L6: SVM II

Shan Wang

Lingnan College, Sun Yat-sen University

2020 Data Mining and Machine Learning LN3119

https://wangshan731.github.io/DM-ML/

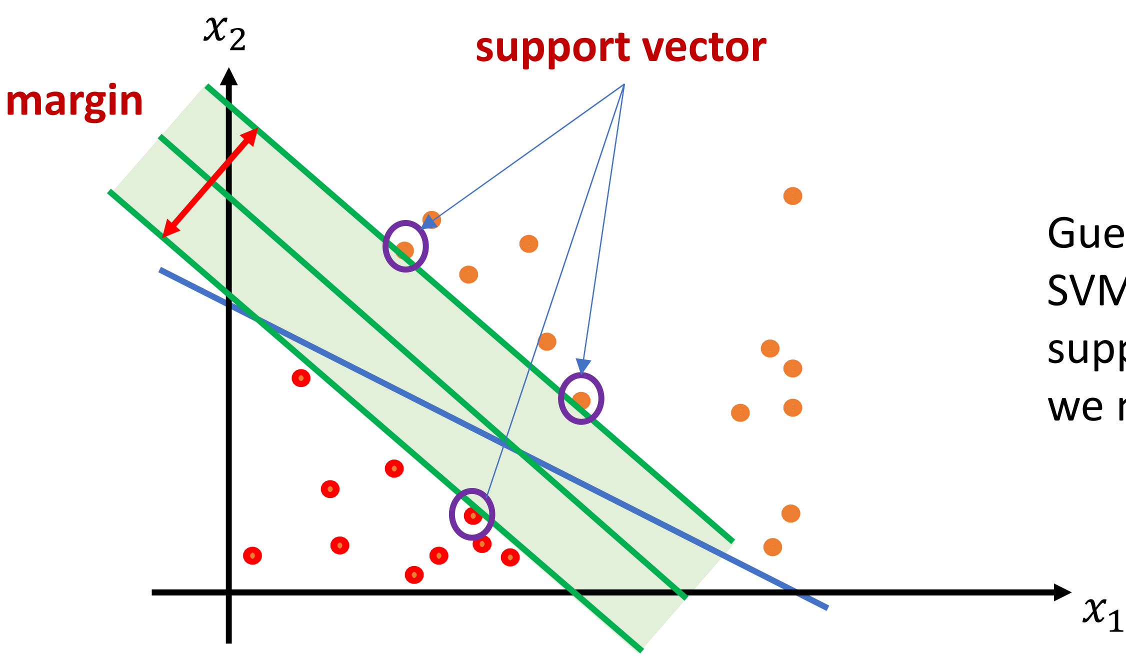
Last and this lecture

- ✓Linear SVM
  - ✓Model
  - ✓Strategy
  - ✓Algorithm
- ✓Regularization
- Kernels
- Application: diabetes care revisit

Reference: CS420, Weinan Zhang (SJTU)

Linear SVM

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \begin{cases} +1, & \text{if } \boldsymbol{\theta}'\boldsymbol{x} + \theta_0 \geq 0 \\ -1, & \text{if } \boldsymbol{\theta}'\boldsymbol{x} + \theta_0 < 0 \end{cases}$$



Guess: in linear SVM, how many support vectors we need?

Prim

$$\min_{\boldsymbol{\theta},\theta_0} \frac{1}{2}\|\boldsymbol{\theta}\|^2$$

$$s.t.\ \ y_i(\boldsymbol{\theta}'\boldsymbol{x_i} + \theta_0) \geq 1, i = 1, \dots, N$$

Dual

$$\max_{\boldsymbol{\alpha \geq 0}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \boldsymbol{x_j}'\boldsymbol{x_i}$$

$$s.t. \sum_{i=1}^{N} \alpha_i y_i = 0$$

Regularization: soft margin

$$\min_{\boldsymbol{\theta},\theta_0} \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$s.t.\ y_i(\boldsymbol{\theta}'\boldsymbol{x_i} + \theta_0) \geq 1 - \xi_i, i = 1, \dots, N$$

$$\xi_i \geq 0, \qquad i = 1, \dots, N$$

Quadratic programming

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \boldsymbol{x_j}'\boldsymbol{x_i}$$

