigma_e^2) \rightarrow N(0,0.2)$

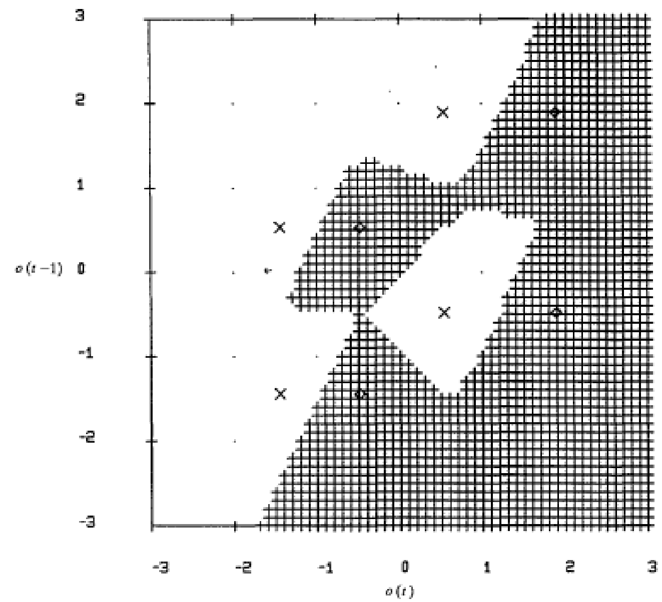
- C = 5 (constraint)
- d = 3 (equalizer kernel order)
- M = 2 (equalizer dimension)
- Kernel = Polynomial

