Kernel Methods and Applications

Theodore B. Trafalis

Laboratory for Optimization and Intelligent Systems
School of Industrial Engineering
University of Oklahoma
(www.lois.ou.edu)

School of ECE, NTUA, Lectures on Kernel Methods and Applications, December 11, 2014

Outline

- Part I: Theory of Kernel Methods
- Part II: Applications to Rainfall Estimation

Part I: Theory of Kernel Methods

Outline of Part I

- Historical perspective
- Overview of kernel methods
- Learning
- The basic idea of kernel methods
- Observations
- Parzen's windows as a kernel method
- Support Vector Machines (SVMs)
- Kernels
- A classical example (XOR with polynomial kernel)
- Kernel methods with uncertain data
- Incremental kernel methods
- Minimax Probability Machine (MPM)
- Analytic Center Machines
- Other topics

Historical Perspective

- Efficient algorithms for detecting linear relations were used in the 1950s and 1960s (perceptron algorithm).
- Handling nonlinear relationships was seen as major research goal at that time but the development of nonlinear algorithms with the same efficiency and stability has proven as an elusive goal.
- In the mid 80s the field of pattern analysis underwent a nonlinear revolution with backpropagation neural networks (NNs) and decision trees (based on heuristics and lacking a firm theoretical foundation, local minima problems, nonconvexity).
- In the mid 90s, kernel based methods have been developed while retaining the guarantees and understanding that have been developed for linear algorithms.

Overview

- Kernel Methods are a new class of machine learning algorithms which can operate on very general types of data and can detect very general types of relations (e.g., Potential function method; Aizerman et al., 1964, Vapnik, 1982, 1995)

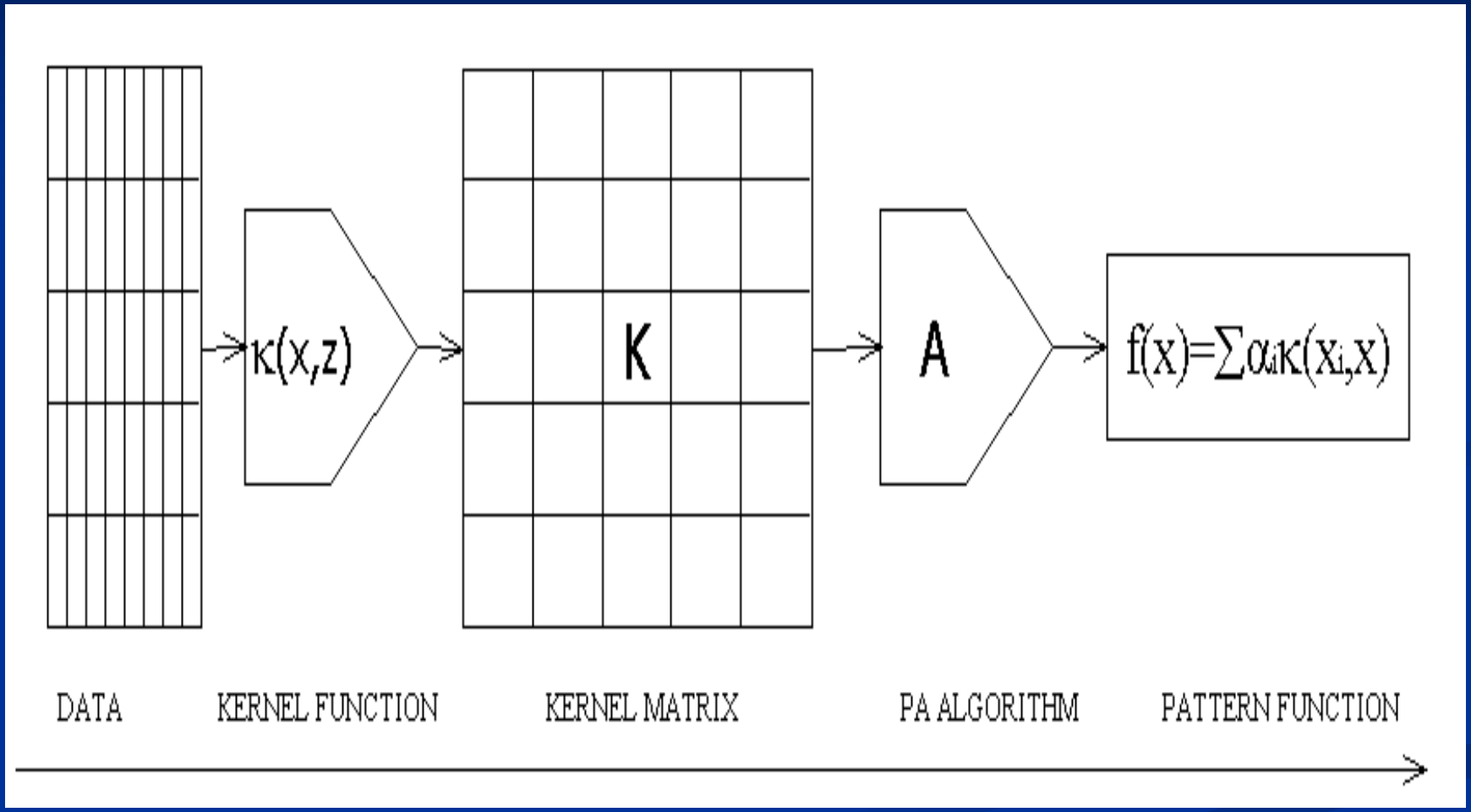
- Correlation, factor, cluster and discriminant analysis are some of the types of machine learning analysis tasks that can be performed on data as diverse as sequences, text, images, graphs and vectors using kernels

- Kernel methods provide also a natural way to merge and integrate different types of data

Cont'd

- Kernel methods offer a modular framework
- In a **first step**, a dataset is processed into a kernel matrix. Data can be of various types and also of heterogeneous types
- In a **second step**, a variety of kernel algorithms can be used to analyze the data, using only the information contained in the kernel matrix

Modular Framework



Source: J. Shawe-Taylor and N. Cristianini, Kernel methods for pattern analysis, 2004

Cont'd

- Computationally most kernel-based learning algorithms reduce to convex optimization problems

- Kernel design is based on various optimization methods. For discrete data (e.g., sequences) often use methods like dynamic programming, branch and bound, discrete continuous optimization, etc

Cont'd

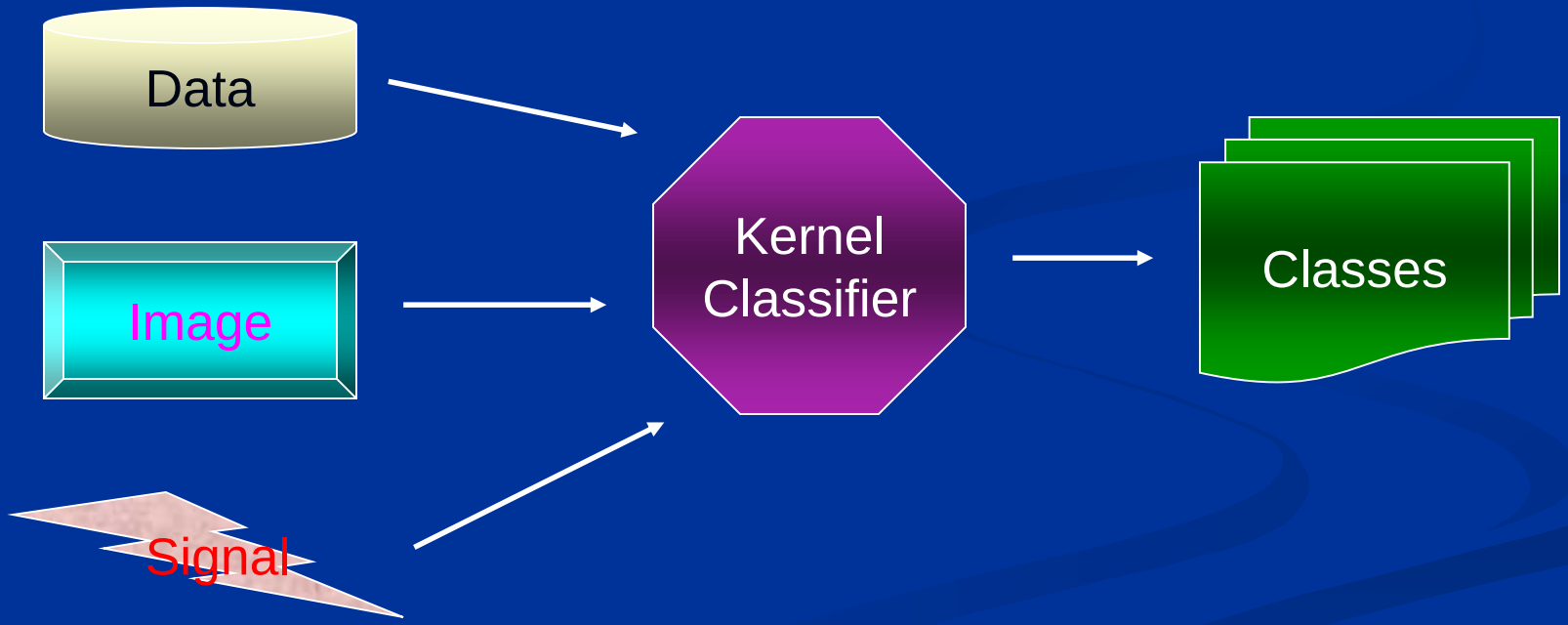
- The flexible combination of appropriate kernel design and relevant kernel algorithms has given rise to a powerful and coherent class of methods, whose computational and statistical properties are well understood (Schölkopf & Smola, 2002; Shawe-Taylor and Cristianini, 2004)

Cont'd

- Increasingly used in applications as diverse as biosequences and microarray data analysis, text mining, machine vision, handwriting recognition, weather prediction, metrology

Learning from Data

- An essential procedure for pattern recognition



Learning from Examples

EXAMPLES

$Input_1$ → $Output_1$

$Input_2$ → $Output_2$

.....