$$s.t. \sum_{i=1}^{N} \alpha_i y_i = 0$$

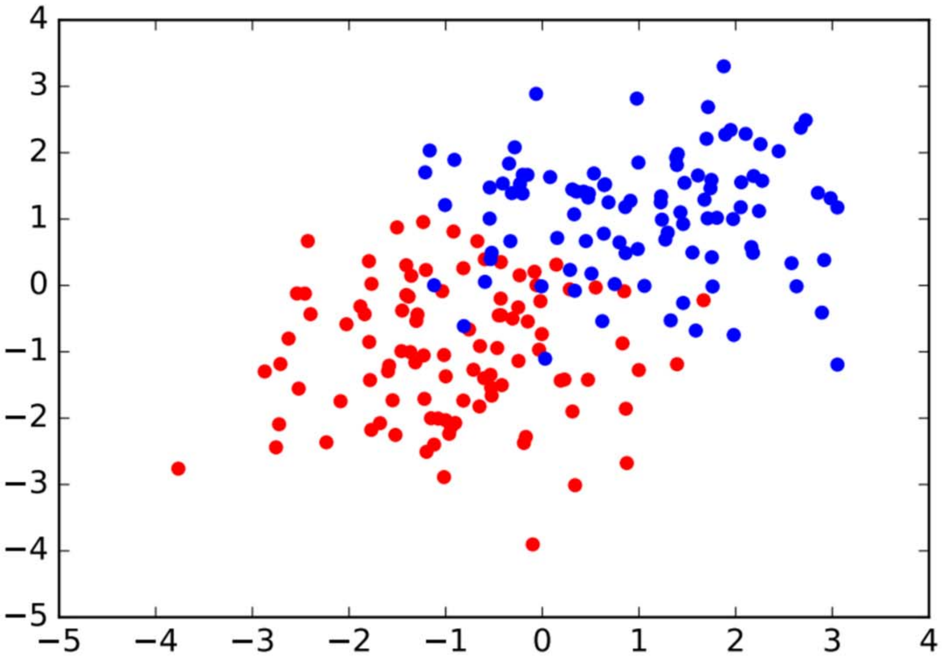
$$0 \leq \alpha_i \leq C, \qquad i = 1, \dots, N$$