Results – Decision Boundaries
Colored Noise

Ref: Chen et al



Correlated Noise – SVM

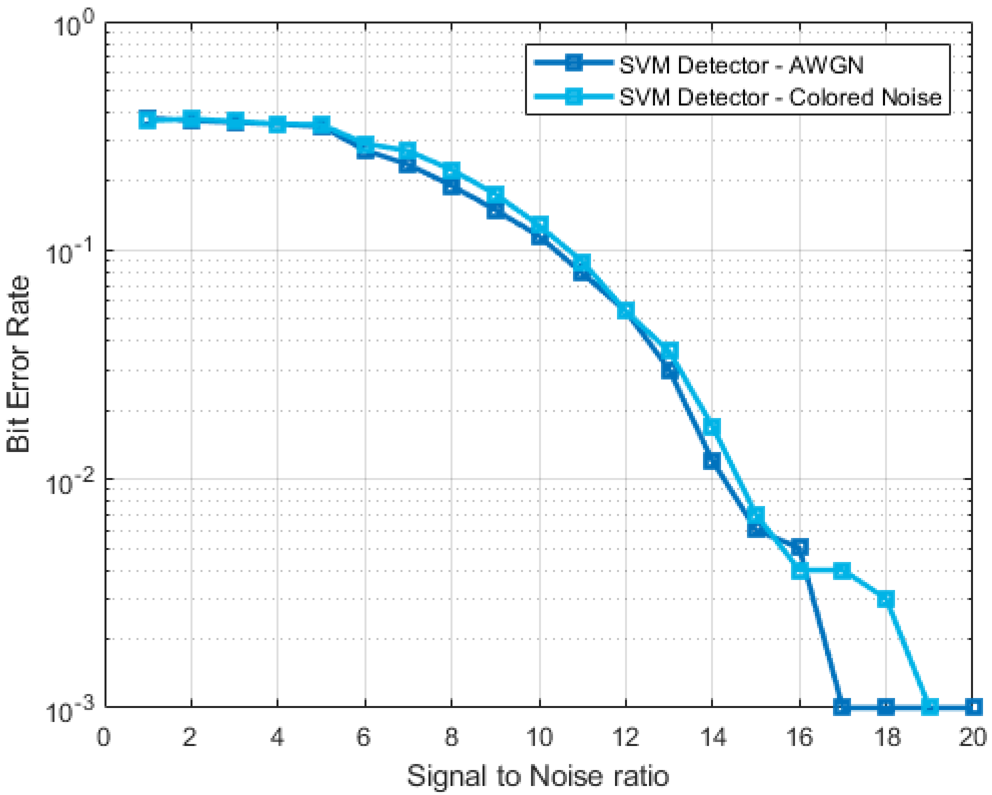


Correlated Noise – Optimum

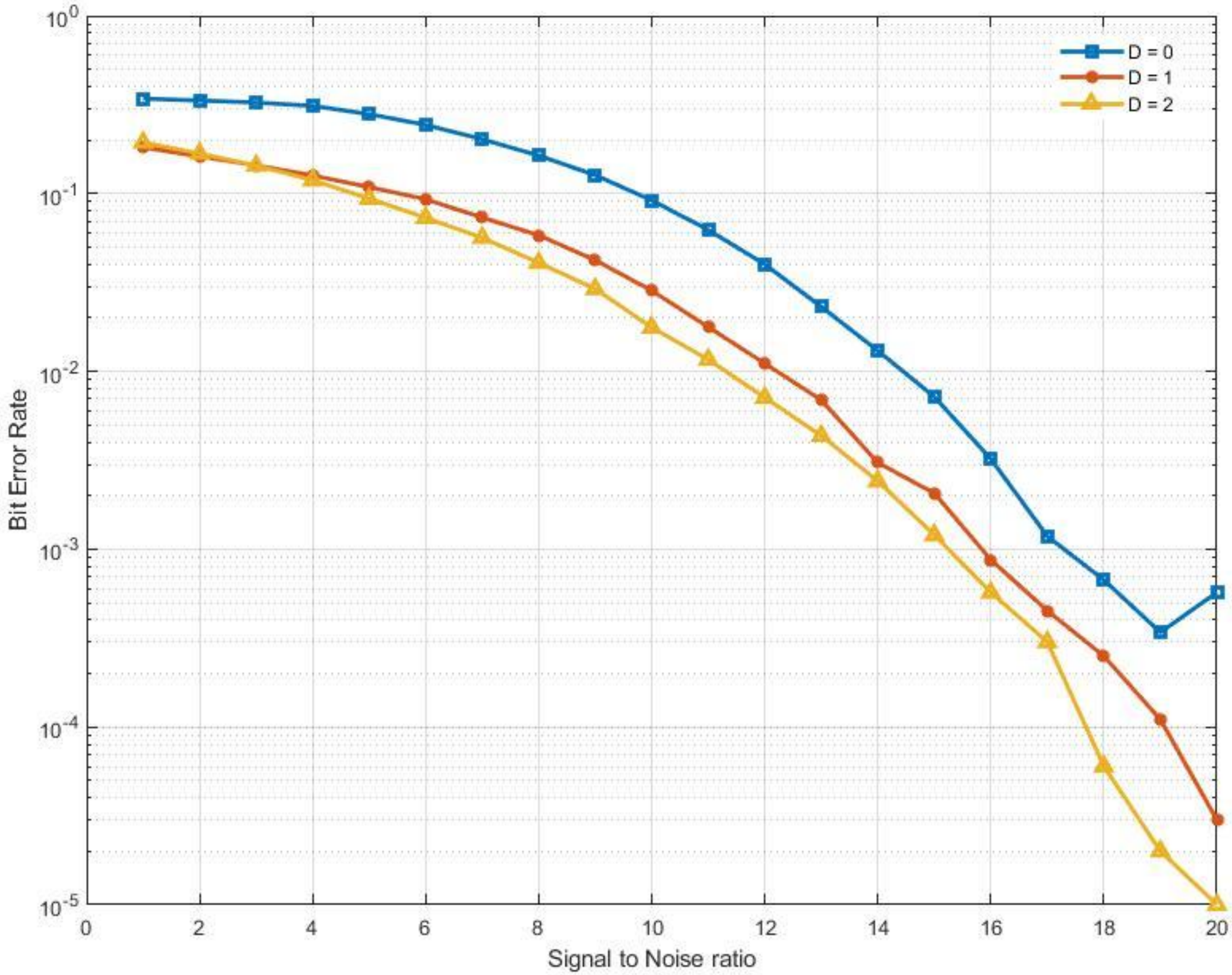
- $CorrMat = \sigma_e^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$
- $\rho = 0.48$
- $M = 2,\ D = 0, d = 3$

- $\hat{x}(n) = \tilde{x}(n) + 0.1\tilde{x}^2 + 0.05\tilde{x}^3$
- $\tilde{x}(n) = 0.5u(n) + u(n-1)$
- $\sigma_e^2 = 0.2$

