SVM application with R

May 13, 2015

Introduction

- Support Vector Machine is one of the most popular supervised machine learning algorithms.
  - The training of the model is very efficient
  - And the kernel function allows us to fit a non-linear curve as the classification boundary
- Agenda
  - Example 1: face recognition
  - Understanding SVM
  - Example 2: spam email classification (hands-on)

Example 1

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.

- 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

* Choose 2 from 38

Example 1: Yale face recognition (2/4)

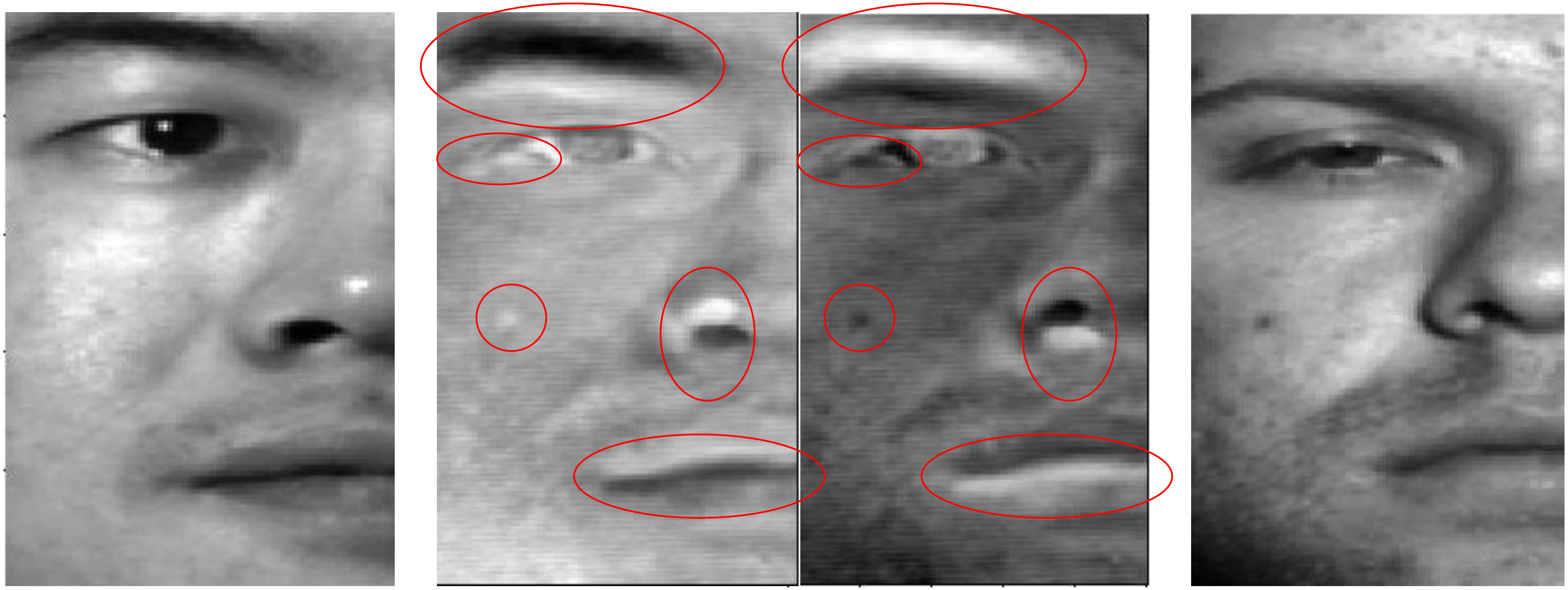
- Solution 1:
  - SVD + 1-nearest-neighbor
  - Result: reduce dimension to less than 444 and achieve the highest accuracy of 47/50.
- Solution 2:
  - SVM + Support Vector Face
  - Result: achieve accuracy of 50/50, SV-Face is used to visualize the model
- Comments:
  - Solution 1 is a quick solution and it fits for huge training set scenarios, yet it does not generate any statistical information of the training set.
  - Solution 2 is slower than solution 1 but it is generalizable, and compared to other classification algorithm, it is more stable because it focus on the boundary of different classes.

