SVM application with R

May 13, 2015

Introduction

- Support Vector Machine is one of the most popular supervised machine learning algorithms.
  - Inner product-training of the model is very efficient
  - Kernel function-it allows us to conduct non-linear fitting
- Agenda
  - Example 1: face recognition
  - Example 2: MRI classification
  - Understanding SVM
    - Cost function & boundary
    - Kernel functions & parameters
  - Example 3: spam email classification (hands-on)

Example 1: Face recognition

Example 1: Yale face recognition (1/4)

- This example aims to compare SVM with KNN* and show the visualization of SVM model
- Data: Yale FR database, 494 frontal face picture of 38 people (13 picture per person), with resolution of 192x168 (32k pixels)

- Learning target: recognize the identity of the photo
- Training and test: 50 photos are randomly selected as test set
- Algorithm comparison: SVD**+KNN vs SVM
- SVM strategy: 703*** "one vs one" classifiers are built

* K-nearest-neighbor * *SVD Singular Value Decomposition *** Choose 2 from 38

Example 1: Yale face recognition (2/4)

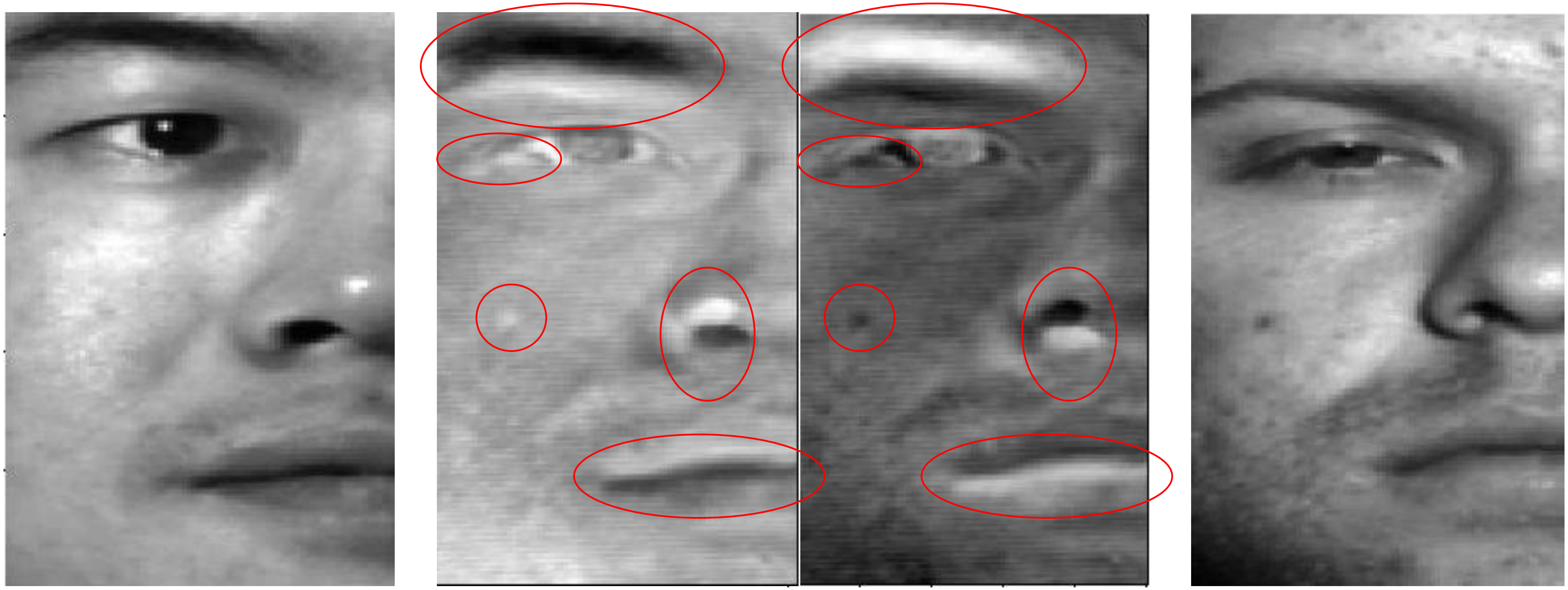
- Solution 1:
  - SVD + 1-nearest-neighbor
  - Result: reduce dimension to less than 444 and achieve the accuracy of 47/50.
- Solution 2:
  - SVM + Support Vector Face
  - Result: achieve accuracy of 50/50, SV-Face is used to visualize the model.