$Input_n$ → $Output_n$

Cont'd

Given a set of l examples (past data)

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_\lambda, y_\lambda)\}$$

Question: find a function f such that

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

is a ***good predictor*** of y for a ***future*** input x

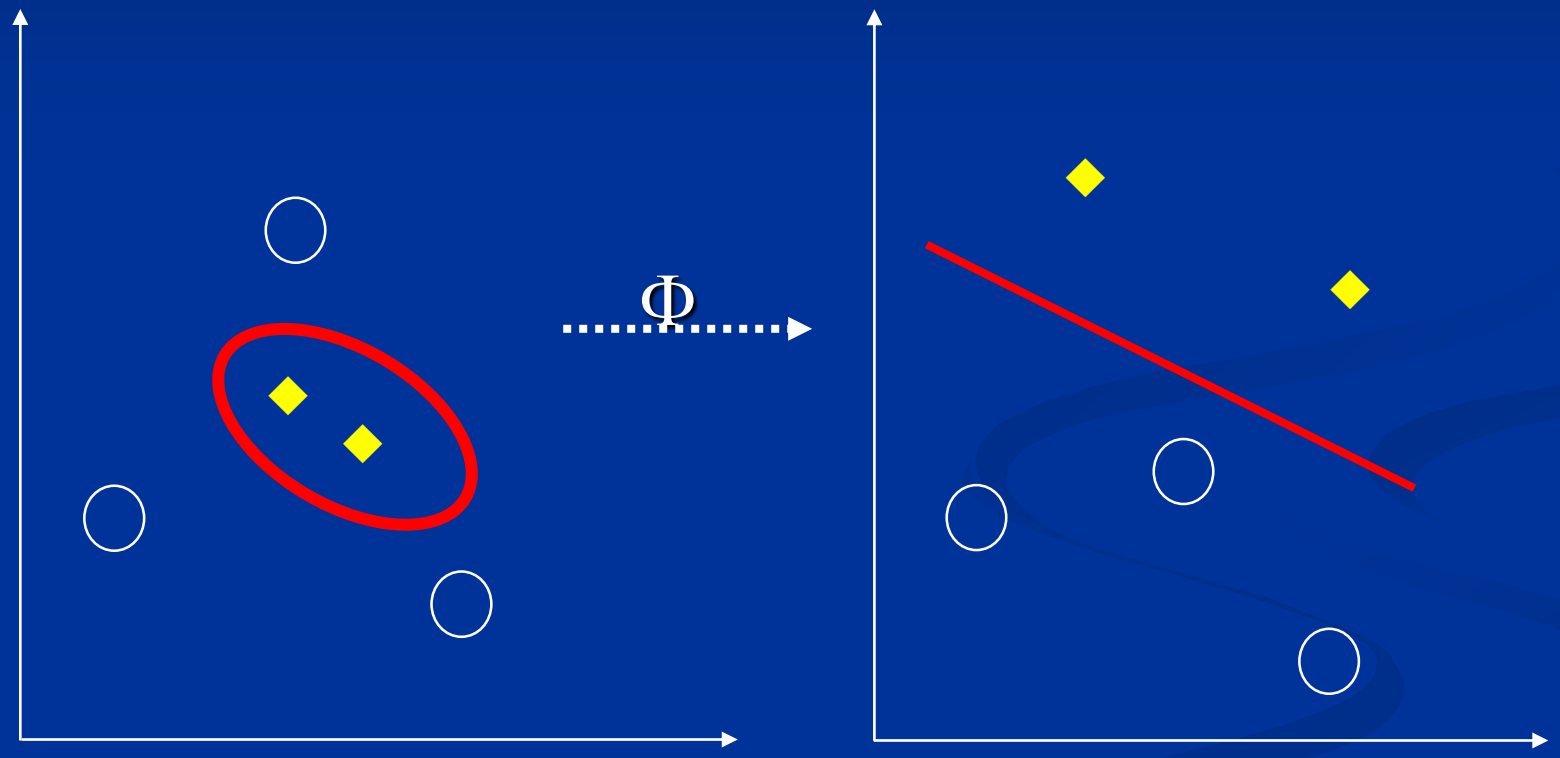
Basic Idea of Kernel Methods

- Kernel Methods work by:
  - Embedding data in a vector space called feature space using a kernel function
  - Looking for linear relations in such a space

Idea of Kernel Methods

input space

feature space



$$K(x, y) = < \Phi(x), \Phi(y) >$$

Observations

- Much of the geometry of the data in the embedding space (relative positions) is contained in all pair-wise inner products (information bottleneck)

- Inner product matrix (Kernel matrix)

$$\mathbf{K} = \begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & K(x_1,x_3) & K(x_1,x_4) \\ K(x_2,x_1) & K(x_2,x_2) & K(x_2,x_3) & K(x_2,x_4) \\ K(x_3,x_1) & K(x_3,x_2) & K(x_3,x_3) & K(x_3,x_4) \\ K(x_4,x_1) & K(x_4,x_2) & K(x_4,x_3) & K(x_4,x_4) \end{bmatrix}$$

Cont'd

- We can work in feature space by specifying an inner product function K between points in it
- In many cases, inner product in the embedding space (feature space) is very cheap to compute

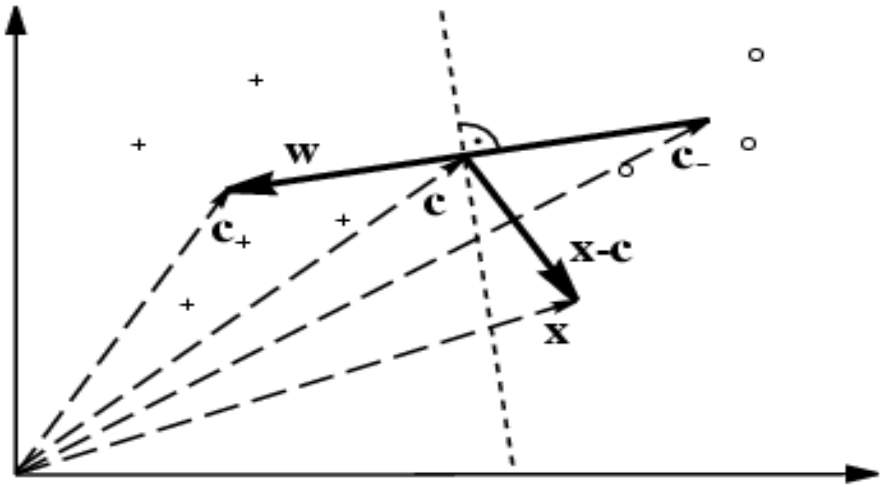
Algorithms

- Algorithms that can be used with inner product information:
  - Parzen's Windows
  - Support Vector Machines
  - Ridge Regression
  - Fisher Linear Discriminant Analysis (LDA)
  - Principal Component Analysis (PCA)
  - Clustering

Parzen's Windows as a Simple Kernel Algorithm

Idea: classify points $\mathbf{x} := \Phi(x)$ in feature space according to which of the two class means is closer.

$$\mathbf{c}_+ := \frac{1}{m_1} \sum_{y_i=1} \Phi(x_i), \quad \mathbf{c}_- := \frac{1}{m_2} \sum_{y_i=-1} \Phi(x_i)$$



Compute the sign of the dot product between $\mathbf{w} := \mathbf{c}_+ - \mathbf{c}_-$ and $\mathbf{x} - \mathbf{c}$.

Source: Schölkopf and Smola, Learning with Kernels, 2002

Cont'd

- where m_1 and m_2 are the number of examples with positive and negative labels, respectively $m_1, m_2 > 0$

$$w = c_+ - c_- = \frac{1}{m_1} \sum_{y_i=+1} \phi(x_i) - \frac{1}{m_2} \sum_{y_i=-1} \phi(x_i)$$

Cont'd

$$f(x) = sign[< w, \phi(x) > + b]$$

$$= sign[< \frac{1}{m_1} \sum_{y_i=+1} \phi(x_i) - \frac{1}{m_2} \sum_{y_i=-1} \phi(x_i), \phi(x) > + b]$$

$$= sign[\frac{1}{m_1} \sum_{y_i=+1} < \phi(x_i), \phi(x) > - \frac{1}{m_2} \sum_{y_i=-1} < \phi(x_i), \phi(x) > + b]$$

$$= sign[\frac{1}{m_1} \sum_{y_i=+1} K(x_i, x) - \frac{1}{m_2} \sum_{y_i=-1} K(x_i, x) + b]$$

$$b = \frac{1}{2}[\frac{1}{m_2^2} \sum_{\{(i,j)|y_i=y_j=-1\}} K(x_i, x_j) - \frac{1}{m_1^2} \sum_{\{(i,j)|y_i=y_j=+1\}} K(x_i, x_j)]$$

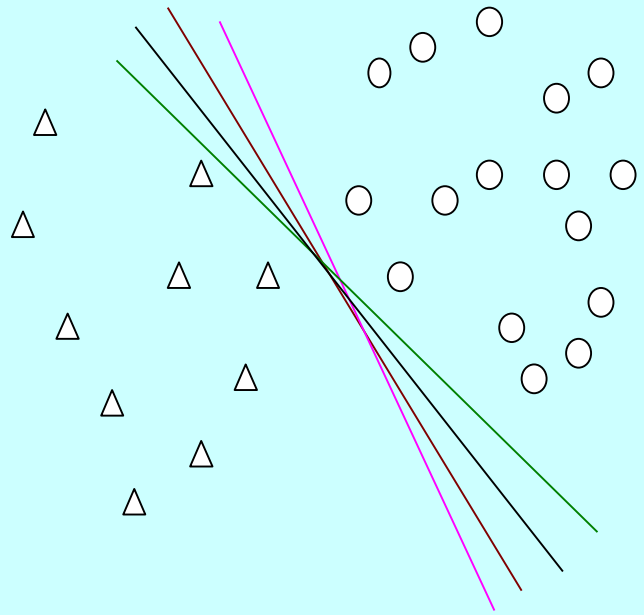
Remarks

- More sophisticated classification algorithms (e.g. SVMs) will be discussed that deviate in the selection of the data points on which the kernels are centered and the choice of weights that are placed on the individual kernels in the decision function.

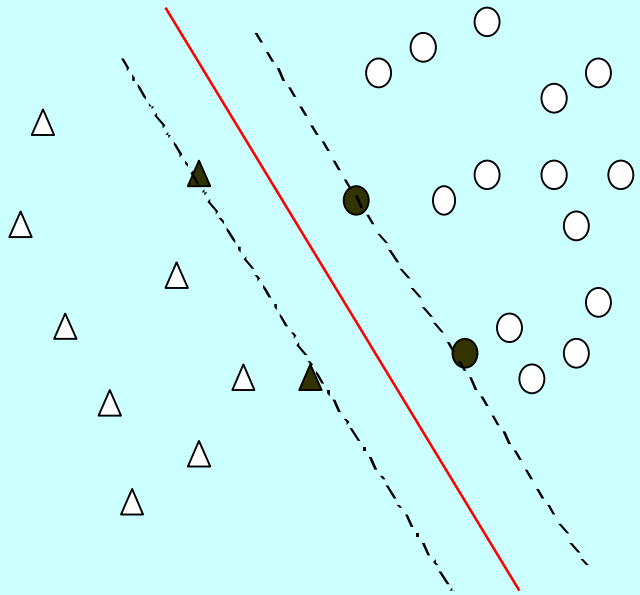
- In SVMs the weights of the kernels will no longer be uniform as in Parzen's windows where the weights are uniform depending on the class to which the pattern belongs.

