Introduction to Support Vector Machine (SVM)

Introduction

-- SVM classification

- Support vector machines (SVMs) developed by Vapnik, have gained wide acceptance because of their high generalization ability for a wide range of applications.
- There are many variations of SVM, including the soft margin classifier, adaptive margin classifier, and so on.
- Even though the two classes divided by the margin are slightly overlapped and noise exists, they all have the common property that the constructed hyper plane effectively separates two classes.

Introduction

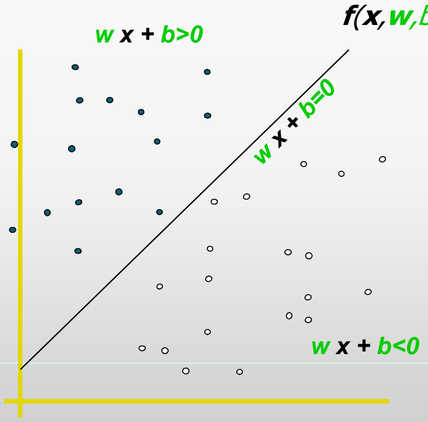
-- SVM classification

- Linear classifier

• denotes +1

◦ denotes -1

$f(\mathbf{x},w,b) = sign(w\,\mathbf{x} + b)$



How would you classify this data?

3

Introduction

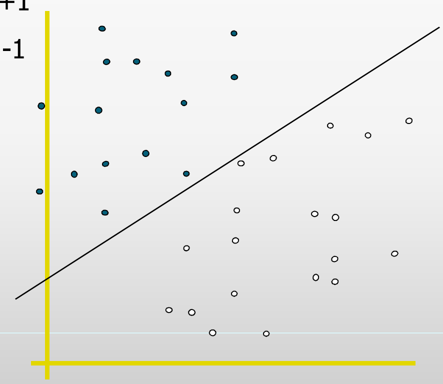
-- SVM classification

- Linear classifier

• denotes +1

◦ denotes -1

$f(\mathbf{x},w,b) = sign(w\,\mathbf{x} + b)$



How would you classify this data?

4

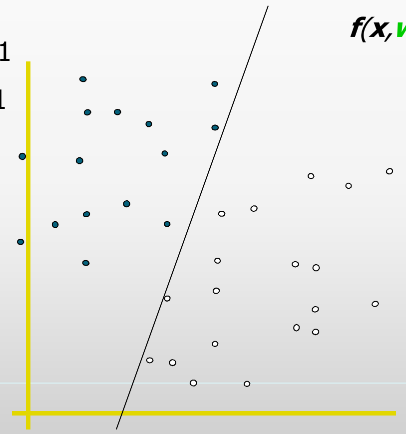
Introduction

-- SVM classification

- Linear classifier

• denotes +1

° denotes -1



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$

How would you classify this data?

5

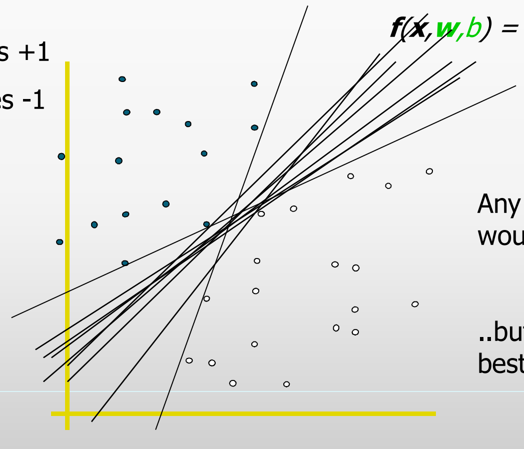
Introduction

-- SVM classification

- Linear classifier

• denotes +1

° denotes -1



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$

Any of these would be fine..

..but which is best?