Example 1: Yale face recognition (3/4)

- Linear classification function:

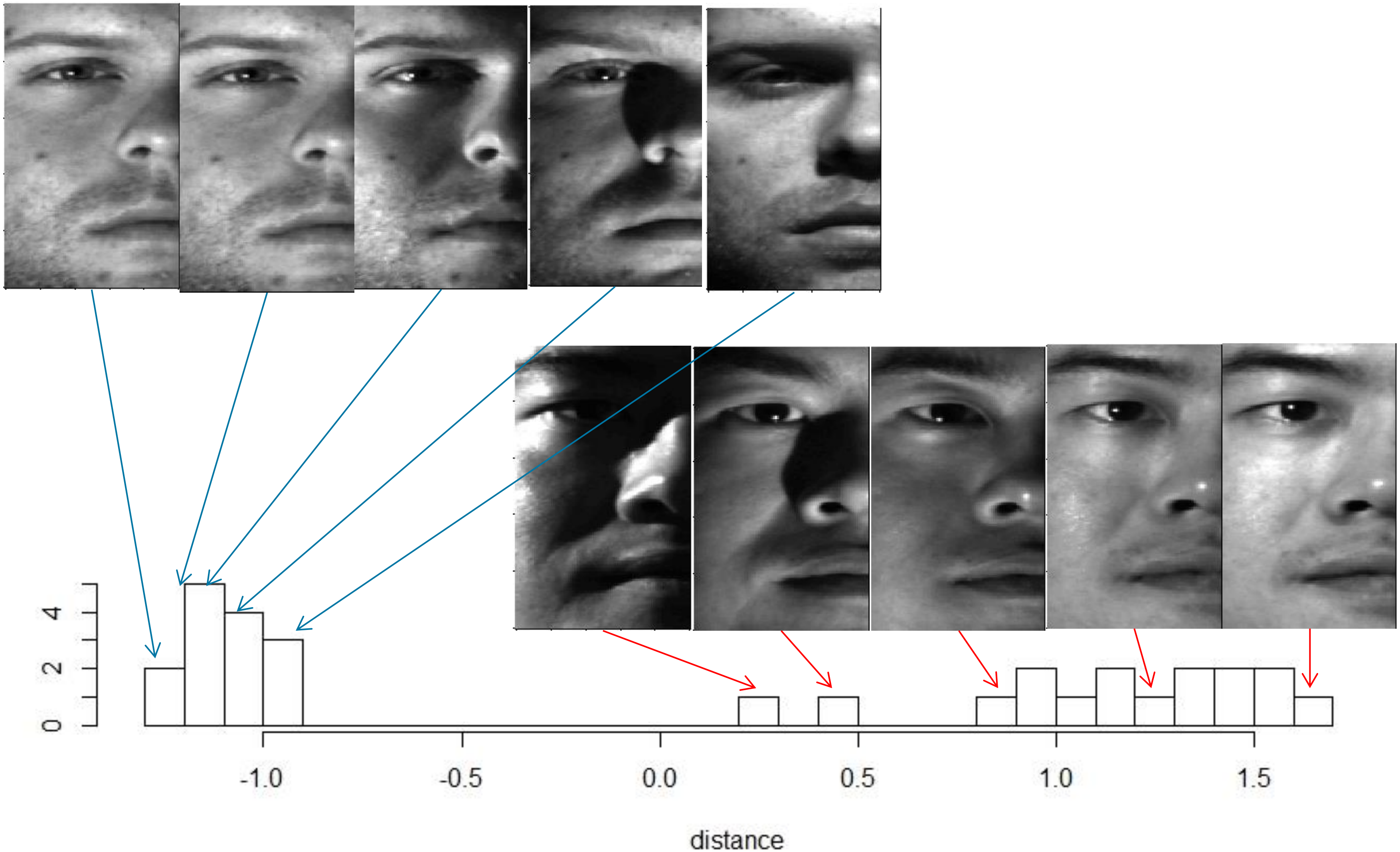
$$f(x) = \left(\sum_i \alpha_i x_i y_i\right)^T x + b$$

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



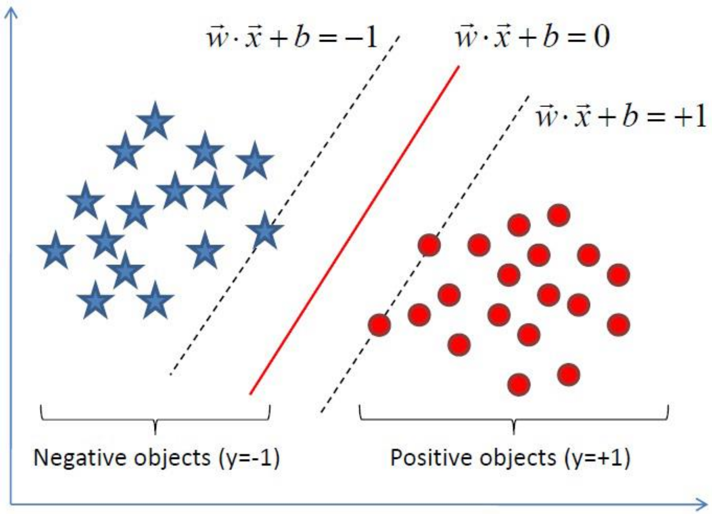
Example 1: Yale face recognition (4/4)