Support Vector Machines (SVMs)

Separating Hyperplane and Optimal Hyperplane



Separating Hyperplane



Optimal Separating Hyperplane

Linear Two Class SVM and Linear Separable Case

- Assume that we are given a set S of points $\mathbf{x}_i \in R^n$ where each $\mathbf{x}_i$ belongs to either of two classes defined by $y_i \in \{1,-1\}$

- The objective is to find a hyperplane that divides S leaving all the points of the same class on the same side while maximizing the minimum distance between either of the two classes and the hyperplane [Vapnik 1995]

Cont'd

- **Definition 1.** The set S is linearly separable if there exists a $w \in R^n$ and $b \in R$ such that

$$\mathbf{w}\mathbf{x}_i + b \geq 1 \qquad \text{if } y_i = 1,$$
$$\mathbf{w}\mathbf{x}_i + b \leq -1 \qquad \text{if } y_i = -1$$

In order to make each decision surface corresponding to one unique pair (w,b), the following constraint is imposed

$$\min_i \left| wx_i + b \right| = 1$$

Cont'd

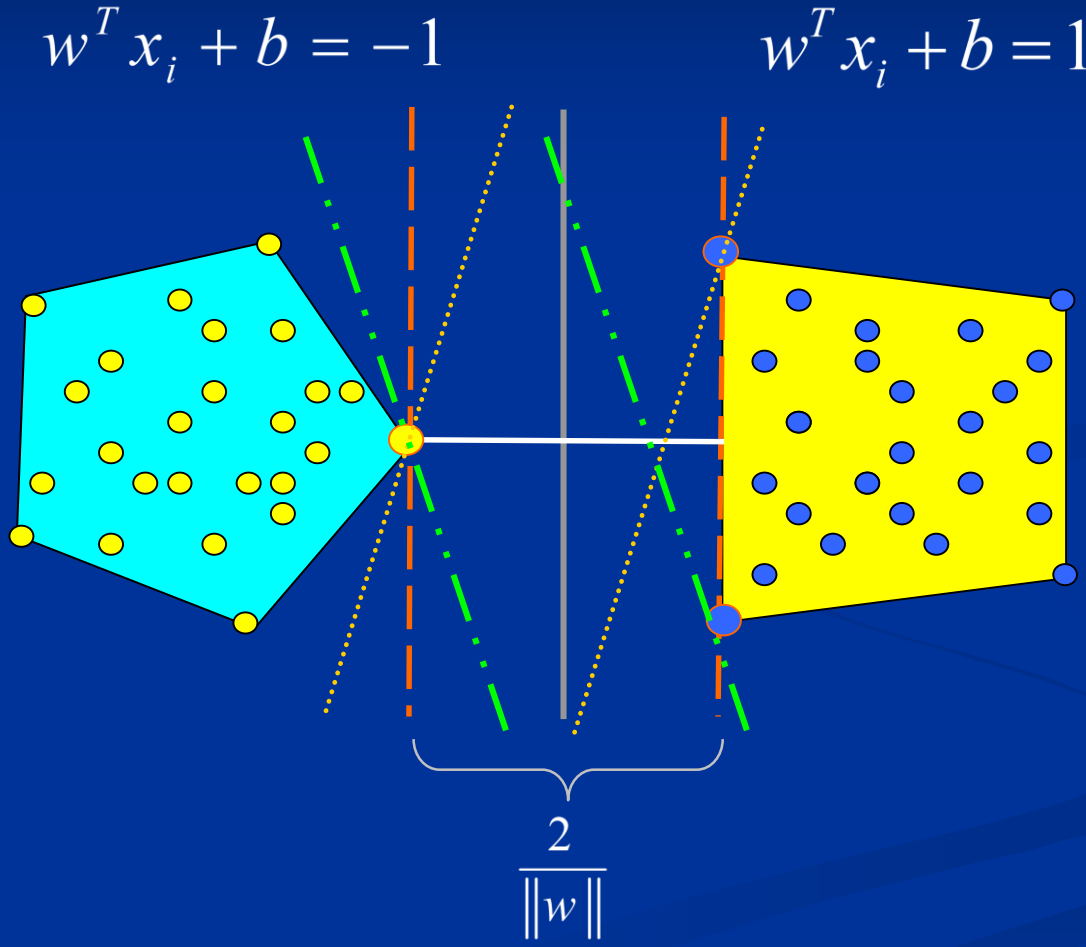
- The distance from a point x to the hyperplane associated to the pair (w,b) is

$$d(x\,;w,b)=\frac{|\,w^T x+b\,|}{\|w\|}$$

- The distance between canonical hyperplane and the closest point is

$$\frac{1}{\|w\|}$$

Maximum Margin Separation



Primal Optimization Problem

$$\min_{\mathbf{w},b} \quad \phi(\mathbf{w}) = \frac{1}{2} \| \mathbf{w} \|^2$$

Subject to

$$y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1$$

$$i = 1,2,\Lambda\ ,l$$

Lagrangian Saddle Point and Optimal Point

- The Lagrangian is

$$L(w,b,\Lambda)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}\lambda_i[y_i(w\cdot x_i+b)-1]$$

- Optimality conditions

$$\frac{\partial L(\mathbf{w},b,\Lambda)}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{l}\lambda_i y_i \mathbf{x}_i=0$$

$$\frac{\partial L(\mathbf{w},b,\Lambda)}{\partial b}=-\sum_{i=1}^{l}\lambda_i y_i=0$$

- Optimal Point

$$w^*=\sum_{i=1}^{l}\lambda_i^* y_i x_i$$

- Support vector: a training vector for which $\lambda_i^*>0$

KKT Conditions

$$\frac{\partial L(w,b,\Lambda)}{\partial w} = w - \sum_{i=1}^{l} \lambda_i y_i x_i = 0$$

$$\frac{\partial L(w,b,\Lambda)}{\partial b} = -\sum_{i=1}^{l} \lambda_i y_i = 0$$

$$y_i (w \cdot x_i + b) - 1 \geq 0$$

$$\lambda_i [y_i (w \cdot x_i + b) - 1] = 0, \qquad \lambda_i \geq 0$$

Dual Optimization Problem

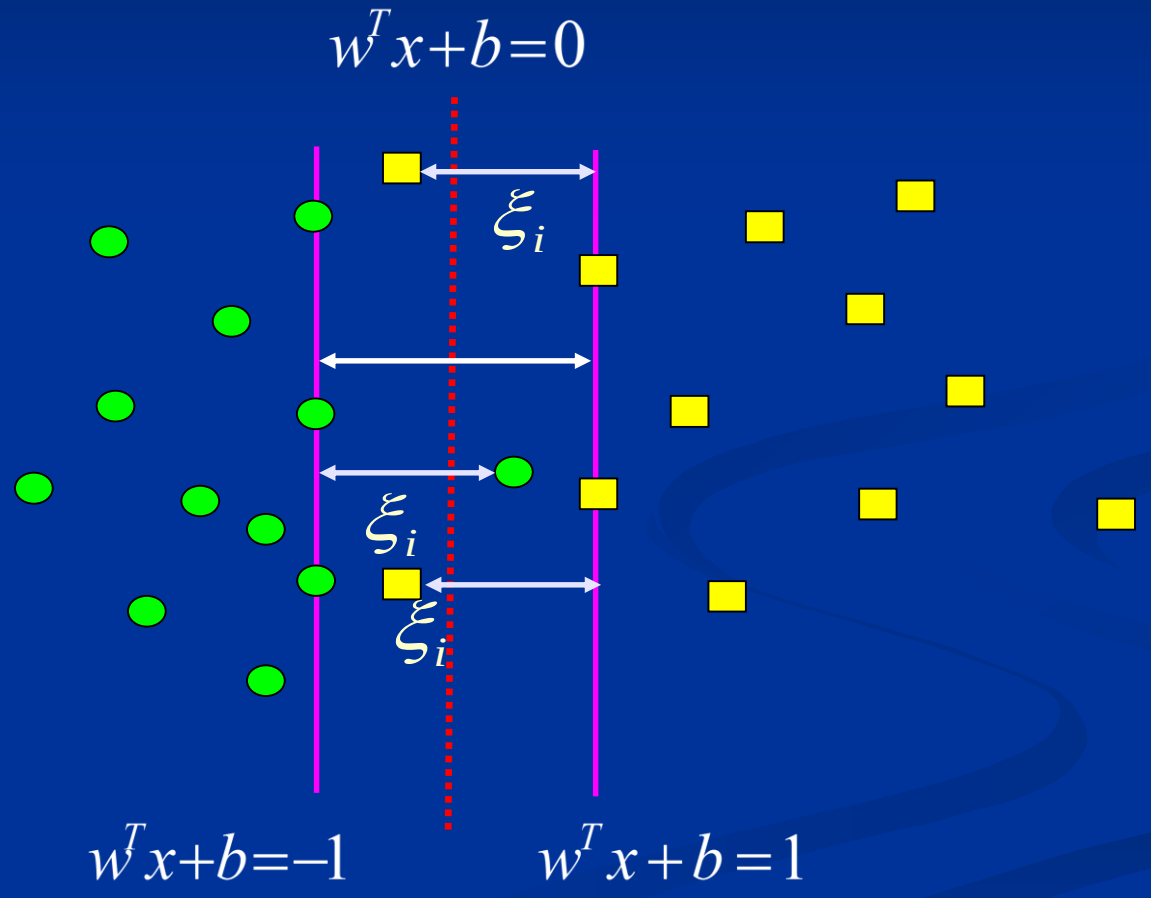
$$\max \quad F(\Lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

Subject to

$$\sum_{i=1}^{l} \lambda_i y_i = 0$$

$$\lambda_i \geq 0, \qquad i = 1,2,\Lambda\ ,l$$

SVMs (nonseparable)



Linearly Non-separable Case (Soft Margin Optimal Hyperplane)

$$\min \quad \phi(w,\xi) = \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{l} \xi_i$$

Subject to

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0,\ i = 1,2,\Lambda\ ,l$$

Lagrangian and Optimality Conditions

$$L(\mathbf{w},b,\Lambda,\xi,\Gamma)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i[y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1+\xi_i]-\sum_{i=1}^{l}\gamma_i\xi_i+C\sum_{i=1}^{l}\xi_i$$

$$\frac{\partial L}{\partial w}=w-\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i=0$$

$$\frac{\partial L}{\partial b}=\sum_{i=1}^{l}\lambda_i y_i=0$$

$$\frac{\partial L}{\partial \xi_i}=C-\lambda_i-\gamma_i=0$$

Dual Problem

$$\max \qquad F(\Lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

Subject to

$$\sum_{i=1}^{l} \lambda_i y_i = 0$$

$$\lambda_i \le C \qquad i = 1,2,\Lambda\ ,l$$

$$\lambda_i \ge 0 \qquad i = 1,2,\Lambda\ ,l$$

Nonlinear Case

- If the data are nonlinear separable, we map the input variable x into a higher dimensional feature space

$$x \rightarrow \phi(x) = (a_1 \phi_1(x), a_2 \phi_2(x), \ldots, a_n \phi_n(x), \ldots)$$

- If we map the input space to the feature space, then we will obtain a hyperplane that separates the data into two groups in the feature space

$$f(\mathbf{x}) = sign(\phi(\mathbf{x}) \cdot \mathbf{w}^* + b^*) \Leftrightarrow f(x) = sign(\sum_{i=1}^{l} \lambda_i^* y_i \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) + b^*)$$

Cont'd

- Kernel function

$$K(x,y) = \phi(x) \cdot \phi(y) = \sum_{n=1}^{\infty} a_n^2 \phi_n(x) \cdot \phi_n(y)$$

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{l} \lambda^*_i y_i K(\mathbf{x}, \mathbf{x}_i) + b^*\right)$$

Dual problem in nonlinear case

- Replace the dot product of the inputs with the kernel function in the nonseparable case

$$\max \quad F(\Lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Subject to

$$\sum_{i=1}^{l} \lambda_i y_i = 0$$

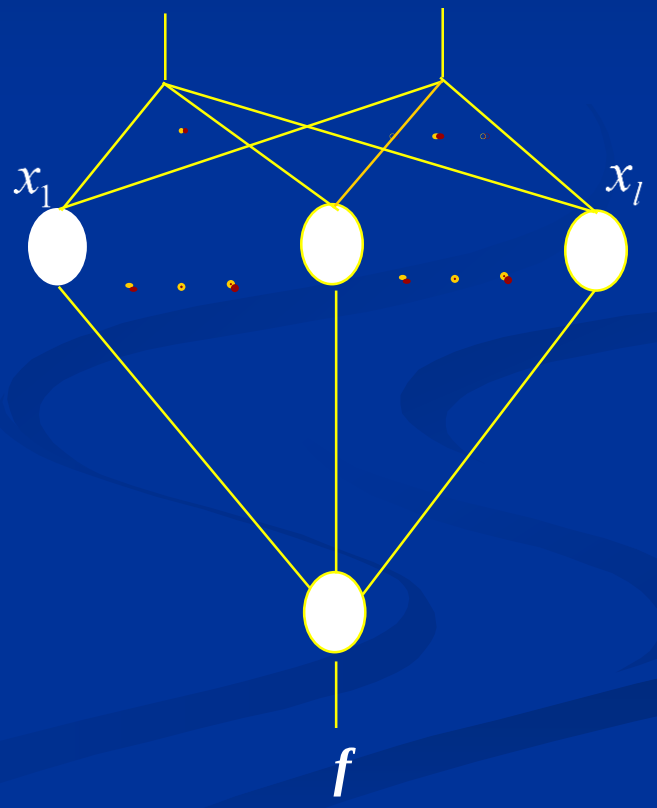
$$\lambda_i \le C \qquad i = 1,2,\Lambda\ ,l$$

$$\lambda_i \ge 0, \qquad i = 1,2,\Lambda\ ,l$$

Equivalence to Networks

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \lambda_i K(\mathbf{x}, \mathbf{x}_i) + b$$

and can be "written" as the SVM network



Kernel Functions in SVMs

- An inner product in feature space has an equivalent kernel in input space

$$K(x, y) = \phi(x) \cdot \phi(y) = \sum_{i=1}^{\infty} a_i^2 \, \phi_i(x) \cdot \phi_i(y)$$