6

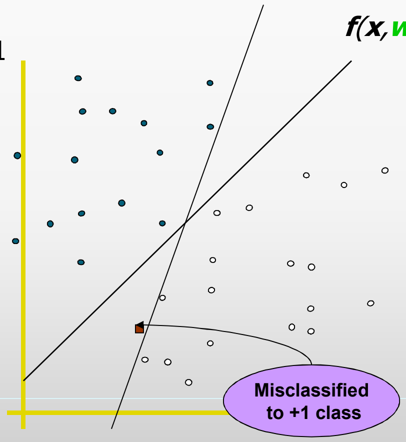
Introduction
-- SVM classification

- Linear classifier

• denotes +1

◦ denotes -1

f(**x**,**w**,*b*) = *sign*(**w x** + *b*)



How would you classify this data?

7

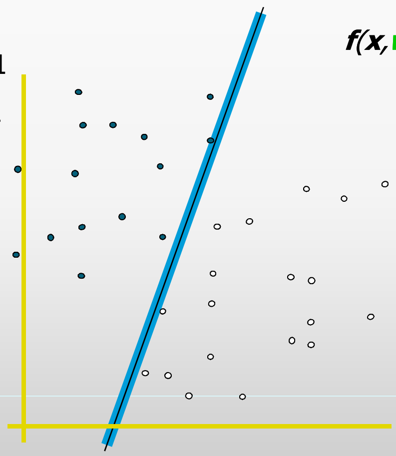
Introduction
-- SVM classification

- Linear classifier

• denotes +1

◦ denotes -1

f(**x**,**w**,*b*) = *sign*(**w x** + *b*)



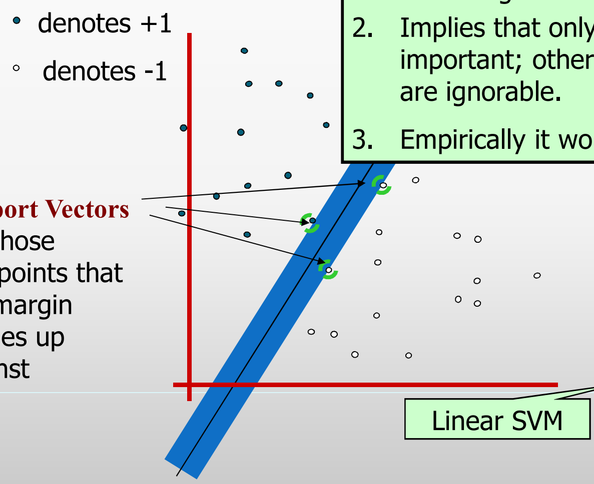
Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

8

Introduction

-- SVM classification

- Linear classifier



1. Maximizing the margin is good according to intuition and PAC theory
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear **classifier** is the linear classifier with the, um, maximum margin.

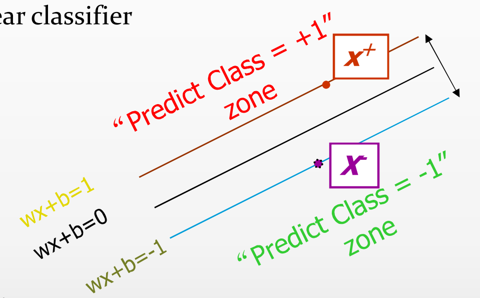
This is the simplest kind of SVM (Called an LSVM)

9

Introduction

-- SVM classification

- Linear classifier



M=Margin Width

What we know:

- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$
- $\boldsymbol{w} \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-) = 2$

$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

10

Introduction

-- SVM classification

- Goal: 1) Correctly classify all training data

$$wx_i + b \geq 1 \qquad \text{if } y_i = +1$$
$$wx_i + b \leq -1 \qquad \text{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \qquad \text{for all i}$$