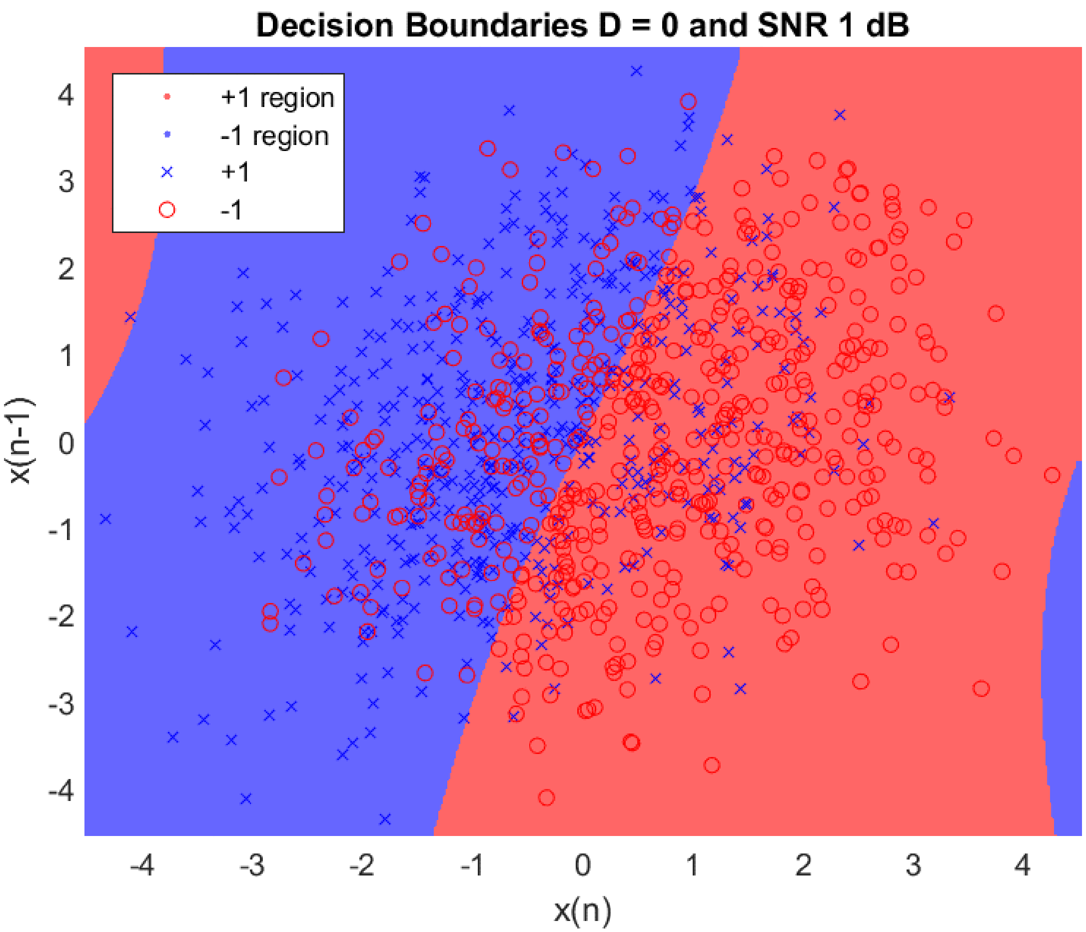
Results – BER
Colored Noise Vs AWGN



Results – BER
For different values of D=0, 1, 2

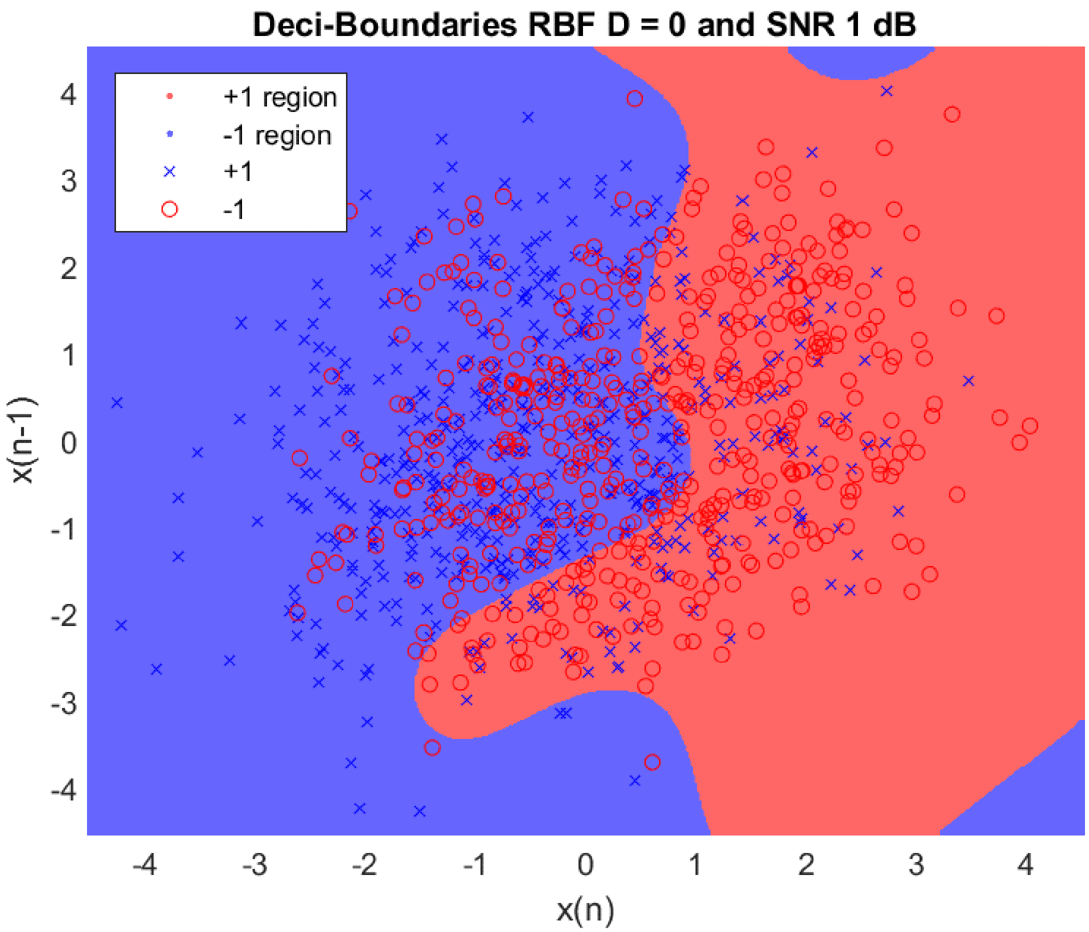


Decision Boundaries and SNR Polynomial Kernel



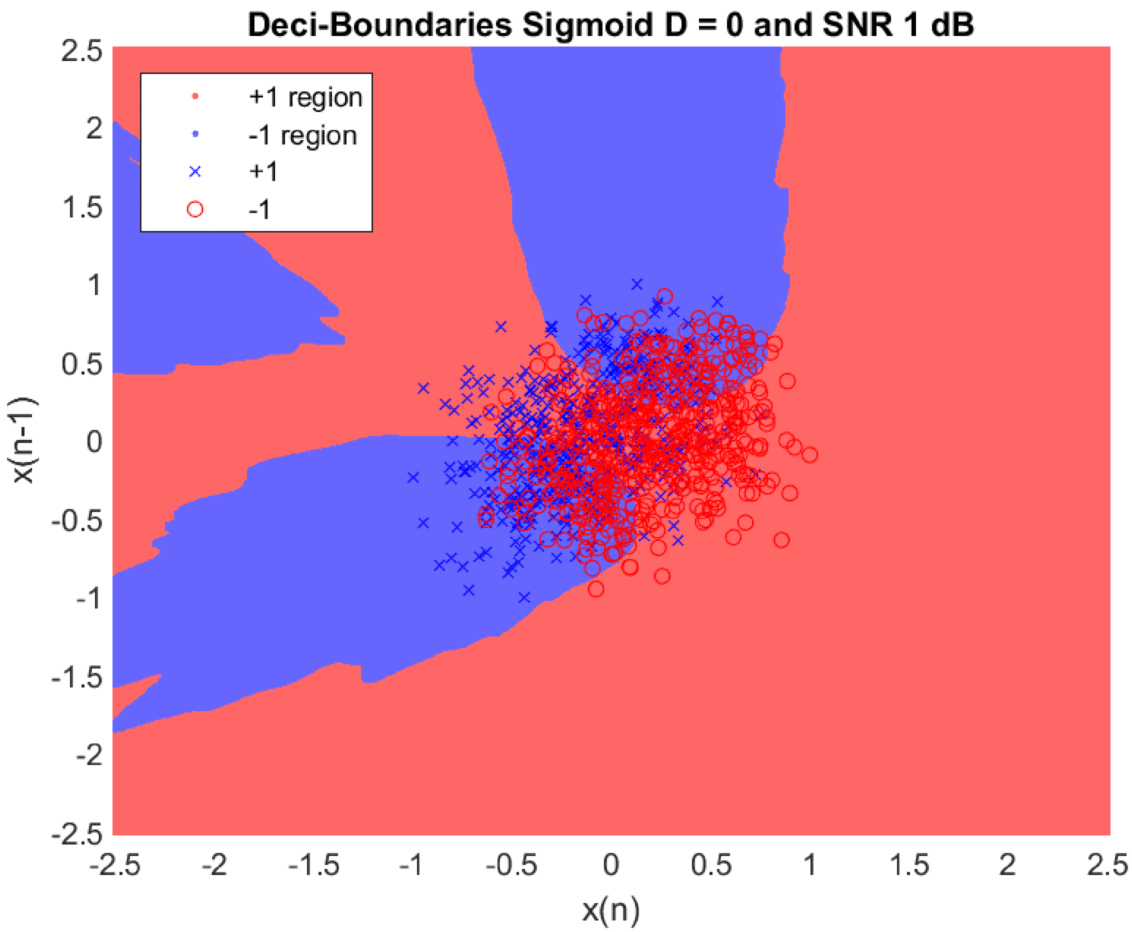
- $K(\mathbf{x},\mathbf{z}) = (\mathbf{x}^T\mathbf{z} + 1)^d$
- d = polynomial order
- All polynomials up to degree d
- For our simulation, d = 3