SMO algorithm

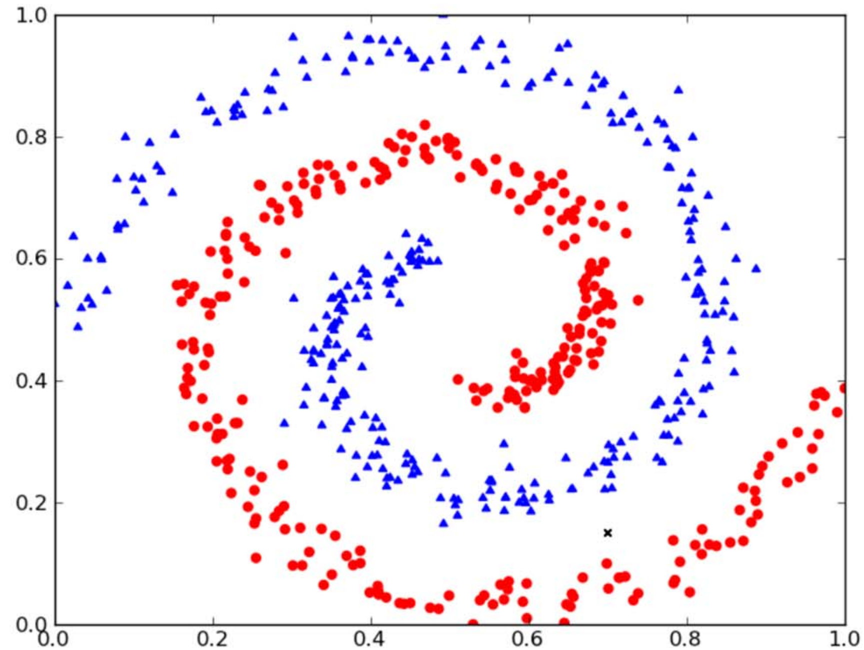
Kernels

Non-separable Data

- When soft margin does not work



May be solved by soft margin

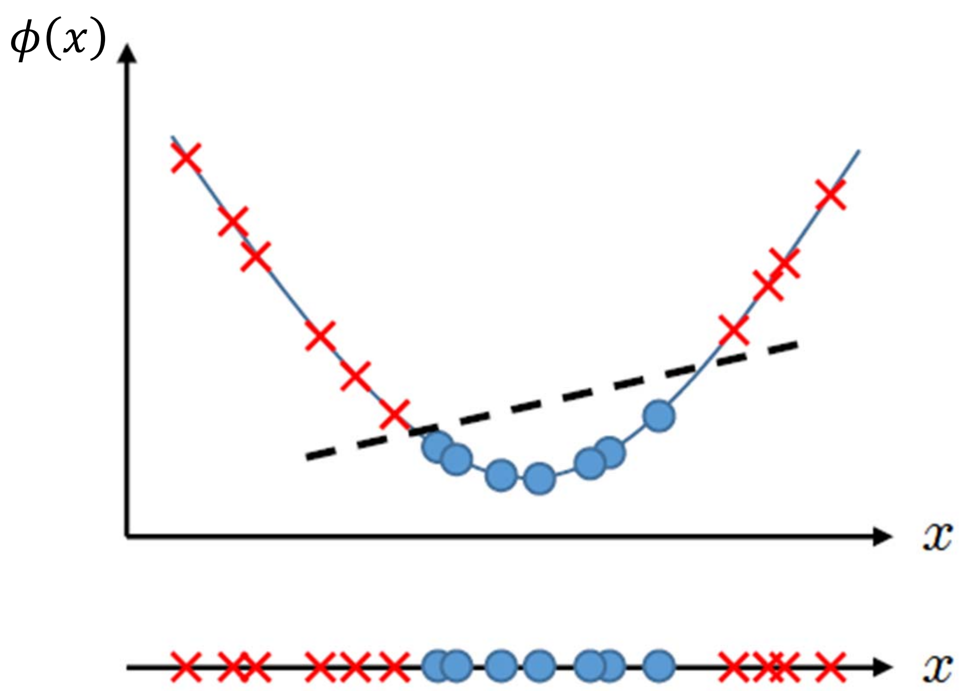


Cannot be solved by soft margin

Non-separable Data (cont.)

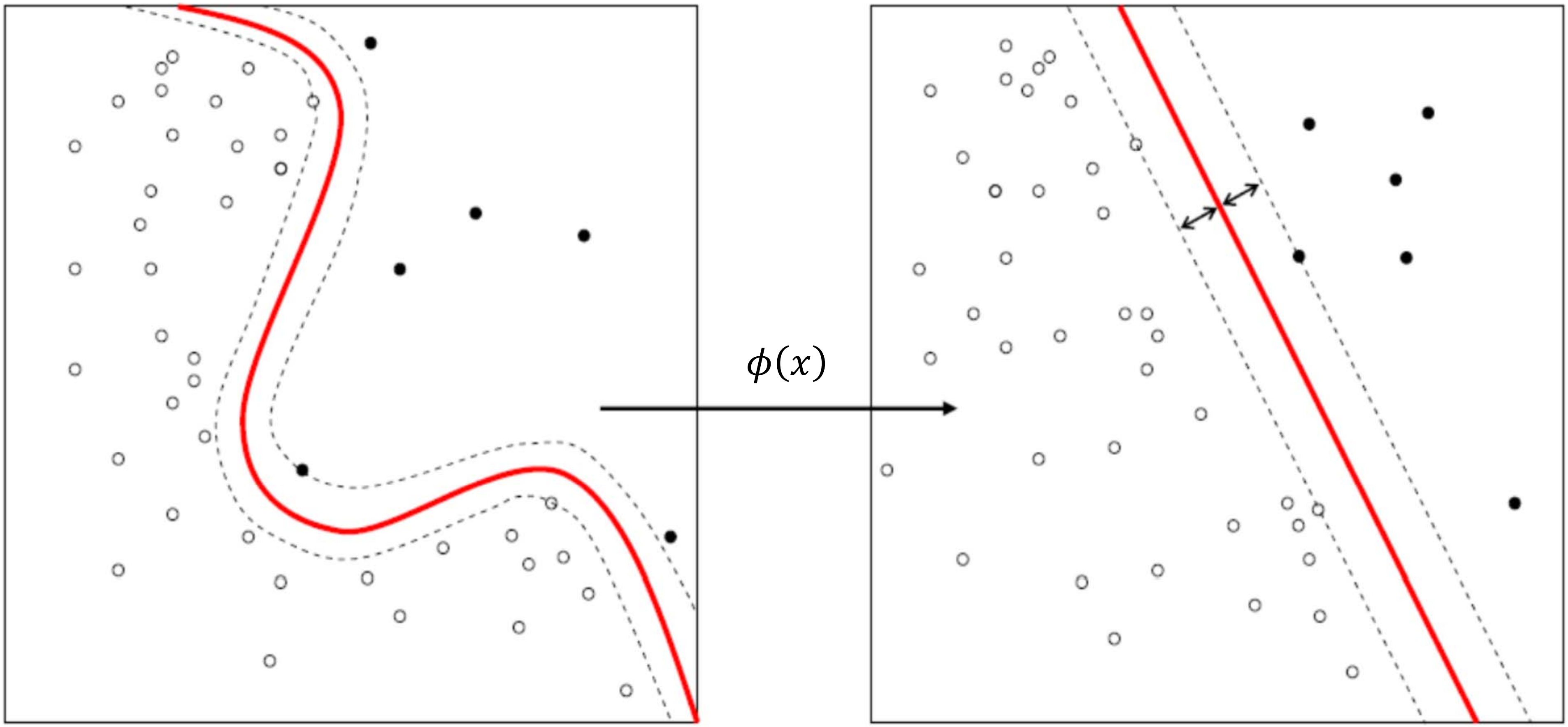
- Mapping feature vectors to a different space
- For example