2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$

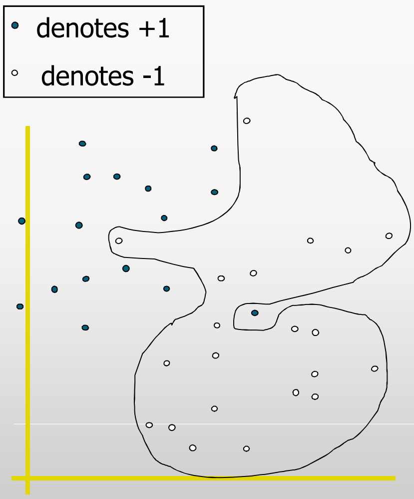
- We can formulate a Quadratic Optimization Problem and solve for w and b
-

Minimize	$\Phi(w) = \frac{1}{2} w^t w$
subject to	$y_i(wx_i + b) \geq 1 \quad \forall i$

11

Introduction

-- SVM classification



- **Hard Margin:** So far we require all data points be classified correctly
  - No training error
- **What if the training set is noisy?**
  - **Solution 1:** use very powerful kernels

OVERFITTING!

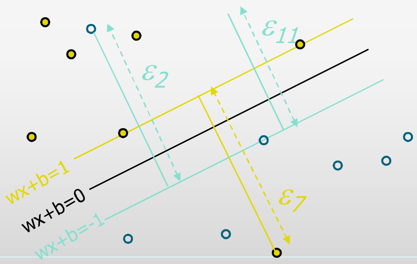
12

Introduction

-- SVM classification

- Non-separable classifier

Slack variables ξ_i **can be added to allow misclassification of difficult or noisy examples.**



What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

13

Introduction

-- SVM classification

- Hard Margin vs. Soft Margin
  - **The old formulation:**

    Find $\mathbf{w}$ and b such that
    $\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
    $y_i\,(\mathbf{w}^T\mathbf{x_i} + \mathrm{b}) \geq 1$

  - **The new formulation incorporating slack variables:**

    Find $\mathbf{w}$ and b such that
    $\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
    $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

  - **Parameter C can be viewed as a way to control overfitting.**

14

Introduction

-- SVM classification

The following Lagrangian should be considered

$$L(w,b,\xi;\ \alpha,v)=\frac{1}{2}w^T w+C\sum_{K=1}^{N}\xi_k-\sum_{k=1}^{N}\alpha_k(y_k[w^T x_k+b]-1+\xi_k)+\sum_{k=1}^{N}v_k\xi_k$$

where $\alpha_k \geq 0,\ \ v_k \geq 0$ for $k=1,\ldots,N$

- The solution is given by the saddle point of the Lagrangian

$$\max_{\alpha,v}\min_{w,b,\xi}(L(w,b,\xi;\alpha,v))$$

$$\begin{cases} \frac{\partial L}{\partial w}=0 \rightarrow w=\sum_{K=1}^{N}\alpha_k y_k x_k \\ \frac{\partial L}{\partial b}=0 \rightarrow \sum_{k=1}^{N}\alpha_k y_k=0 \\ \frac{\partial L}{\partial \xi}=0 \rightarrow 0 \leq \alpha_k \leq C, k=1,\ldots,N \end{cases}$$

15

Introduction

-- SVM classification

- Linear SVMs: Overview

  - **The classifier is a *separating hyperplane.***
  - **Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i.**

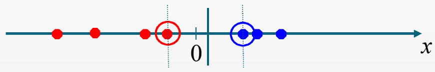
Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i^T x_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$f(x) = \Sigma\alpha_i y_i x_i^T x + b$

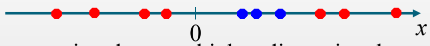
16

Introduction
-- SVM classification

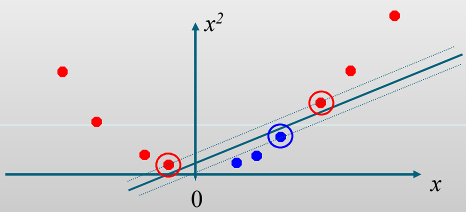
- Nonlinear classifier
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