- Any symmetric positive semi-definite function (Smola, 1998), which satisfies the Mercer's Conditions can be used as kernel function in the SVM context. Mercer's Conditions can be written as

$$\int\int K(x, y)\, g(x)\, g(y)\, dxdy \geq 0, \quad \int g^2(x)\, dx < \infty$$

Some Kernel Functions

- Polynomial Type: $K(x,y) = ((x \cdot y) + 1)^d, \; d = 1,2,.....$
- Gaussian Radial Basis Function (GRBF):

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

- Exponential Radial Basis Function:

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

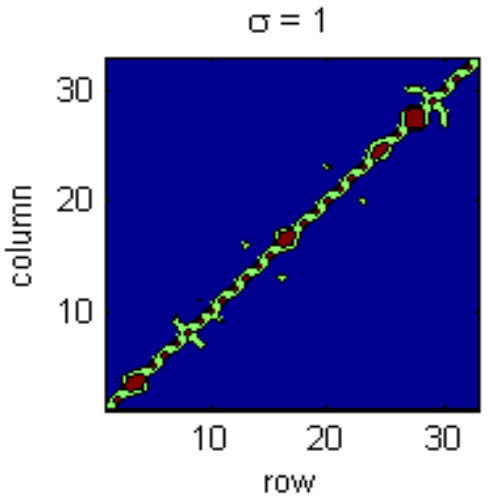
- Multi-Layer Perceptron:

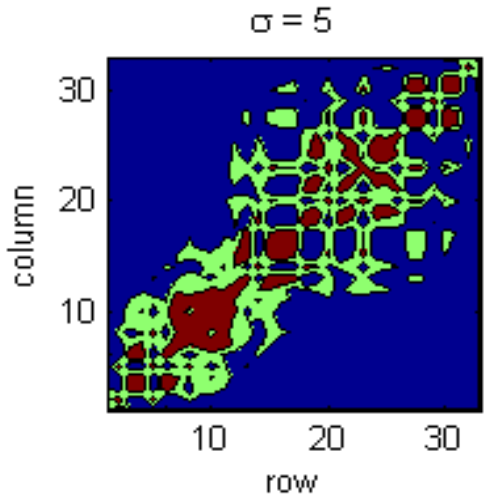
$$K(x,y) = tanh\ (\phi(x \cdot y) + \theta)$$

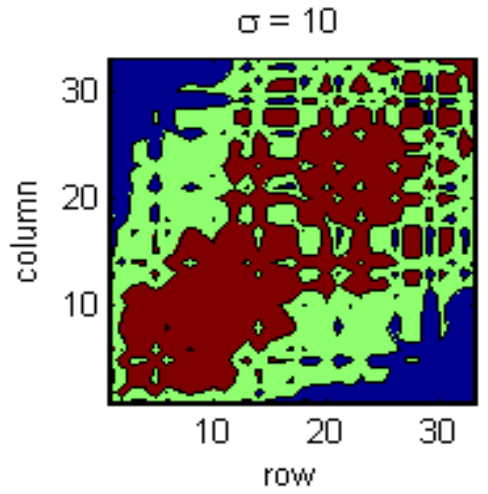
- Fourier Series:

$$K(x,y) = \frac{sin\ ((N + 0.5)(x - y))}{sin\ (0.5(x - y))}$$

Contour Plots of A Kernel Matrix







Contour plots of the values of a Gaussian kernel matrix.

High values are in red and low values in blue.

Data were randomly generated and sorted according to the ascendig order of the first feature.

What we have achieved?

- Replaced problem of NN architecture by kernel definition
- More natural
- Can be applied to non-vectorial data
- Gained more flexible generalization control
- No local minima (convex optimization)

Separating Hyperplanes (Source: Hastie et al., 2004)

- Mixture_linear
- Mixture_medium
- Mixture_rough
- Mixture_smooth
- Small_balanced_overlap
- Small_overlap
- Small_separated
- Small_unbalanced

XOR Problem (Nonlinear Separable Case)

x_1	x_2	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	1

- We map the input variable x into a higher dimensional feature space

$$\phi: R^2 \rightarrow R^3$$

$$(x_1, x_2) \ \alpha \ (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

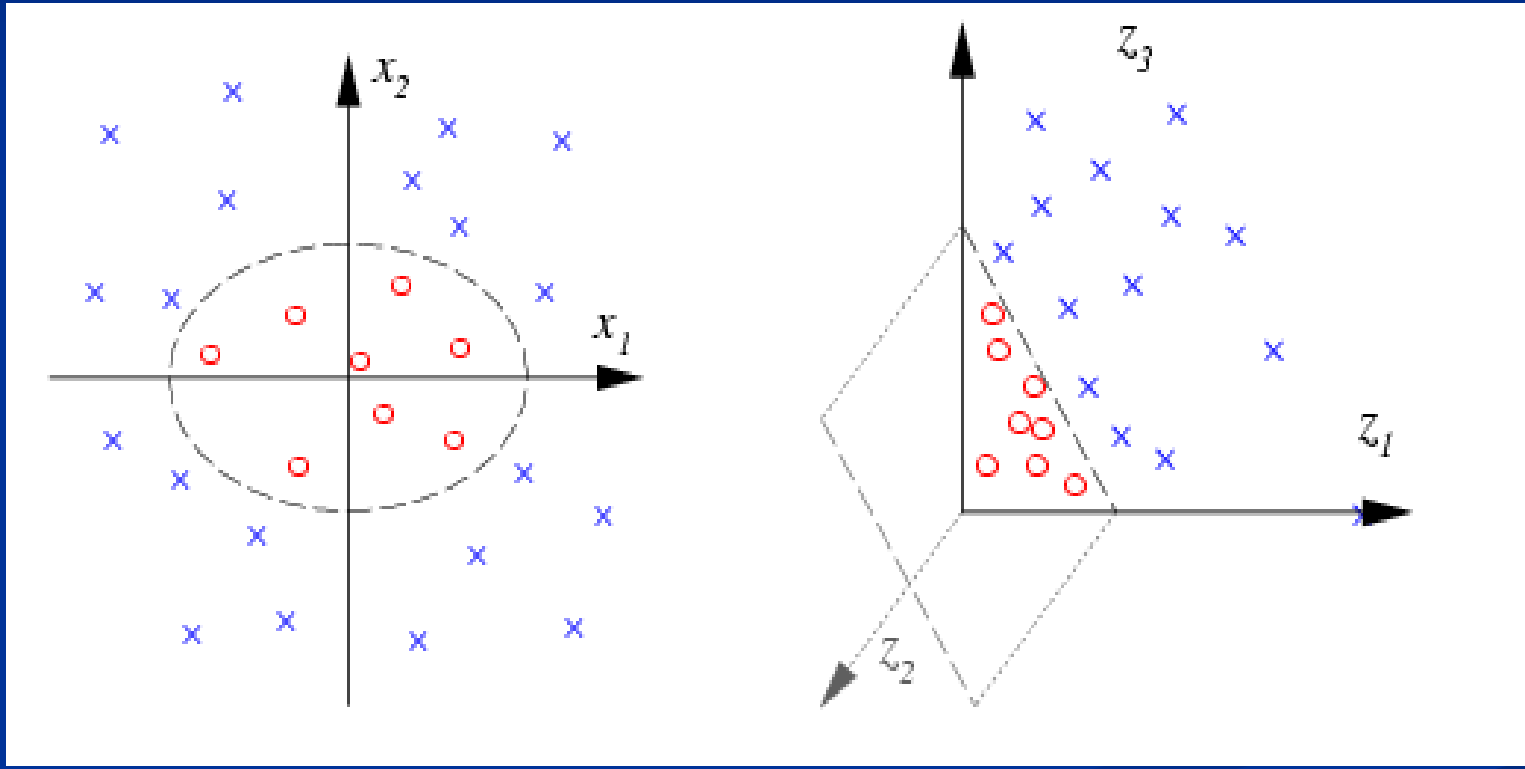
$$(1, \ 1) \ \alpha \ (1, \sqrt{2}, \ 1)$$

$$(1, -1) \ \alpha \ (1, -\sqrt{2}, 1)$$

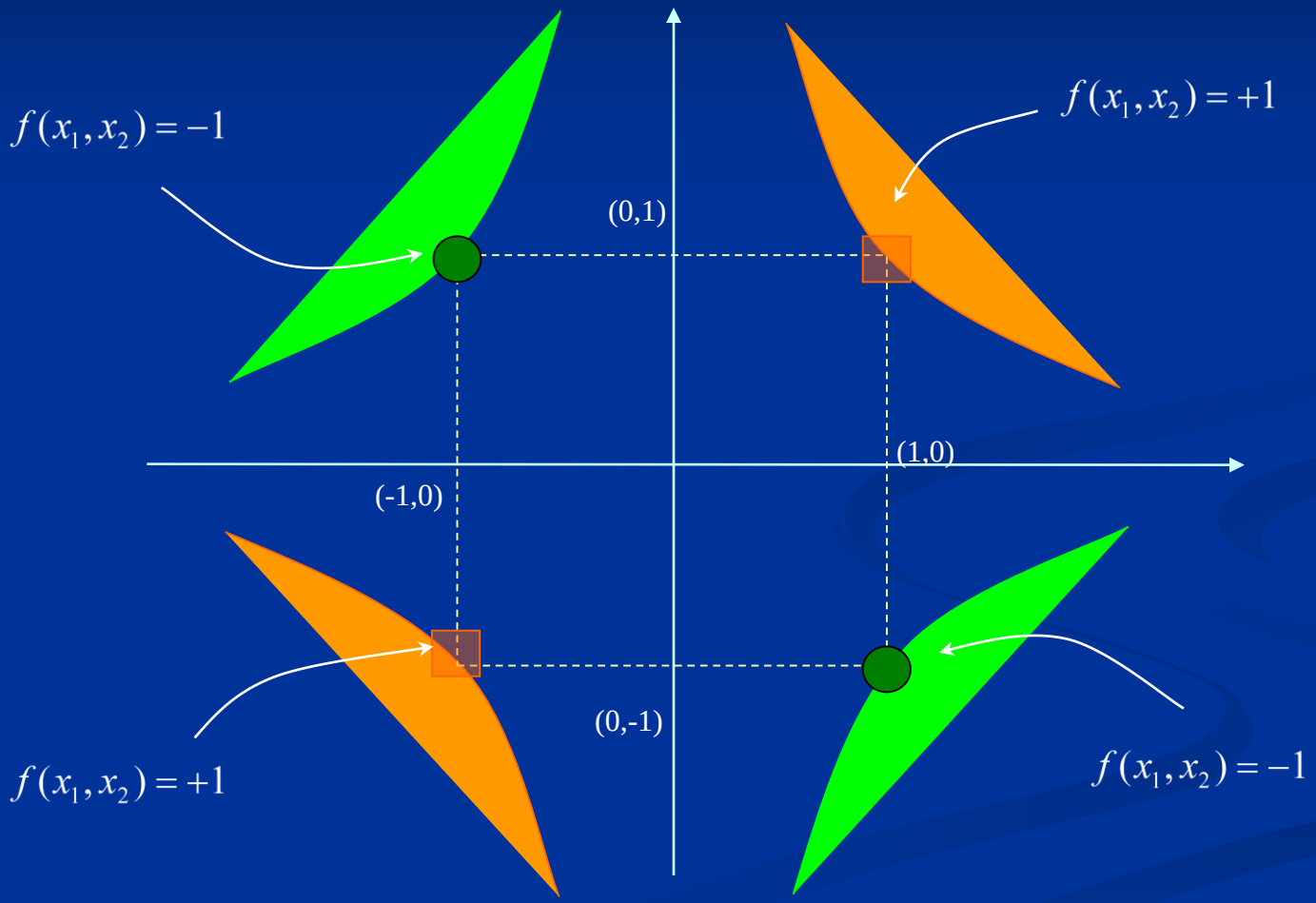
$$(-1, \ 1) \ \alpha \ (1, -\sqrt{2}, 1)$$