Example 1: Yale face recognition (3/4)

- Boundary function:

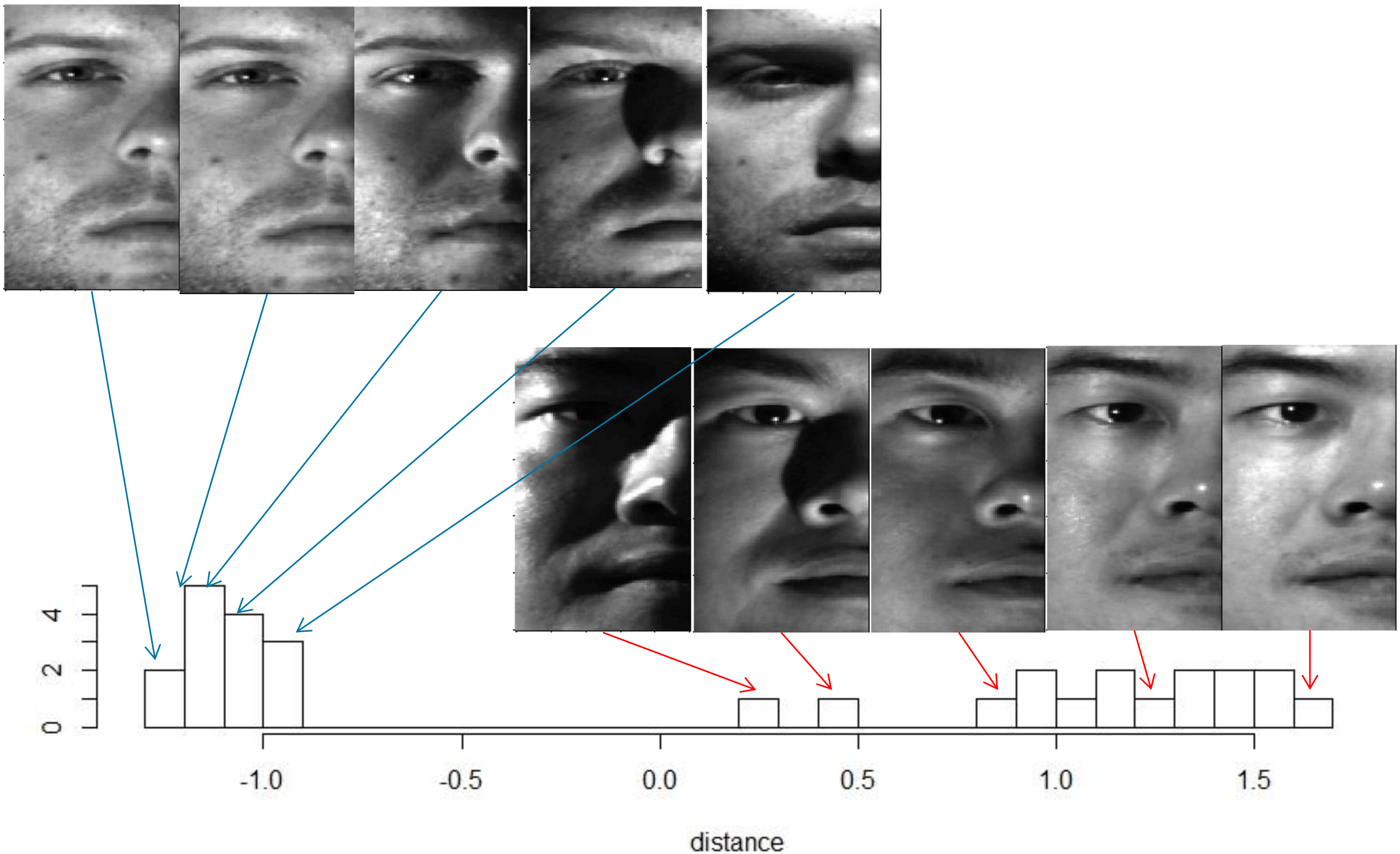
$$f(x) = w^T x + b$$

- Support Vector Face—visualization of $f(x)$



Example 1: Yale face recognition (4/4)