- $(\mathbf{x}^T\mathbf{z} + 1)^d = O(n)$ computation
- Feature space might be non – unique

Decision Boundaries and SNR RBF Kernel

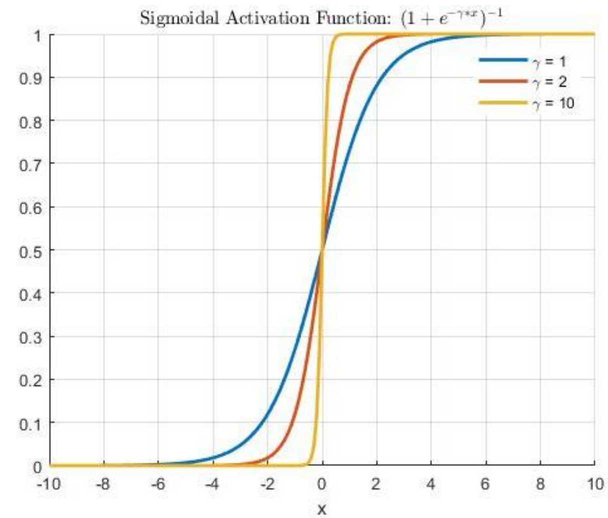


- $K(\mathbf{x},\mathbf{z}) = \exp(-\gamma\|\mathbf{x}-\mathbf{z}\|_2^2)$
- Infinite dimensional space
- Parameter $= \gamma$
- As γ increases, the model overfits
- As γ decreases, the model underfits
- For our simulation, $\gamma = 1$