$$(-1, -1) \ \alpha \ (1, \sqrt{2}, \ 1)$$

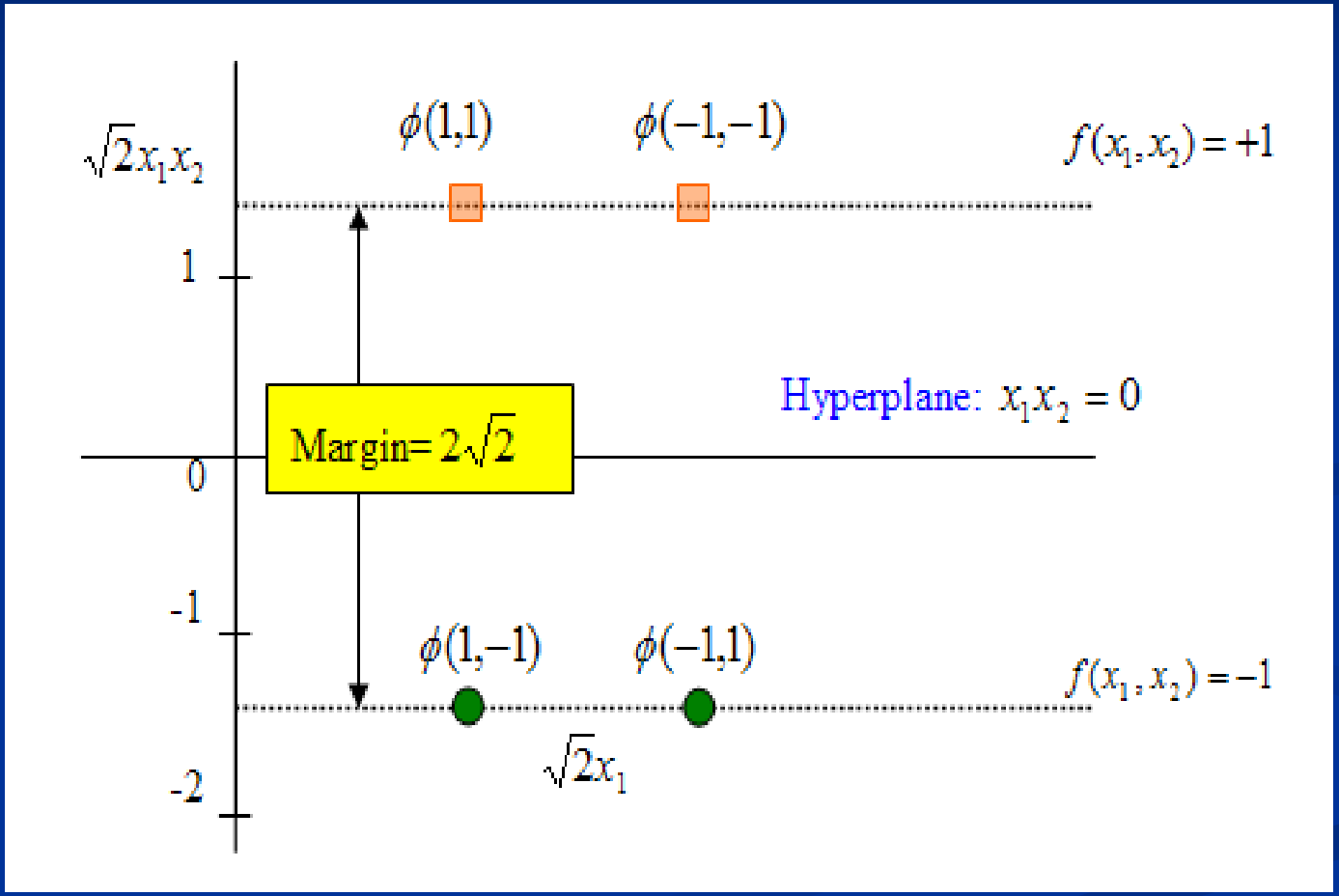
Cont'd



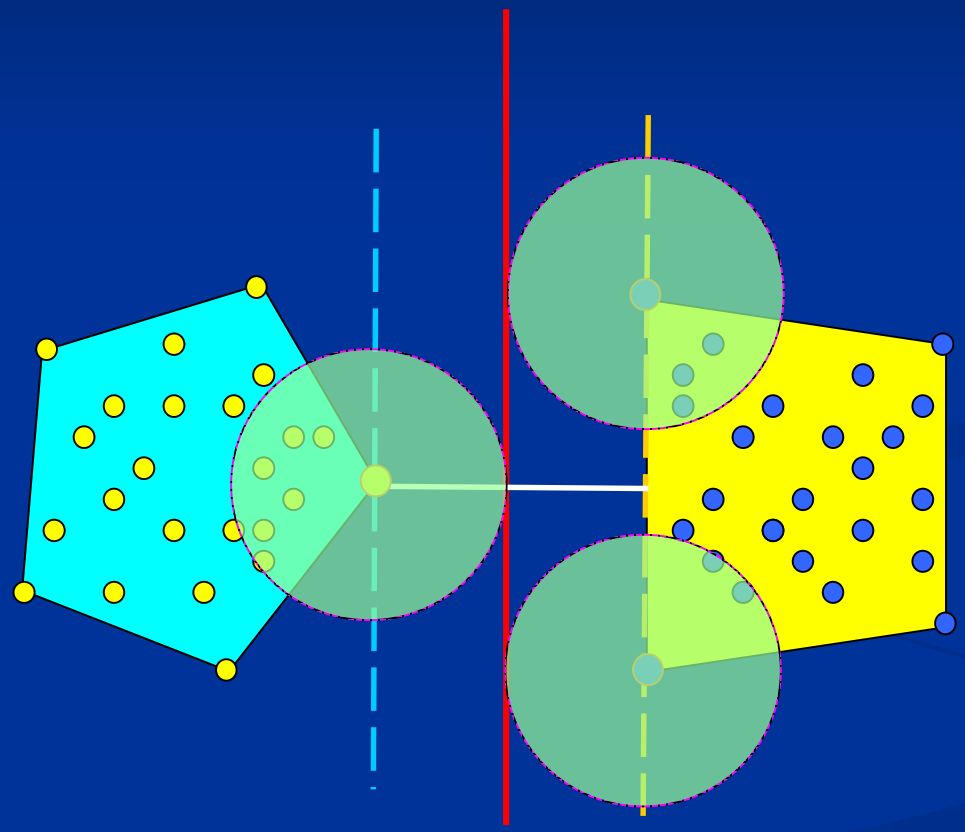
Classifier in Input Space



Classifier in Feature Space



Robust Support Vector Machines with Data Uncertainty



- Permit noise of data
- SVM solutions are robust for perturbation of data

Cont'd

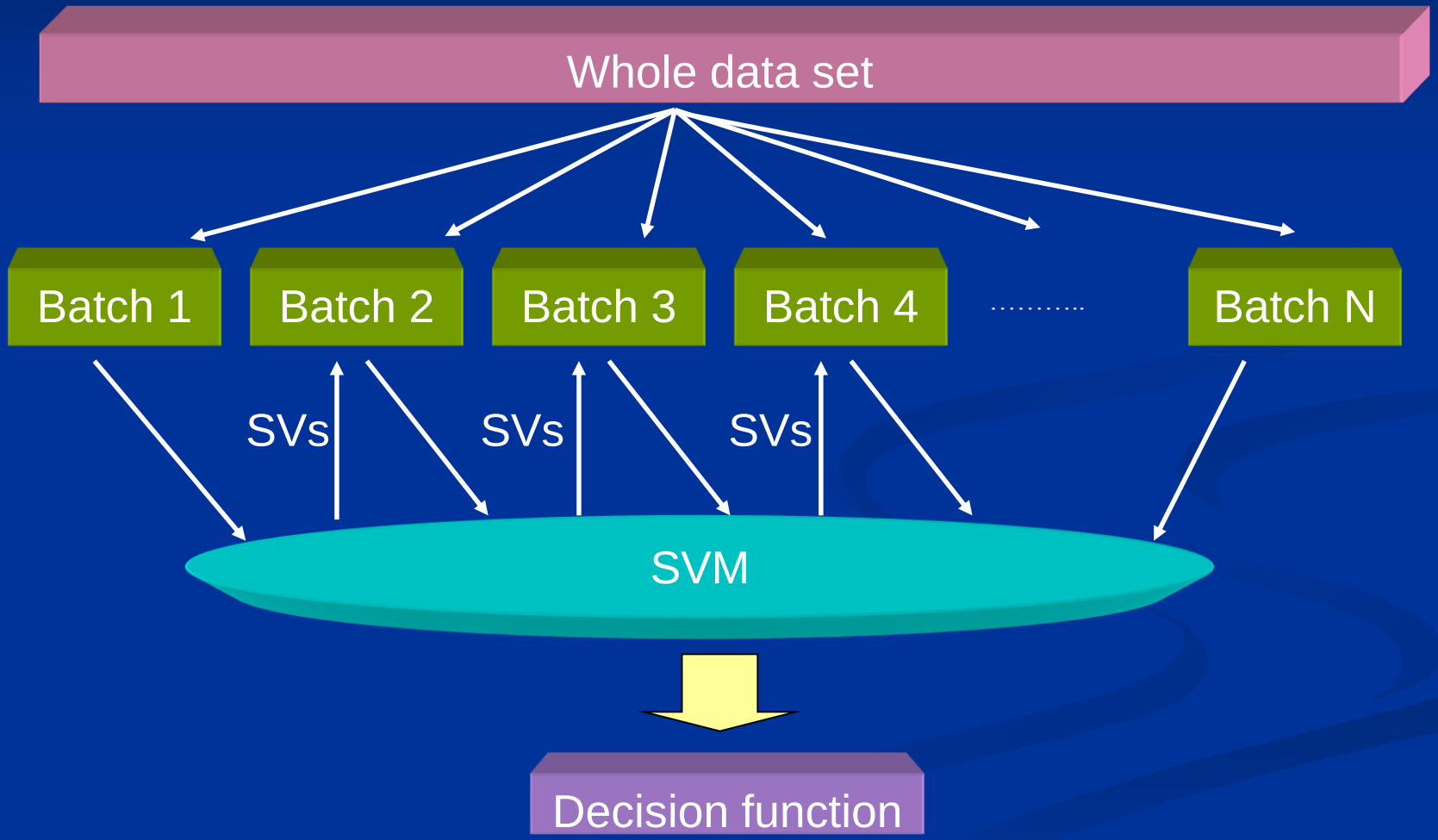
$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$