- Visualization of the distance



Understanding SVM (1/5)

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



- Boundary function

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

Boundary points defines separation line

- Cost function

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

Change to kernel function
$K(w, x_i)$ for non-linear fitting

Tolerance of outliers,
avoiding over fitting

Understanding SVM (2/5)

Input & Output

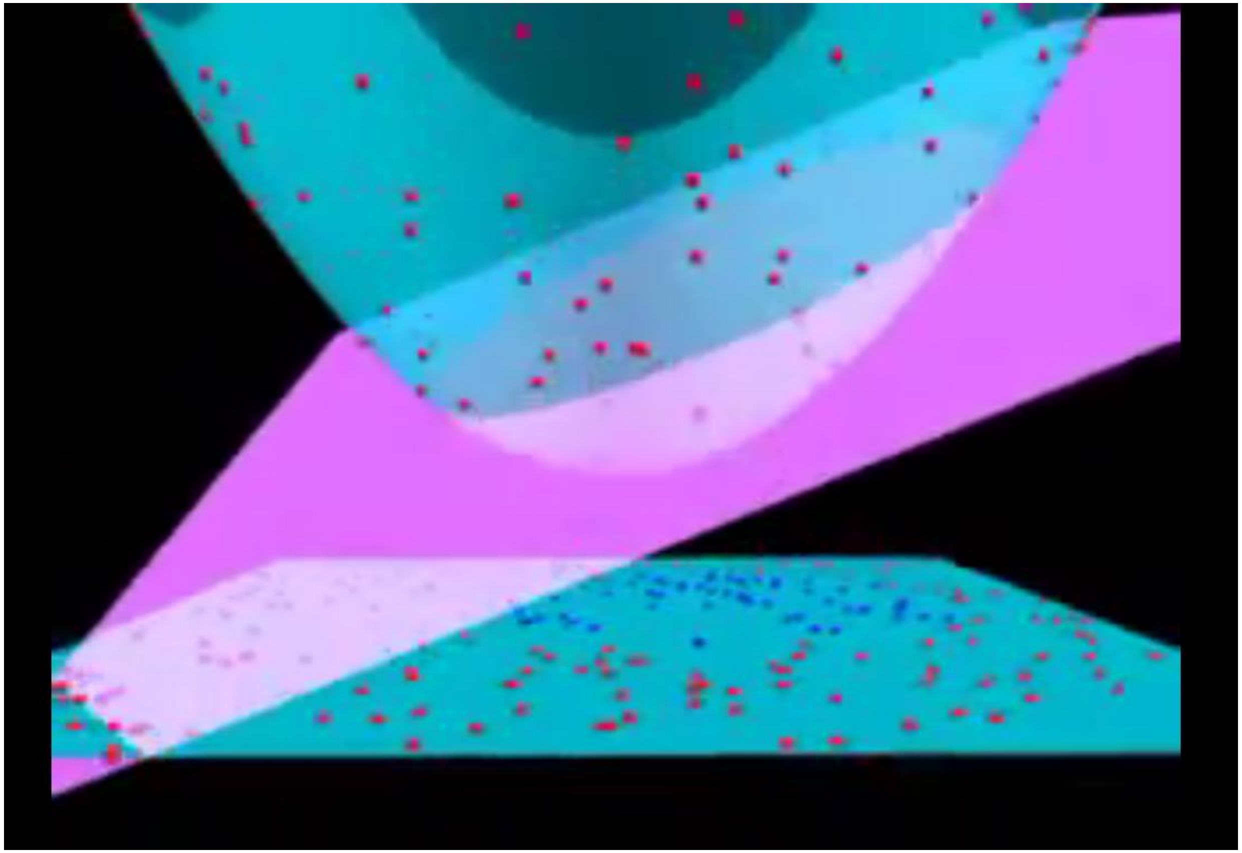
- Input:
  - Data: numerical, ordinal, categorical
  - Parameters:
    - Slack variable C
    - Kernel functions and parameters
- Output:
  - SV index: which are the boundary points
  - Coefficients