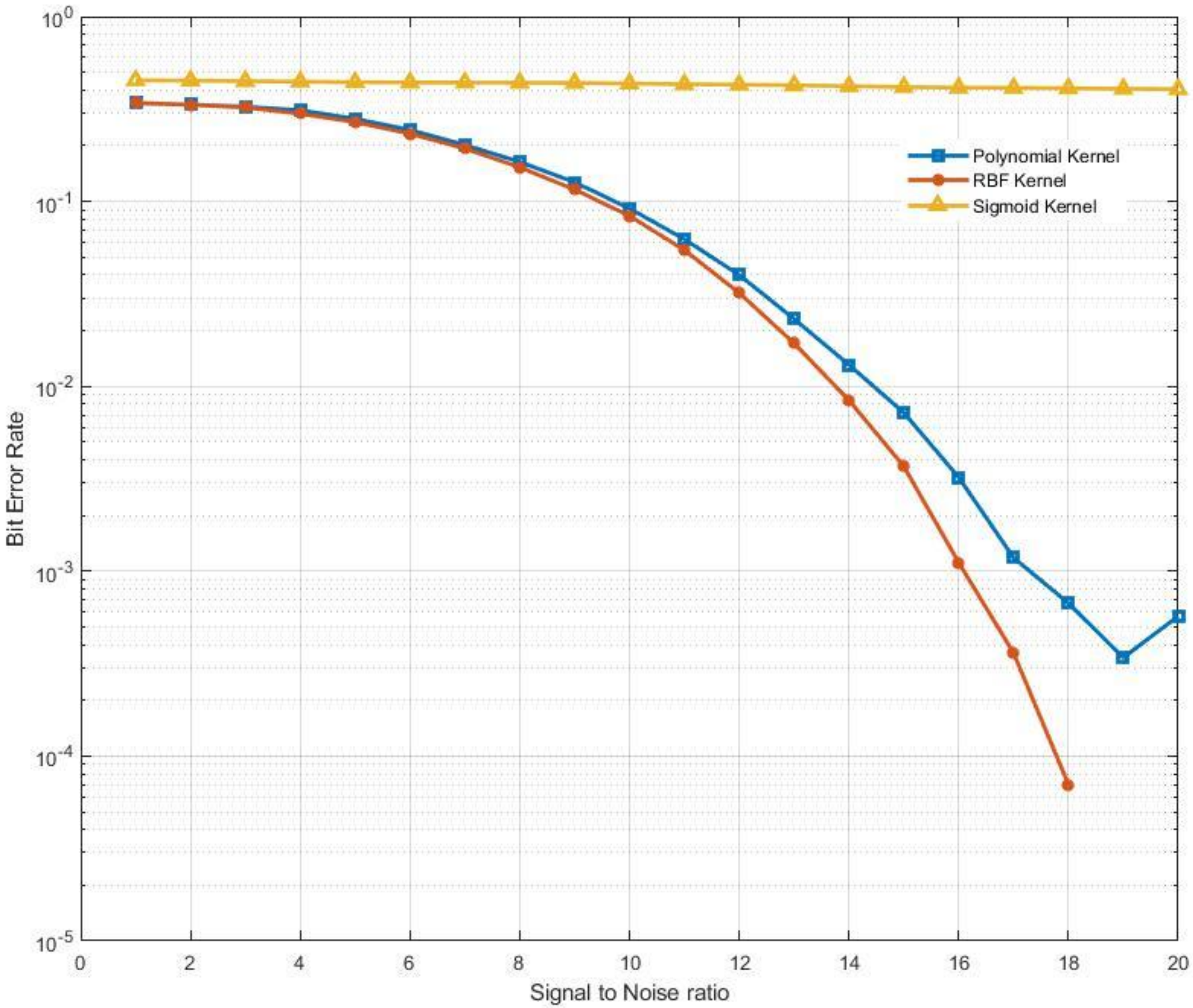
Decision Boundaries and SNR Sigmoid Kernel