$$subject\ \ to$$

$$y_i\,(w^T x_i + b) - \eta\|w\| \geq 1$$

$$i = 1, \Lambda\ , l$$

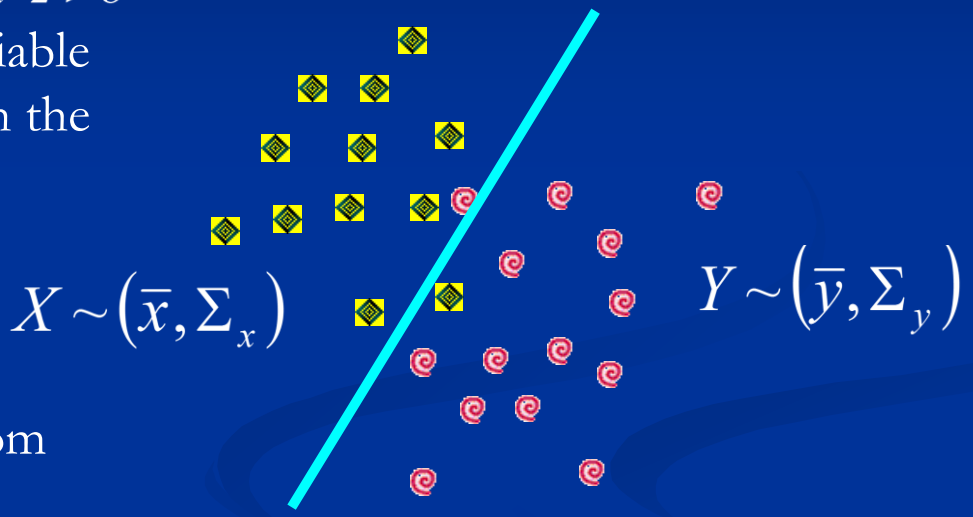
Incremental SVMs



MPM: Problem Description

Given data samples from two different classes. Find $\{a \in R^n, b \in R\}$ such that if $a^T z > b$ then z is identified with the random variable x, an if $a^T z < b$ then z is identified with the random variable y

$a^T z = b$: decision hyperplane



Notation:

Let $x \in R^n$ and $y \in R^n$ denote random vector with

$$E(x) = \bar{x}, \quad E(y) = \bar{y}$$

$$E((x - \bar{x})(x - \bar{x})^T) = \Sigma_x$$

$$E((y - \bar{y})(y - \bar{y})^T) = \Sigma_y$$

- $X \sim (\bar{x}, \Sigma_x)$ will denote the set of distributions which has the mean $\bar{x}$ and covariance Σ_x

source: Lanckriet et al, 2002

MPM

- MPM approach was introduced by Lanckriet et al. (2002)
- Minimizing the maximum probability of misclassification of the future data points

$$\max_{\alpha, a \neq 0, b} \quad \alpha$$

$$Subject\ \ to$$

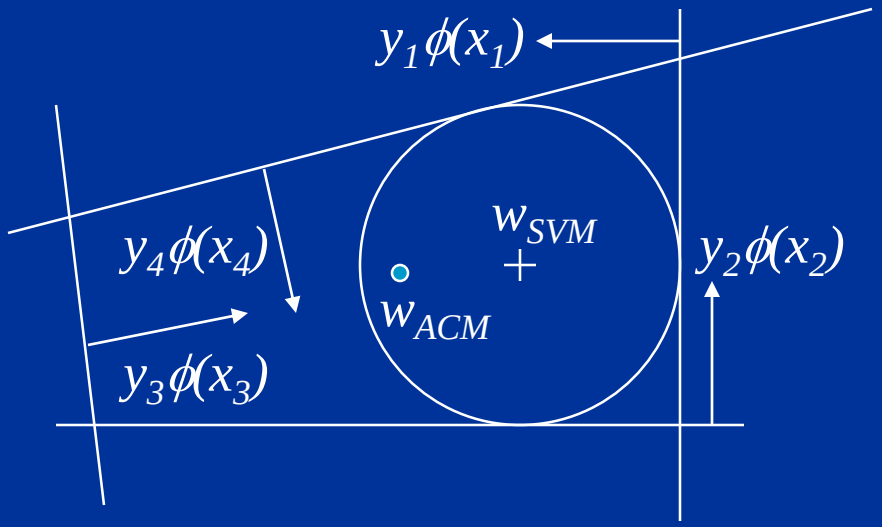
$$\inf_{x \sim (\bar{x}, \Sigma_x)} \Pr\{a^T x \geq b\} \geq \alpha$$

$$\inf_{y \sim (\bar{y}, \Sigma_y)} \Pr\{a^T y \leq b\} \geq \alpha$$

Analytic Center Machines

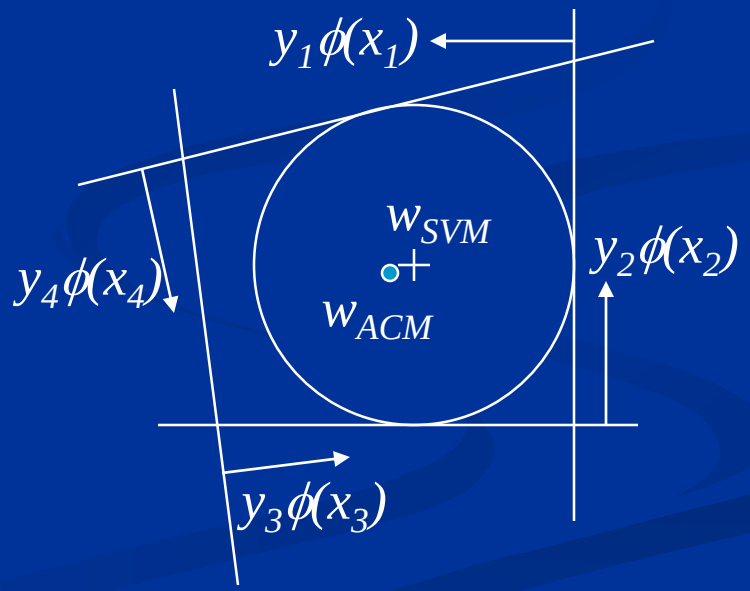
Elongated version space :

$w_{ACM} \neq w_{SVM}$



"Normal" version space :

$w_{ACM} \approx w_{SVM}$



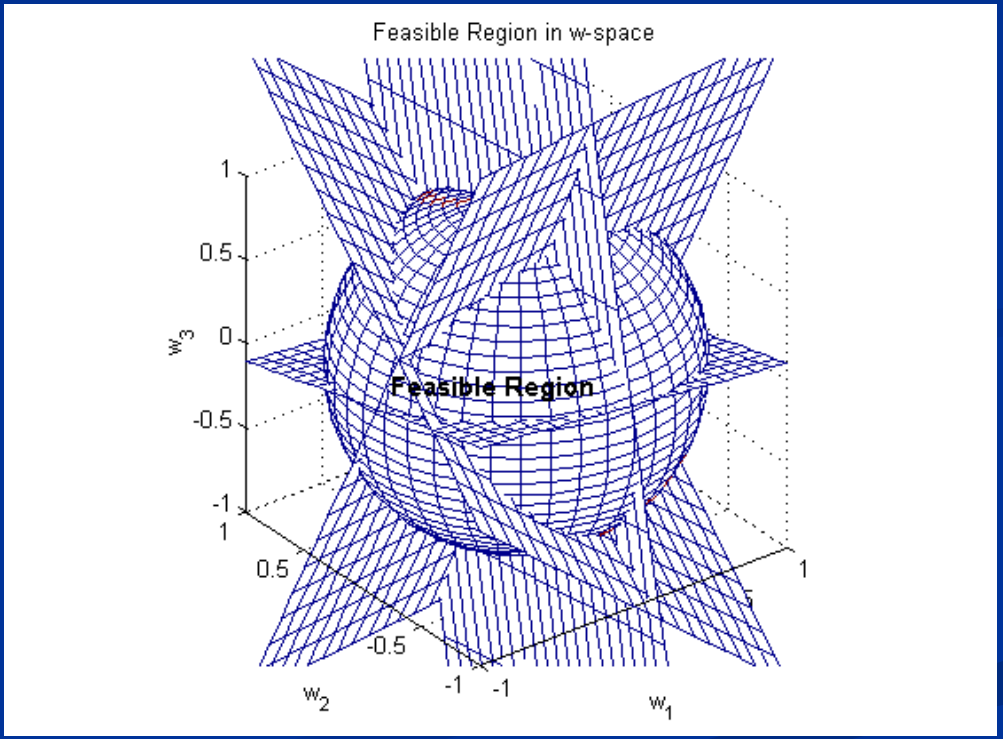
Source: Trafalis and Malyscheff, An Analytic Center Machine, *Machine Learning*, 2002

Cont'd

$$\min \quad -\sum_{i=1}^{l} \ln\left(y_i \cdot \left(\tilde{x}_i^{\,T} \tilde{w}\right)\right)$$

Subject to

$$\frac{1}{2}\tilde{w}^T \tilde{w} = 1$$



where

$\tilde{w}^T = \left(w^T, b\right)$ and $\tilde{x}_j^{\,T} = \left(x_j^{\,T}, 1\right)$

$y_j \cdot \left(\tilde{x}_j^{\,T} \tilde{w}\right) \geq 0, \ \forall j = 1,\ldots,l$

$\tilde{x}_j, \tilde{w} \in \Re^{d+1}$ and $y_j \in \Re$ (b is like an additionalweight)

Other Topics

- Bayesian Kernel Methods
- Kernel Feature Extraction
- Kernel Principal Component Analysis (KPCA)
- Kernels for structured data (text, strings, trees, etc.)
- Optimization methods with large scale data mining problems

Suggested Reading

- V. N. Vapnik. **The Nature of Statistical Learning Theory.** Springer, 1995.
- V. N. Vapnik. **Statistical Learning Theory.** Wiley, 1998.
- B. Schölkopf and A. J. Smola. **Learning with Kernels**. MIT press, 2002.
- J. Shawe-Taylor and N. Cristianini. **Kernel Methods for Pattern Analysis**. Cambridge, 2004.
- N. Cristianini and J. Shawe-Taylor. **Introduction To Support Vector Machines.** Cambridge, 2000.
- T. Evgeniou and M. Pontil and T. Poggio. **Regularization Networks and Support Vector Machines.** Advances in Computational Mathematics, 2000.
- F. Cucker and S. Smale. **On The Mathematical Foundations of Learning.** Bulletin of the American Mathematical Society, 2002.