- Visualization of the distance



Example 2: MRI classification

Example 2: MRI classification (1/2)

- This Example shows the framework to introduce prior knowledge to SVM through regularizing the coefficients and designing kernel
- Literature :Spatial and anatomical regularization of SVM: a general framework for neuroimaging data
- Model:

$$Alzheimer \sim SVM(\,256 * 256 * 180 voxels)$$

- To much degree of freedom
  - Hundreds of training samples but millions of variables
- Introduce constrain
  - Penalize coefficients of “similar “ voxels using adjacency/similarity matrix (11m by 11m)
  - Similarity: spatial and anatomical prior knowledge
- Ease of computation
  - Kernalization

Example 2: MRI classification (2/2)

Without regularization

With regularization

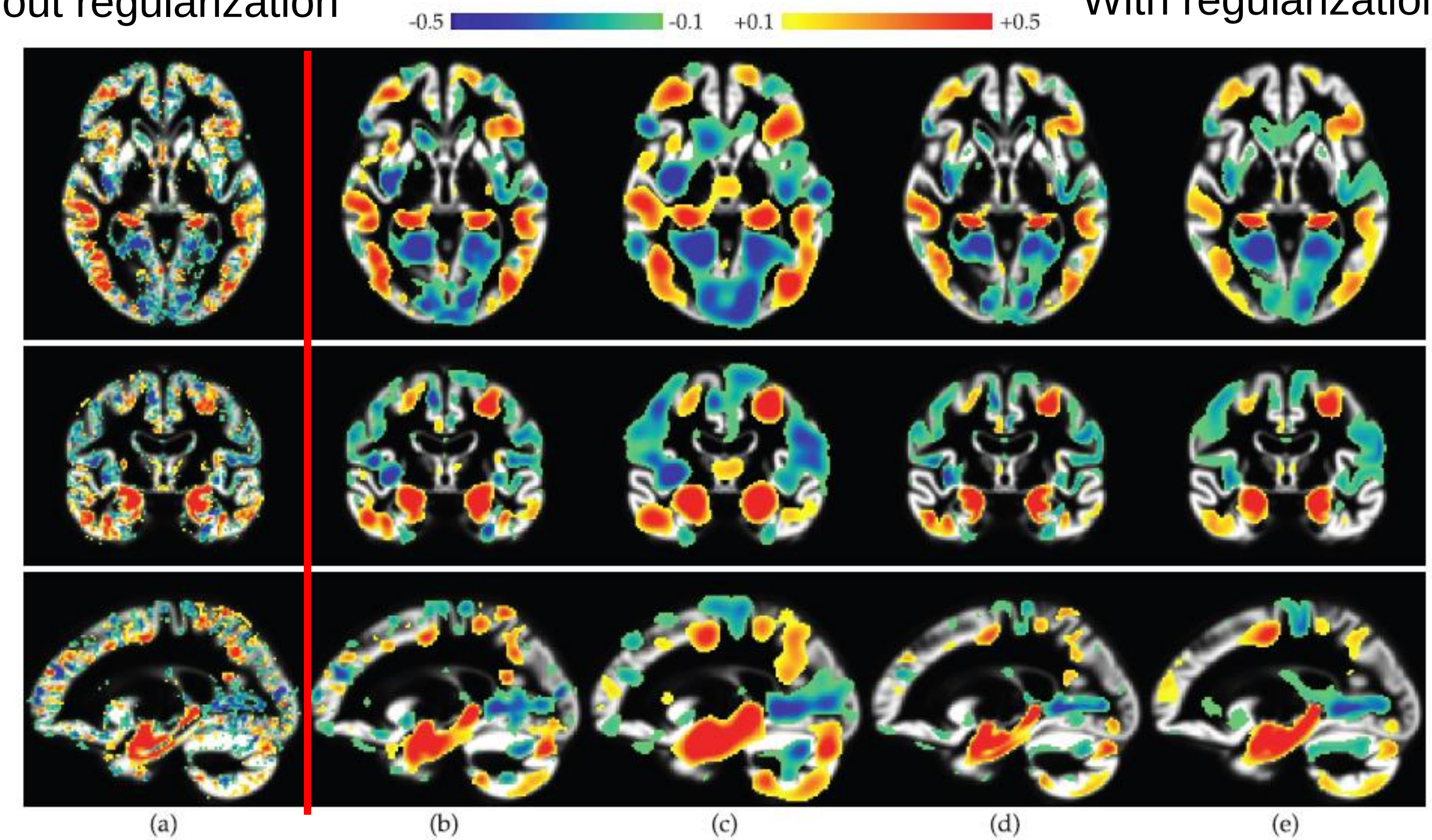
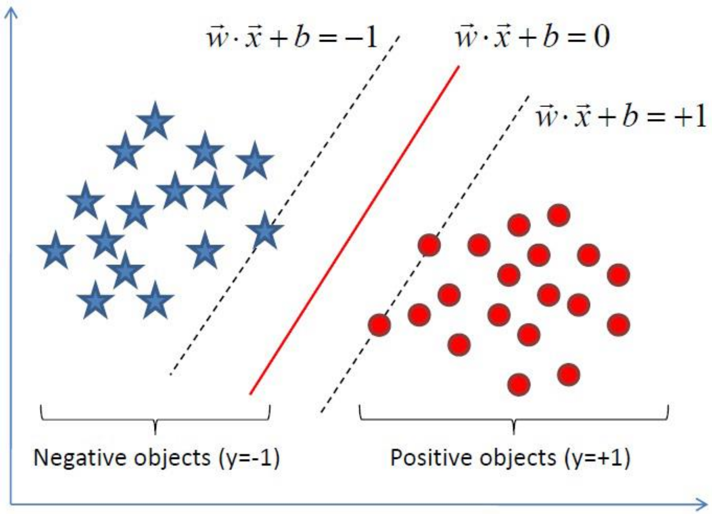