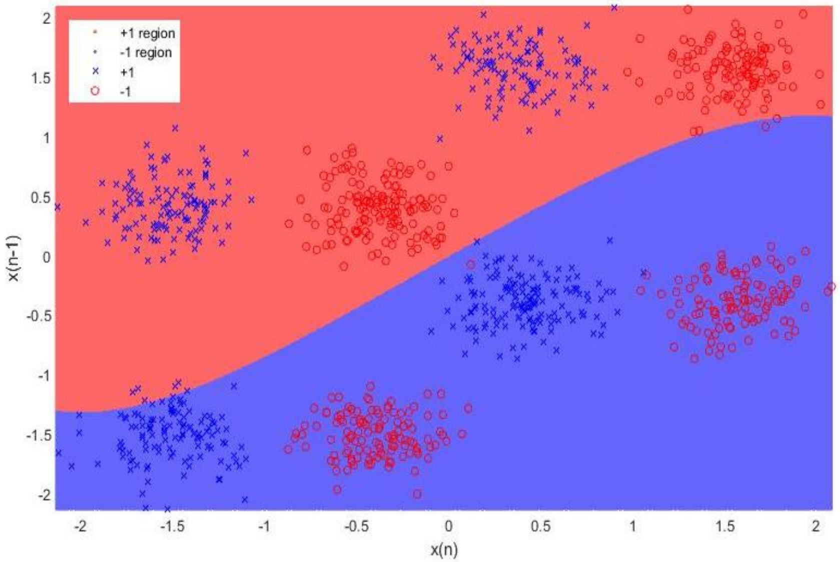
- $K(\mathbf{x}, \mathbf{z}) = \tanh\left(k\mathbf{x}^T\mathbf{z} - \delta\right)$
- $k =$ slope
- $\delta =$ intercept
- For our simulation,
  - $k = 10, \delta = 10$
- Sigmoidal kernels can be thought of multi-layer perceptron