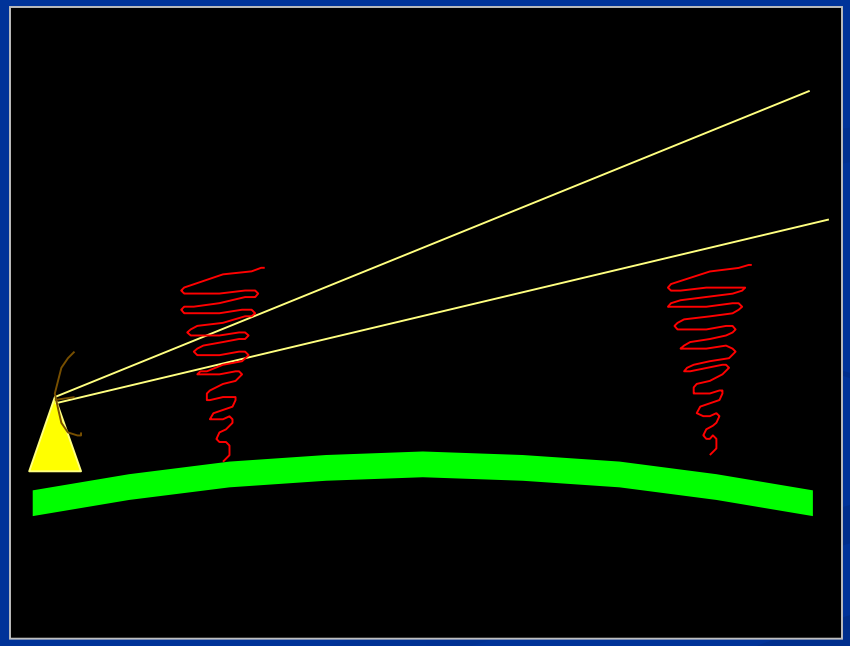
Part II :
Weather Applications

How do we begin?

- Data!
- Radar data provide the best observational tool for gathering weather information
- Weather Surveillance Radar – 1988 Doppler (WSR-88D)

Radar Horizon

- Phenomena of similar sizes are not necessarily resolvable at near and far ranges.



Radar continued

- Information we get from radar data
  - Reflectivity (Z): ratio of the radiant energy reflected by a given surface to the total incident energy
  - Velocity (V): velocity of target
  - Spectrum Width (W): measure of dispersion of velocities within the radar sample

Prediction of Rainfall From WSR-88D Radar Using Support Vector Regression (SVR) and Least Squares SVR

Introduction

- Flash floods kill more people than any other weather phenomenon
- Our ability to estimate precipitation and flooding from current state of the science technology is frequently inaccurate and can be improved. Existing techniques, known as Z-R relations, used to estimate rainfall rates are based on empirical fits to radar reflectivity. These are known to be inaccurate in very light and very heavy rain situations
- By using SVR and LS-SVR we want to utilize the native variables from the WSR-88D, namely reflectivity (Z) to predict rainfall. It may be possible to incorporate additional information into the forecasts
- SVR is being used as it has a property to generalize well with lower error as compared to traditional regression techniques

Objectives

- Utilize Support Vector Regression (SVR) and Least Squares SVR, to predict rainfall in Chandler, OK based on WSR-88D
- Compare the forecasts of rainfall given by SVR and LS-SVR to that of traditional regression
- Compare SVR to existing Z-R relation

Problem Statement

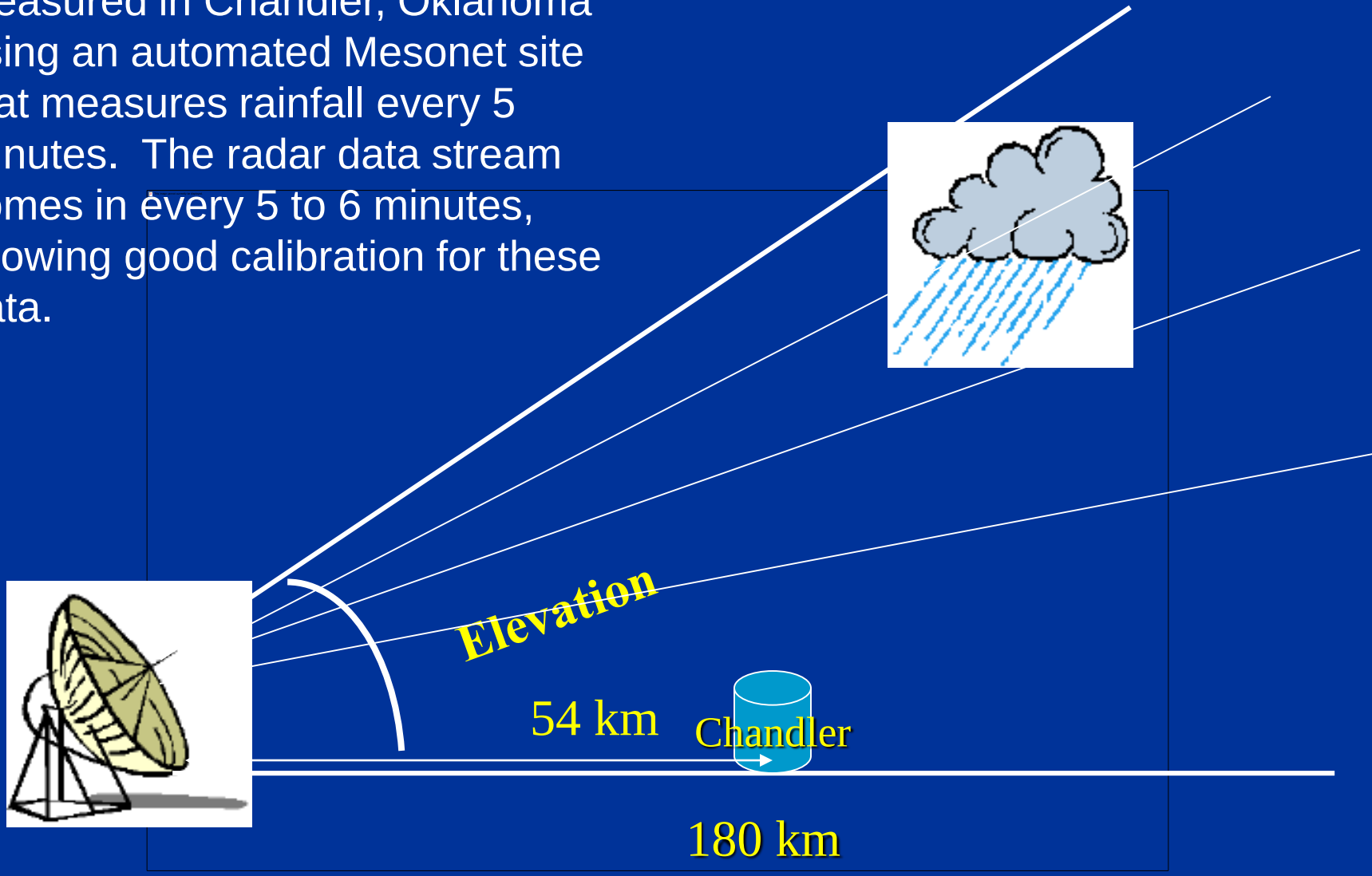
- The most common form of the rainfall rate (RR) and reflectivity (Z) is the empirical relationship:

$$RR=(0.036)(10)^{(0.0625)(Z)}$$

- WSR-88D records digital database containing 3 variables: velocity (V), reflectivity (Z), and spectrum width (W)

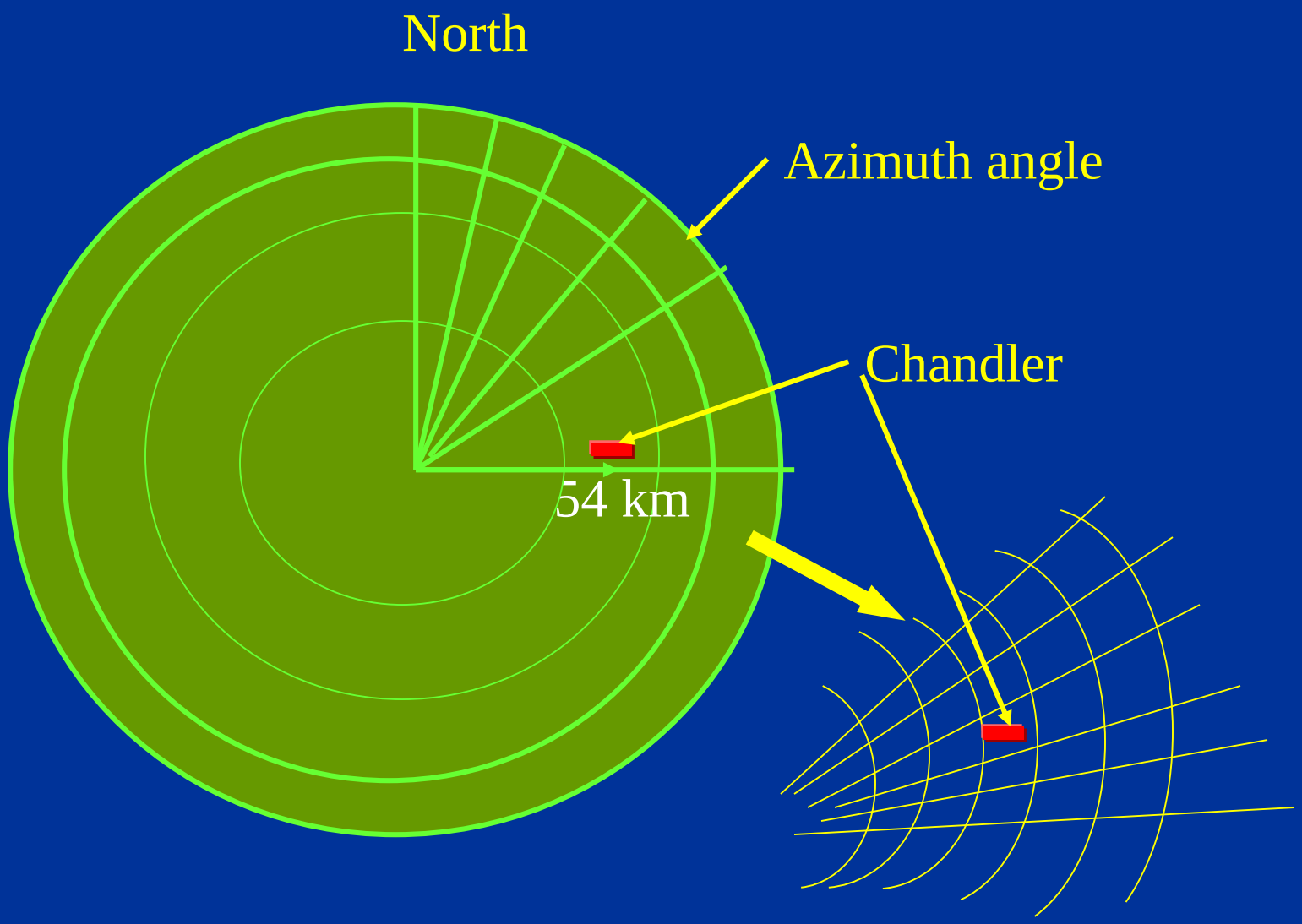
- The primary focus of the research is to capitalize on Z.

Radar is located in Norman, Oklahoma and the rainfall is measured in Chandler, Oklahoma using an automated Mesonet site that measures rainfall every 5 minutes. The radar data stream comes in every 5 to 6 minutes, allowing good calibration for these data.