Fig. 4. Normalized $\mathbf{w}^{\mathrm{opt}}$ coefficients for: (a) *Voxel-Direct*, (b) *Voxel-Regul-Spatial* (FWHM = 4 mm), (c) *Voxel-Regul-Spatial* (FWHM = 8 mm), (d) *Voxel-Regul-CombineFisher* (FWHM$\sim$4 mm, $\sigma_{\mathrm{loc}} = 10$), and (e) *Voxel-Regul-CombineFisher* (FWHM$\sim$8 mm, $\sigma_{\mathrm{loc}} = 10$).

Understanding SVM

Understanding SVM (1/5)

- SVM is trying to find
  - Boundary points (support vectors)
  - Maximum in-between distance



- Boundary function

$$y = \sum_s \alpha_s y_s x_s x + b; \quad s\ for\ all\ support\ vectors$$

Boundary points defines separation line

- Cost function

$$cost = \frac{1}{2}||w||^2 - \frac{1}{n}\sum_i l_{hinge}(\alpha_i(y_i(wx_i + b) - 1)) + C\sum_i \xi_i - \sum_i r_i \xi_i\ for\ i = 1,2,\ldots,n$$

Can be changed to kernel function $K(w, x_i)$ for non-linear fitting

Tolerance of outliers, avoiding over fitting

Understanding SVM (2/5)