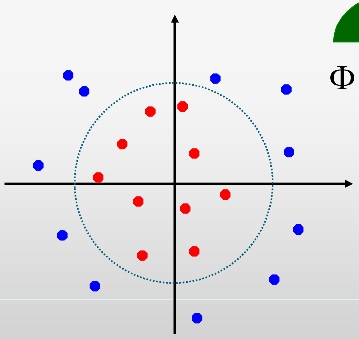
- How about… mapping data to a higher-dimensional space:



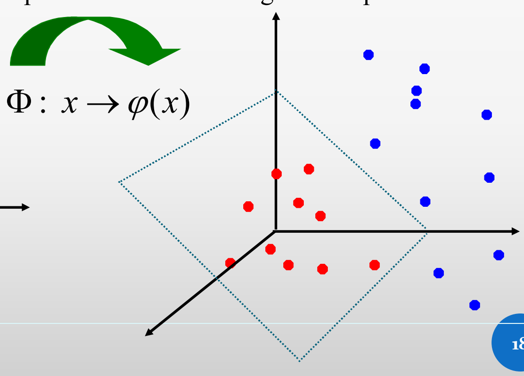
17

Introduction
-- SVM classification

- Nonlinear classifier
- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



$$\Phi: x \rightarrow \varphi(x)$$



18

Nonlinear SVM Classifier

- **Find $\alpha_1 \ldots \alpha_N$ such that**

$$\max_{\alpha_1 \ldots \alpha_N} Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to (1) $\sum_{i=1}^{N} \alpha_i y_i = 0$, (2) $0 \le \alpha_i \le C$ for $\forall \alpha_i$

- The function

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$

is called kernel function.

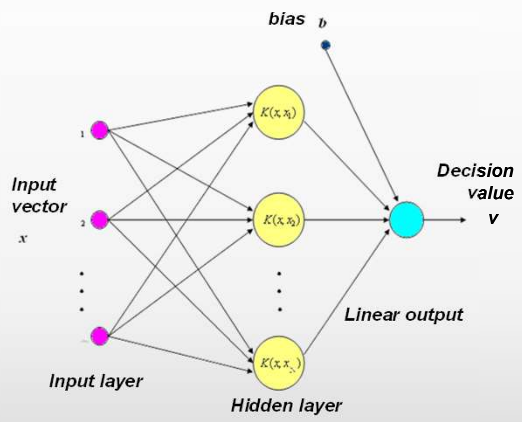
- ***Decision value*** is

$$v = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b$$

- ***Classifier*** is $y(x) = \text{sign}(v)$

19

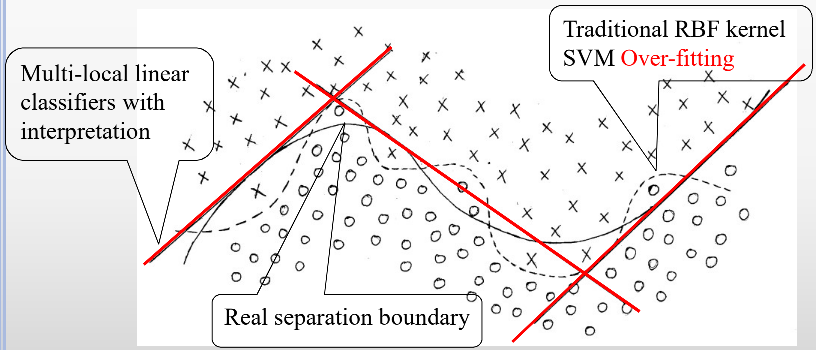
Nonlinear SVM Classifier (cont'd)



20

PIESEWISE LINEAR MODELING

- Multi-local linear models with interpolation can adjust and reduce the flexibility of separation boundary so as to prevent the over-fitting.