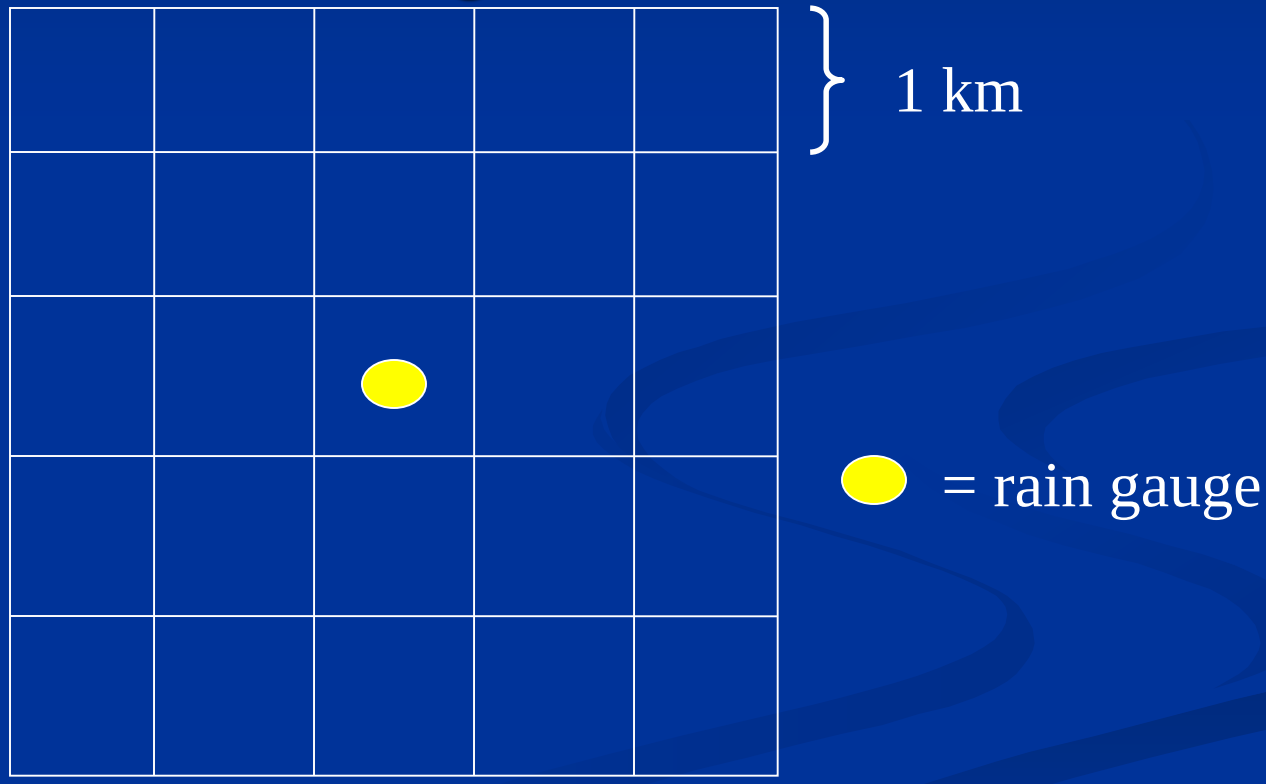
Details on Radar Data Used

The empirical formula (Z-R) uses reflectivity from the lowest scan (elevation) angle. The SVM approach uses a multi-level approach with 5 elevation angles to better sample the vertical storm environment. Additionally, the SVM approach uses a 5 x 5 set of azimuth bins from adjacent areas above Chandler to better sample the horizontal storm environment.


North
Azimuth angle
Chandler
54 km

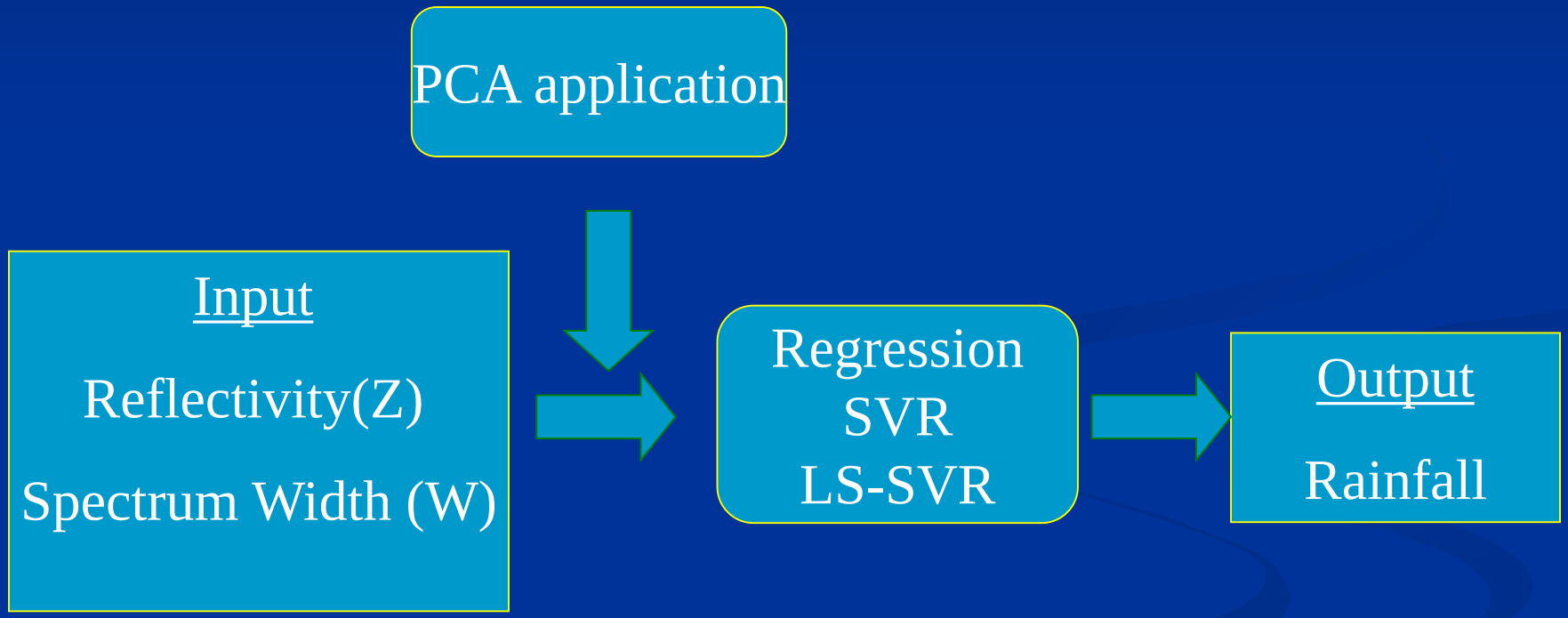
Retrieving Data

- 5x5 grid of points centered on rain gauge for each elevation angle



(25 boxes) x (5 levels) = 125 input components for each data point

Methodology



Formulation of SVR

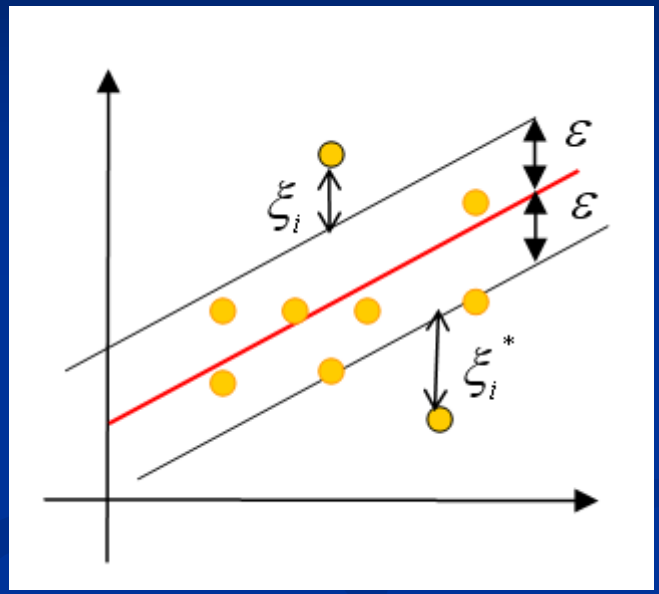
- Min $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$

Subject to

$$y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \le \varepsilon + \xi_i$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$



- *ε- insensitive loss function*

$$L(f(x)-y) = \begin{cases} |f(x)-y| - \varepsilon, & \text{if } |f(x)-y| > \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

Dual SV Regression Problem

- Using Lagrangian duality we obtain the following quadratic SV regression problem

$$\max \quad -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)K(\mathbf{x}_i,\mathbf{x}_j)-\varepsilon\sum_{i=1}^{l}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{l}y_i(\lambda_i-\lambda_i^*)$$

Subject to

$$\sum_{i=1}^{l}(\lambda_i-\lambda_i^*)=0$$

$$\lambda_i,\lambda_i^*\in[0,C]$$

- At the optimal solution, we obtain

$$f(x)=\sum_{i=1}^{l}(\lambda_i-\lambda_i^*)K(x_i,x)+b$$

Least Square-Support Vector Regression (Suykens)

- Primal formulation

$$\min_{w,b,e} \quad J(w,e) = \frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{\lambda} e_k^2$$

Subject to

$$y_i = w^T \varphi(x_i) + b + e_i,\ i = 1,\ldots\lambda$$

- Lagrange formulation

$$L(w,b,e;\alpha) = J(w,e) - \sum_{i=1}^{\lambda} \alpha_i \{w^T \varphi(x_i) + b + e_i - y_i\}$$

Cont'd

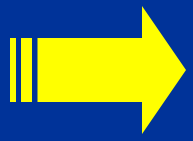
- KKT Optimality Conditions

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{\lambda} \alpha_i \varphi(x_i)$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{\lambda} \alpha_i = 0$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i,$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0,$$



$$\left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & K + \frac{1}{C} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right]$$

where $y = [y_1,\ldots,y_\lambda]$, $1_v = [1_1,\ldots,1_\lambda]$, $\alpha = [\alpha_1,\ldots, \alpha_\lambda]$ and $K_{ij} =< \varphi(x_i), \varphi(x_j) >$ for $i, j = 1,\ldots, \lambda$.

Results

MSE of training and testing for traditional regression, SVR, LS-SVR, Formula (mm2)

Method	Predictors	Training Average MSE	Testing Average MSE
Traditional Regression	PC Z	0.3494	0.4665
ANNs	PC Z	0.1957	0.3383
SVR	PC Z	0.1507	0.3099
LS-SVR	PC Z	0.1754	0.2709
Formula	Z	0.2887	0.3728

Conclusions

- For rainfall estimation, the LS-SVR outperforms the ANN by over 50% in the R^2 statistic and over 27% in MSE reduction
- LS-SVR R^2 is 50.6% higher than for the rainfall rate (RR) formula
- For rainfall detection, the SVR method has 11.9% more skill than LS-SVR and 24.8% more skill than the RR Formula.

References

- Trafalis, T.B., B. Santosa and M. B. Richman, “Learning networks in rainfall estimation”, *Computational Management Science, pp. 229-251, 2005.*
- Trafalis, T. B., M. B. Richman and B. Santosa, "Prediction of Rainfall from WSR-88D Radar Using Kernel-based Methods", *International Journal of Smart Engineering System Design*, **2003**.
- Trafalis, T.B., M. Richman, and B. Santosa, “Prediction of Rainfall from WSR-88D Radar Using Support Vector Regression”, Book Published of Collection: Intelligent Engineering Systems Through Artificial Neural Networks, (C.H. Dagli, A.L. Buczak, J. Ghosh, M.J. Embrechts, O. Ersoy, and S.W. Kercel, eds.), ASME Press, 12:639-644, 2002.
- Trafalis, T.B., M. Richman, A. White, and B. Santosa, “Data Mining Techniques for Improved WSR-88D Rainfall Estimation”, *Computers and Industrial Engineering*, 43:775-786, 2002.