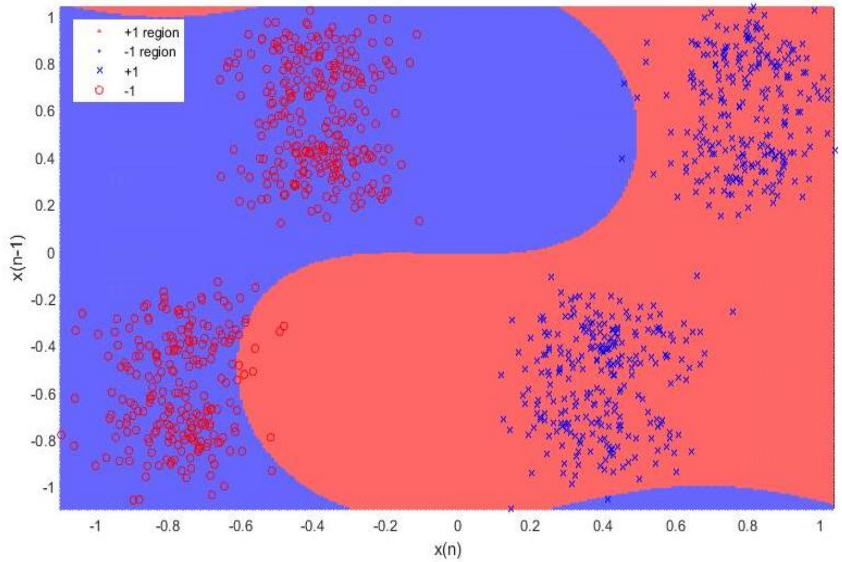


Results – BER
For different SVM Kernels



Offline Training
Generalization over different channels

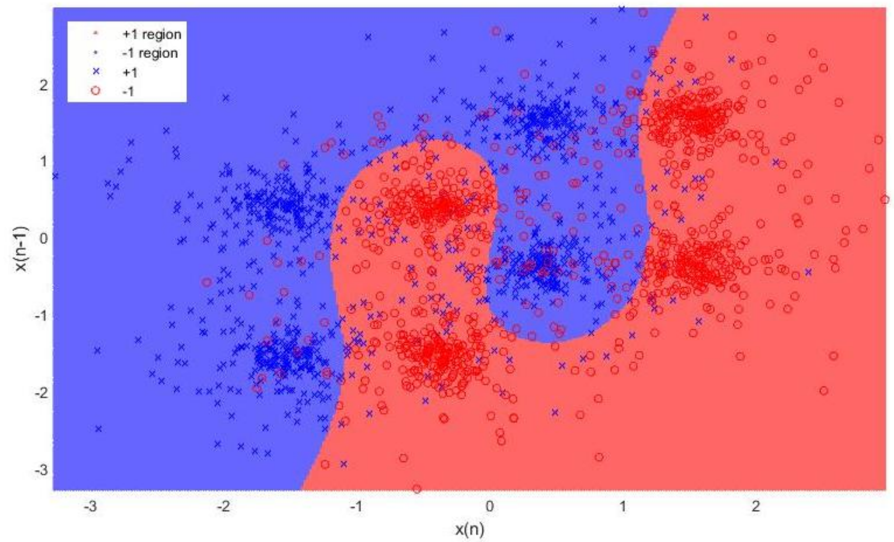


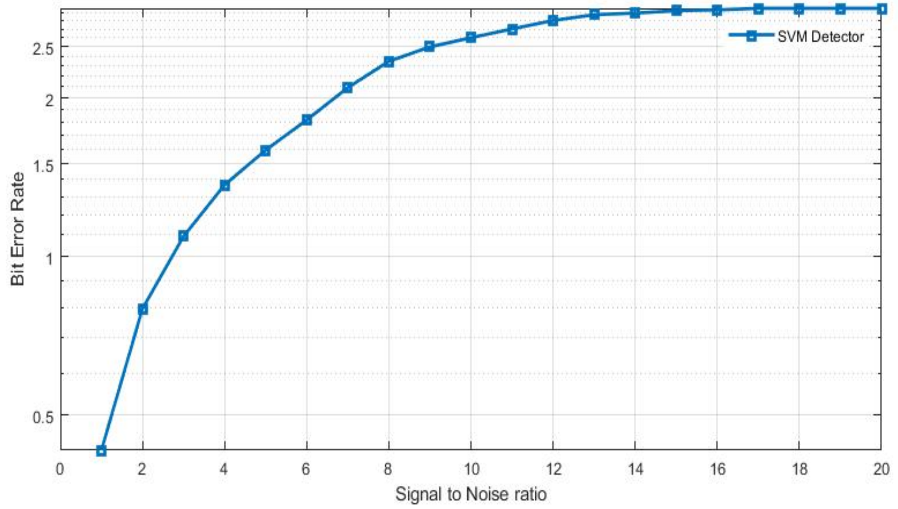


- $\hat{x}_{train}(n) = \tilde{x}(n) - 0.9\tilde{x}^3(n)$
- $\tilde{x}_{train}(n) = u(n) + \mathbf{0.9}u(n-1)$
- $\hat{x}_{test}(n) = \tilde{x}(n) - 0.9\tilde{x}^3(n)$
- $\tilde{x}_{test}(n) = u(n) + \mathbf{0.5}u(n-1)$

- $\hat{x}_{train}(n) = \tilde{x}(n) - \mathbf{0.5}\tilde{x}^3(n)$
- $\tilde{x}_{train}(n) = u(n) + 0.6u(n-1)$
- $\hat{x}_{test}(n) = \tilde{x}(n) - \mathbf{0.3}\tilde{x}^3(n)$
- $\tilde{x}_{test}(n) = u(n) + 0.6u(n-1)$

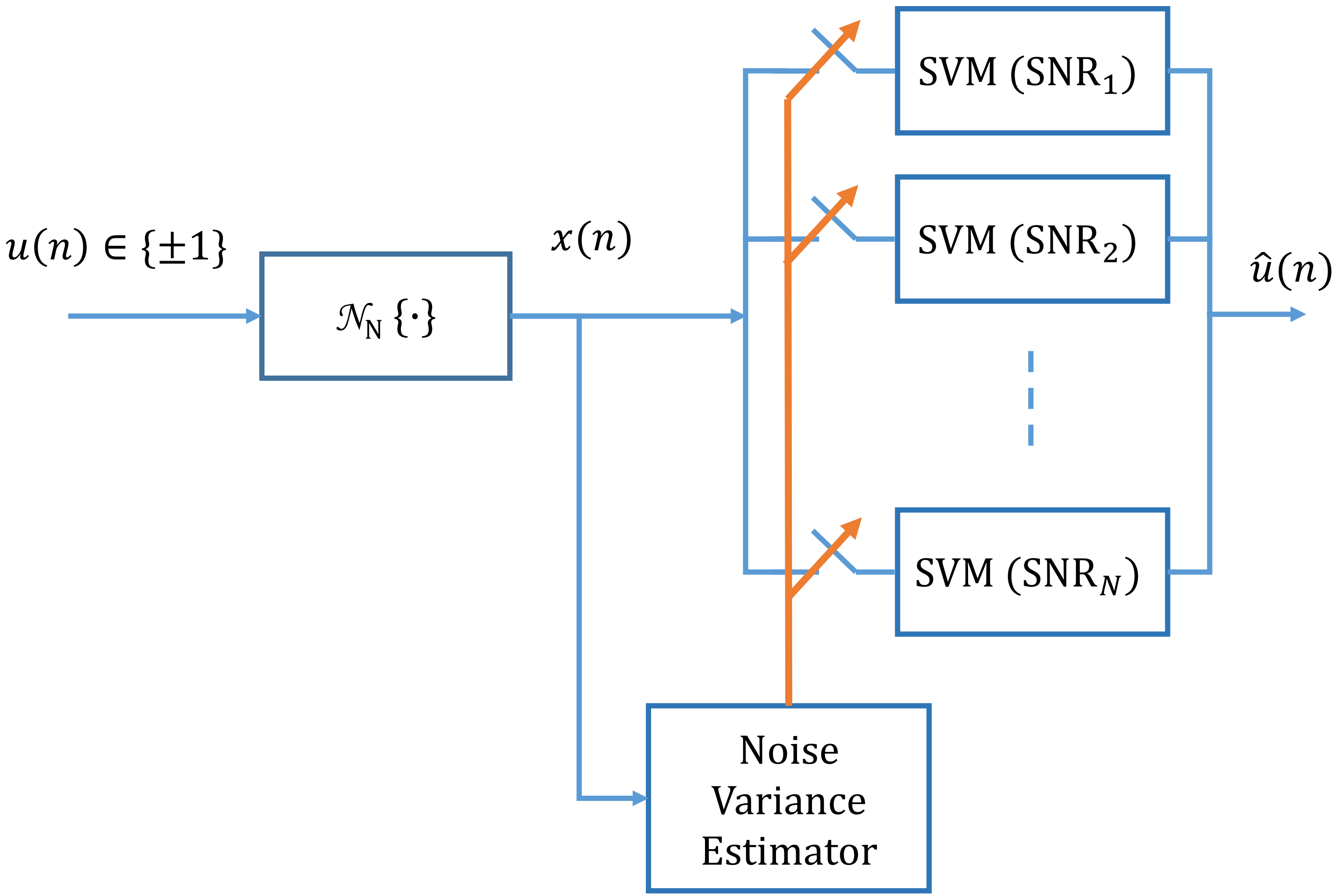
Offline Training
Generalization over different SNRs