Understanding SVM (3/5)

Kernel function

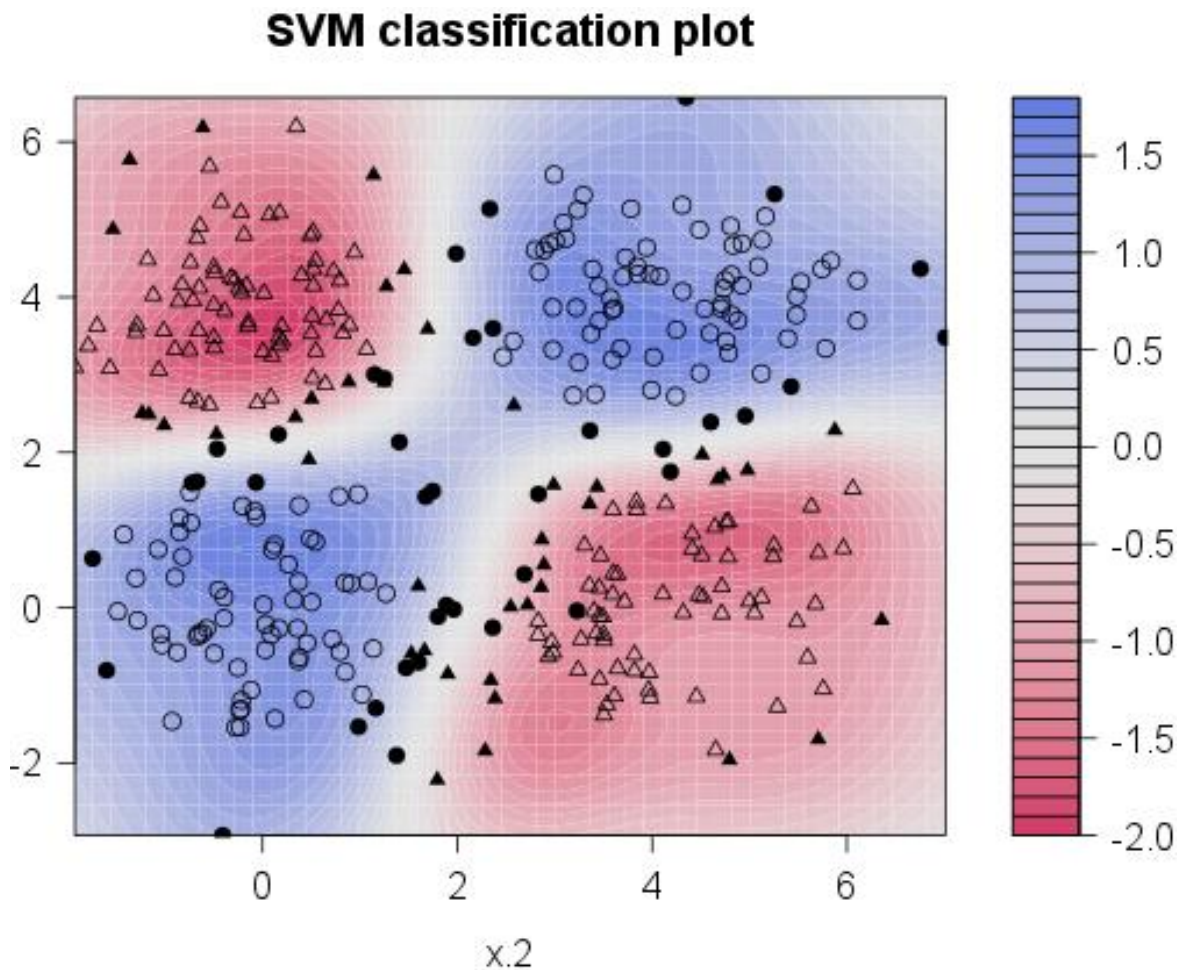
Why we need kernel?

-Non-linear boundary and high dimension projection



Understanding SVM (4/5)

Gaussian Kernel in 2D space



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 2: spam email

- R code hands-on
  - Loading library, data
  - Segregating the dataset to training and testing set
  - Training SVM and result interpretation
  - Tuning parameters

- Tips for R programming
  - Describe your problem with key words and Google them
  - Use R help
  - Visualize each steps

Example 2: spam email

- Error on the test set

Kernel/ parameter		Nonspam-> spam False positive	Spam->Nonspam False negative	False rate
Linear		6.7%	11.1%	8.4%
Gausian(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!