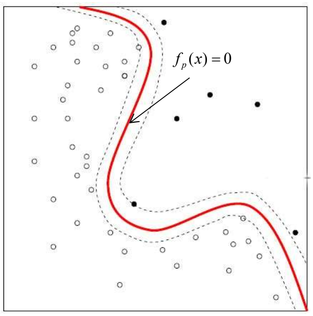
21

NONLINEAR CLASSIFICATION PROBLEM

Consider a two-class nonlinear classification problem whose classification boundary can be described by

$$f_p(x) = g(x)$$
$$x \in R^n$$

$g(\cdot)$ is a nonlinear function.



22

(CONT'D)

Applying Taylor expansion to the nonlinear function $g(\cdot)$ around the region $x = 0$, we get a regression form:

$$\begin{aligned} f_p(x) &= g(0) + g'(0)x + \frac{1}{2}x^T g''(0)x + \dots \\ &= g(0) + \left(g'(0) + \frac{1}{2}x^T g''(0) + \dots \right) x \end{aligned}$$

$$\theta(x) = \left(g'(0) + \frac{1}{2}x^T g''(0) + \dots \right)^T$$

$$f_p(x) = g(0) + x^T \theta(x)$$

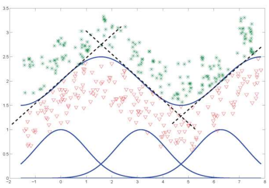
QUASI-LINEAR REGRESSION MODEL

Introduce an MIMO RBF network to parameterize $\theta(x)$

$$\theta(x) = \sum_{j=1}^{M} \Omega_j R_j(x) + \Omega_0$$

$$g(0) + x^T \Omega_0 \Rightarrow \sum_{j=1}^{M} b_j R_j(x) + b$$

$$f_p(x) = \sum_{j=1}^{M} (x^T \Omega_j + b_j) R_j(x) + b$$



Quasi-linear regression model is a nonlinear classifier consisting of multi-local linear classifiers with interpretation.

AN SVM APPROACH

- Introducing two parameter vectors, we have a regression form of the quasi-linear regression model

$$f_p(x) = \sum_{j=1}^{M} (x^T \Omega_j + b_j) R_j(x) + b$$

$$\Phi(x) = \left[R_1(x), x^T R_1(x), \ldots, R_M(x), x^T R_M(x) \right]^T$$

$$\Theta = [b_1, \Omega_1^T, \ldots, b_M, \Omega_M^T]^T$$

$$\Downarrow$$

$$f_P(x) = \Theta^T \Phi(x) + b$$

25

AN SVM APPROACH (CONT'D)

- Applying the Structural Risk Minimization principle similar to a standard SVM approach, we have

$$\min_{\Theta, b, \xi} J_P = \frac{1}{2} \Theta^T \Theta + c \sum_{k=1}^{N} \xi_k$$

$$s.t. \begin{cases} y_k [\Theta^T \Phi(x_k) + b] \geq 1 - \xi_k, k = 1, \ldots, N \\ \xi_k \geq 0, k = 1, \ldots, N \end{cases}$$

$$\Downarrow$$

The solution is given by the saddle point of the Lagrangian

$$L(\Theta, b, \xi; \alpha, v) = J_P(\Theta, \xi) - \sum_{k=1}^{N} (\alpha_k y_k [\Theta^T \Phi(x_k) + b] - 1 + \xi_k) - \sum_{k=1}^{N} v_k \xi_k$$

26

AN SVM APPROACH (CONT'D)

$$\max_{\alpha,v}\min_{\Theta,b,\xi} L(\Theta,b,\xi;\alpha,v)$$

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \Theta} = 0 \rightarrow \Theta = \sum_{k=1}^{N} \alpha_k y_k \Phi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 \rightarrow 0 < \alpha_k < c, k = 1, \dots, N \end{cases}$$

SVM with Quasi-linear Kernel

$$\max_\alpha J_D(\alpha) = -\frac{1}{2}\sum_{k,l=1}^{N} y_k y_l K(x_k,x_l)\alpha_k\alpha_l + \sum_{k=1}^{N}\alpha_k$$

$$s.t. \begin{cases} \sum_{k=1}^{N} \alpha_k y_k = 0 \\ 0 \le \alpha_k \le c, k = 1, \dots, N \end{cases}$$