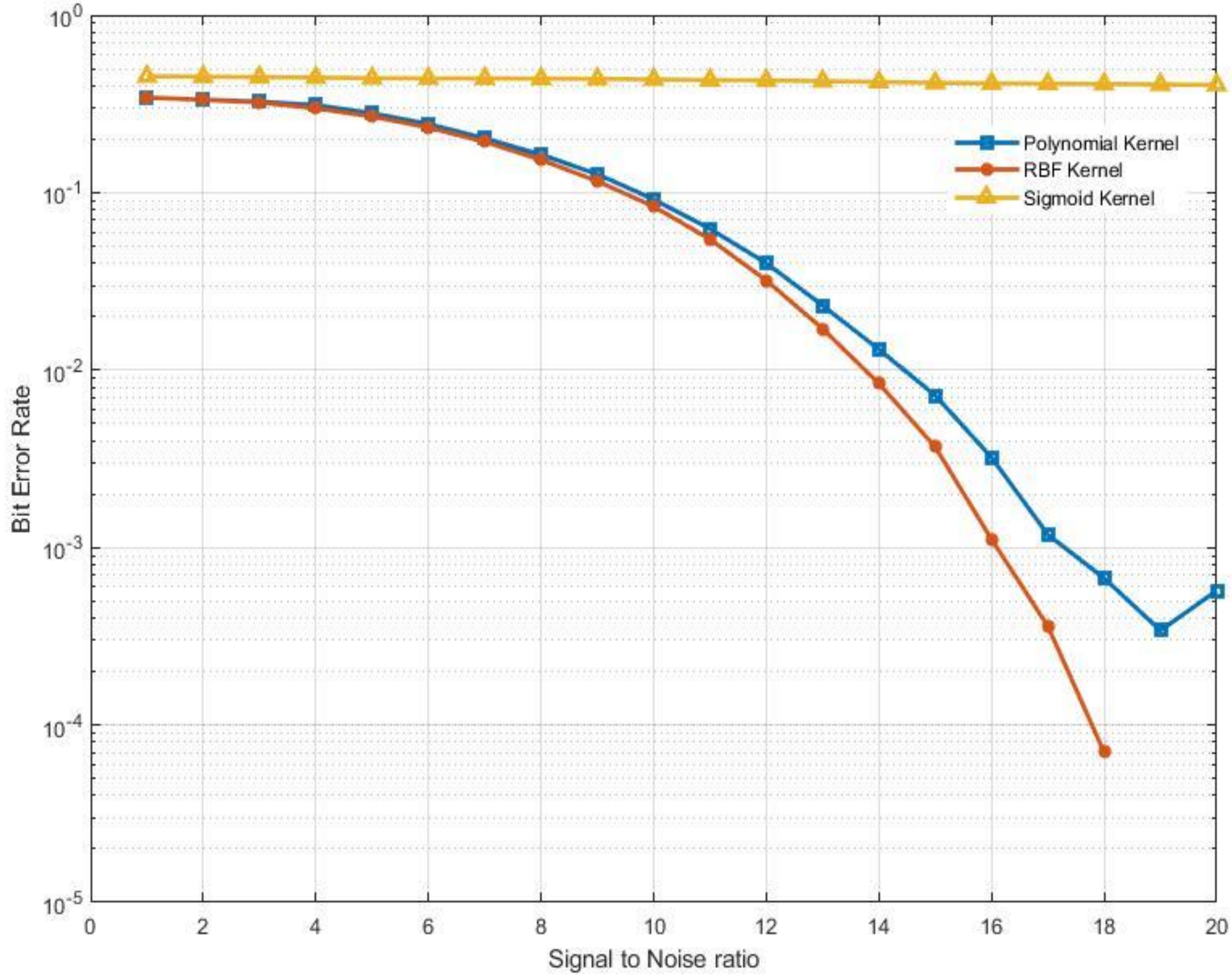


- Training SNRs = 1: 20 dB
- Testing SNRs = 1: 20 dB
- Does not generalize well over different SNR values and multiple channels

SVM-Bank for Different SNR signals



Results – BER
For Bank of SVM



Summary

- We looked at SVM as a Dual Lagrangian Optimization Problem and how it fits in non-linear equalization problem.
- We developed a non-linear channel communication system and applied SVM equalizer.
- For different values of Detector Delay (D) and SVM kernels, we found different BER performance of the SVM equalizer.
- For Unknown SNR, the SVM equalizer does not generalize well to unknown channel and unknown SNR.
- To solve the issue of SNR, we proposed a bank of SVM with SVM models trained with different SNR values. After receiving the signal, noise variance estimator block will select the desired SVM model for equalization.

Thank You