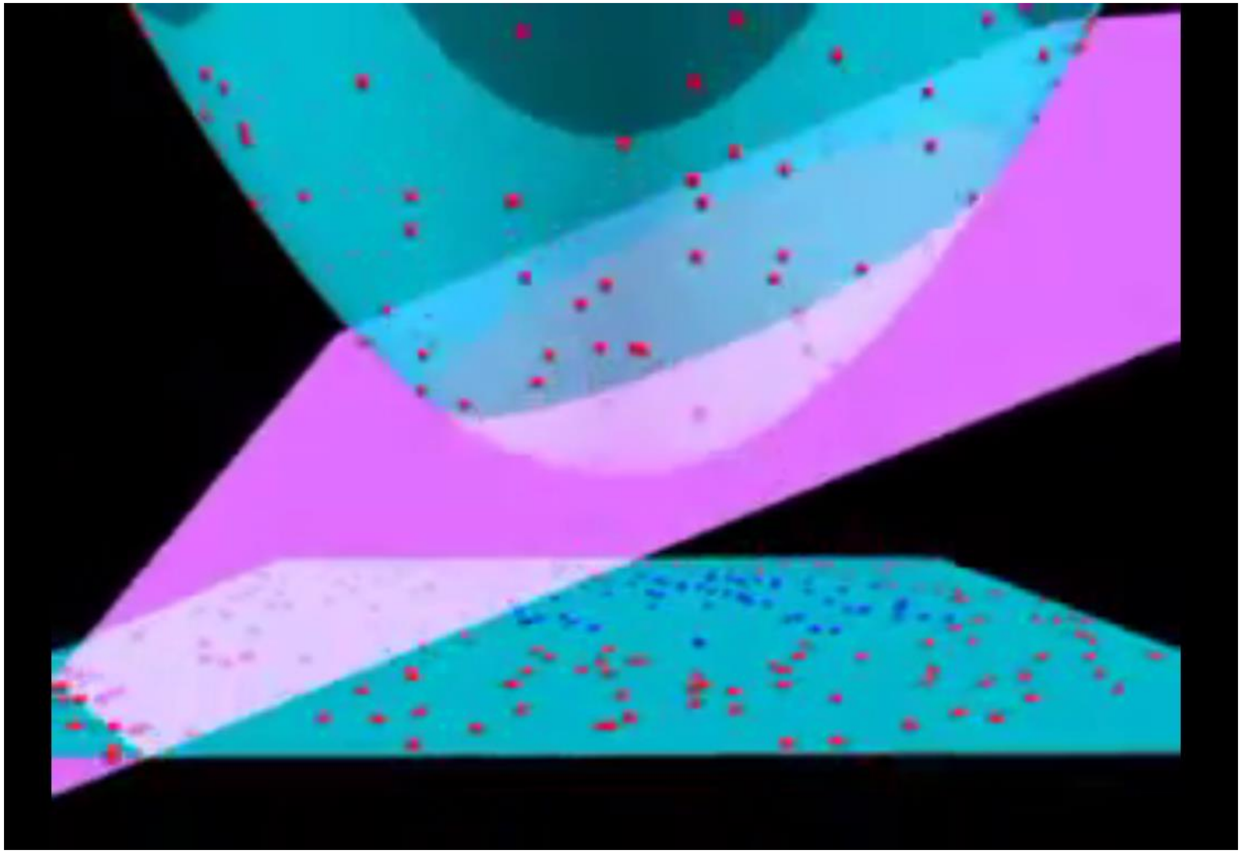
Input & Output

- Input:
  - Data: numerical, ordinal, categorical
  - Parameters:
    - Slack variable C
    - Kernel functions and parameters
- Output:
  - SV index: which samples are the boundary points
  - Coefficients

Understanding SVM (3/5)