By compositing the kernel using machine learning method, it is possible to construct an optimal one for specific applications

$$K(x_k,x_l) = \Phi^T(x_k)\Phi(x_l) = (1 + x_k^T x_l)\sum_{j=1}^{M} R_j(x_k)R_j(x_l)$$

27

AN SVM APPROACH (CONT'D)
-- QUASI-LINEAR KERNEL FUNCTION

$$K(x_k,x_l) = \Phi^T(x_k)\Phi(x_l) = (1 + x_k^T x_l)\sum_{j=1}^{M} R_j(x_k)R_j(x_l)$$

Linear kernel

$$K(x_k,x_l) = 1 + x_k^T x_l$$

M=1

M $\rightarrow \infty$

Nonlinear kernel, e.g. RBF kernel function

$$K(x_k,x_l) = k_n(x_k,x_l)$$

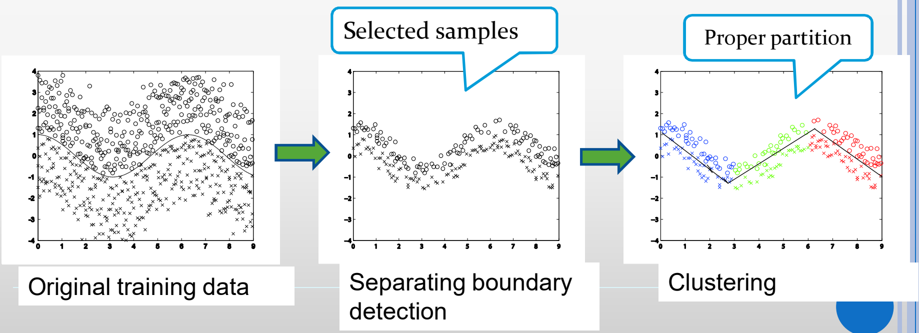
In cases of a nonlinear classification problem where an SVM with RBF kernel has poor performance than an SVM with linear kernel, the proposed SVM with quasi-linear kernel will be a good choice.

28

Implementation

- Kernel Composition
  - Dataset Partition (along the separation boundary)
    - Step 1: detect separation boundary;
    - Step 2: cluster the selected training samples.

29

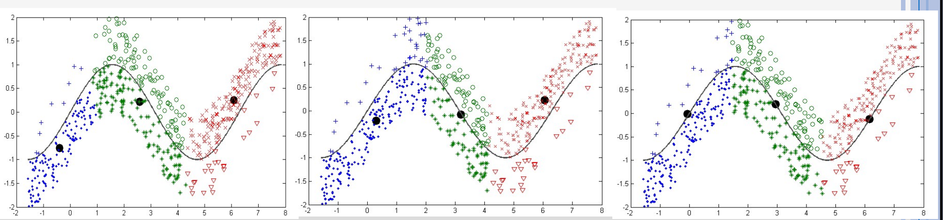


Partition method ---Modified Kmeans

- A Modified *k-means* with label and distance guidance

$$\min\{\sum_{j=1}^{k}\sum_{i=1}^{n}||X_i - C_j|^2 + \underbrace{\lambda_1 \sum_{j=1}^{k}|\sum_{i=1}^{n} Z_{i,j}Y_i|}_{\text{label guidance}} + \underbrace{\lambda_2 \sum_{j=1}^{k} D_j}_{\text{distance guidance}}\}$$

30



IMPLEMENTATION (CONT'D)

- Kernel Composition
  - Kernel function Construction
    - **Quasi-linear kernel function**

$$K(x_k, x_l) = (1 + x_k^T x_l)\sum_{j=1}^{M} R_j(x_k) R_j(x_l)$$

- **R_i is the radial basis function expressed as Gaussian**

$$R(x) = e^{-\frac{(x-\mu)^2}{\lambda\sigma^2}}$$

Subset center

Scale parameter

Subset radius

31