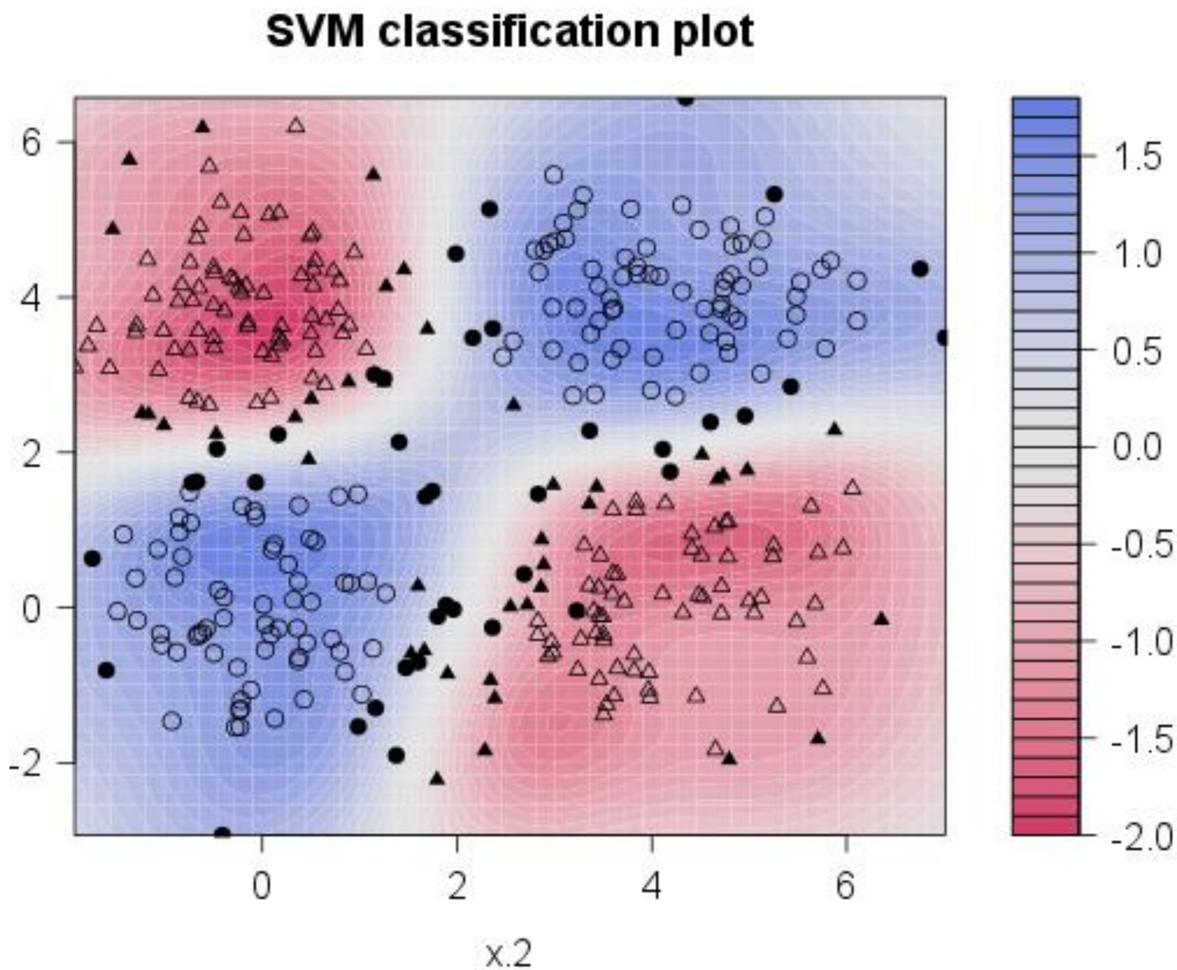
Kernel function

- Why we need kernel?
  - High dimension projection -> Non-linear boundary



Understanding SVM (4/5)

Gaussian Kernel—the 2D "gravitational field"



Understanding SVM (5/5)

Cost function

$$cost = \frac{1}{2}||w||^2 - \sum_i \alpha_i (y_i (K(w, x_i) + b) - 1) + C \sum_i \xi_i - \sum_i r_i \xi_i$$

$$for\ i = 1,2,\dots,n$$

- Polynomial Kernel

$$K(w,x) = (w^T x + b)^d$$

- Gaussian Kernel

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

/ larger/smaller value increases the boundary curvature

- Other Kernels

Example 3: spam email classification

Example 3: spam email

- Spam email classification:
  - Use extracted email features to predict whether an email is a spam email
- R code hands-on
  - Loading library
  - Accessing data
  - Training SVM and interpreting results
  - Tuning parameters
- Learn through practice—tips for R programming
  - Describe your problem with key words and Google them
  - Use R help
  - Find sample code and visualize each step

Get your hands dirty!

Example 3: spam email

- Error on the test set

Kernel/ parameter	False positive: Nonspam-> spam	False negative: Spam->Nonspam	False rate
Linear	6.7%	11.1%	8.4%
Gaussian(RDB)			
(sigma) 0.01	4.5%	11.5%	7.3%
0.03	4.5%	12.4%	7.7%
0.05	4.1%	14.3%	8.1%
0.07	4.3%	16.0%	9.0%
0.09	4.1%	18.1%	9.7%
Polynomial			
(degree) 2	10.3%	13.2%	11.4%
3	9.4%	15.3%	11.7%
4	11.7%	14.7%	12.9%

Adding curvature

Adding curvature

Thank you!