$$\phi(x) = x^2$$



Non-separable Data (cont.)

- More generally,



From linear SVM to kernel SVM

- Linear SVM:

$$\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boxed{\boldsymbol{x_i}'\boldsymbol{x_j}}$$

  - the inner product of $\boldsymbol{x_i}, \boldsymbol{x_j}$: $\boldsymbol{x_i}'\boldsymbol{x_j}$

- After feature mapping

$$\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boxed{\phi(\boldsymbol{x_i})'\phi(\boldsymbol{x_j})}$$

  - the inner product of $\phi(\boldsymbol{x_i}), \phi(\boldsymbol{x_j})$: $\phi(\boldsymbol{x_i})'\phi(\boldsymbol{x_j})$

- Kernel function: $K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \phi(\boldsymbol{x_i})'\phi(\boldsymbol{x_j})$

Kernel trick

$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)'\phi(\boldsymbol{x}_j)$

- With the example feature mapping function

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

- The corresponding kernel is

$$K(x_i, x_j) = \phi(x_i)'\phi(x_j) = x_i x_j + {x_i}^2 {x_j}^2 + {x_i}^3 {x_j}^3$$

[Trick]

- For many cases, without defining $\phi(\cdot)$, we can directly define $K(x_i, x_j)$ for each pair of $\boldsymbol{x_i}, \boldsymbol{x_j}$
- For prediction, only need $K(\boldsymbol{x_i}, \boldsymbol{x})$ on support vector $\boldsymbol{x_i}$

Kernel matrix

- For a finite set of instances $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N}\}$
- The kernel matrix K is defined as $\left[K_{ij}\right]_{i,j=1,2,\dots,N}$ where $K_{ij} = K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right)$
- If $K(\cdot,\cdot)$ is a valid kernel (that is, is defined by some feature mapping ϕ), then the corresponding kernel matrix $K = \left[K_{ij}\right]_{ij} \in \mathbb{R}^{N*N}$ is symmetric positive semi-definite matrix
  - Symmetric: $K_{ij} = K_{ji}$
  - Positive semi-definite: for $\forall\ z \in \mathbb{R}^N$, we have $z'Kz \geq 0$

How to prove it?

Example valid kernels

- Gaussian kernel
  - $K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2})$
  - Radial basis function (RBF) kernel
- Simple polynomial kernel
  - $K(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}'\boldsymbol{z})^d$
- Cosine similarity kernel
  - $K(\boldsymbol{x}, \boldsymbol{z}) = \frac{\boldsymbol{x}'\boldsymbol{z}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}$
- Sigmoid kernel
  - $K(\boldsymbol{x}, \boldsymbol{z}) = \tanh(\alpha \boldsymbol{x}'\boldsymbol{z} + c)$
  - $\tanh(b) = \frac{1-\exp(-2b)}{1+\exp(-2b)}$

Pros and cons of SVM

- Advantages:
  - The solution is globally optimal
    - Based on convex optimization
  - Can be applied to both linear/non-linear classification problems
  - Can be applied to high-dimensional data
    - The complexity of the data set mainly depends on the support vectors
  - Complete theoretical guarantee
    - Compared with deep learning
- Disadvantages:
  - The number of parameters α is number of samples, thus hard to apply to large-scale problems
    - SMO can ease the problem a bit
  - Mainly applies to binary classification problems
    - For multi-classification problems, can solve several binary classification problems, but might face the problem of imbalanced data

Application

Application: Diabetes Care Revisit

Claims Data

- Predict poor quality care or not

Medical Claims

Diagnosis, Procedures,
Doctor/Hospital, Cost

Pharmacy Claims

Drug, Quantity, Doctor,
Medication Cost

- Electronically available
- Standardized

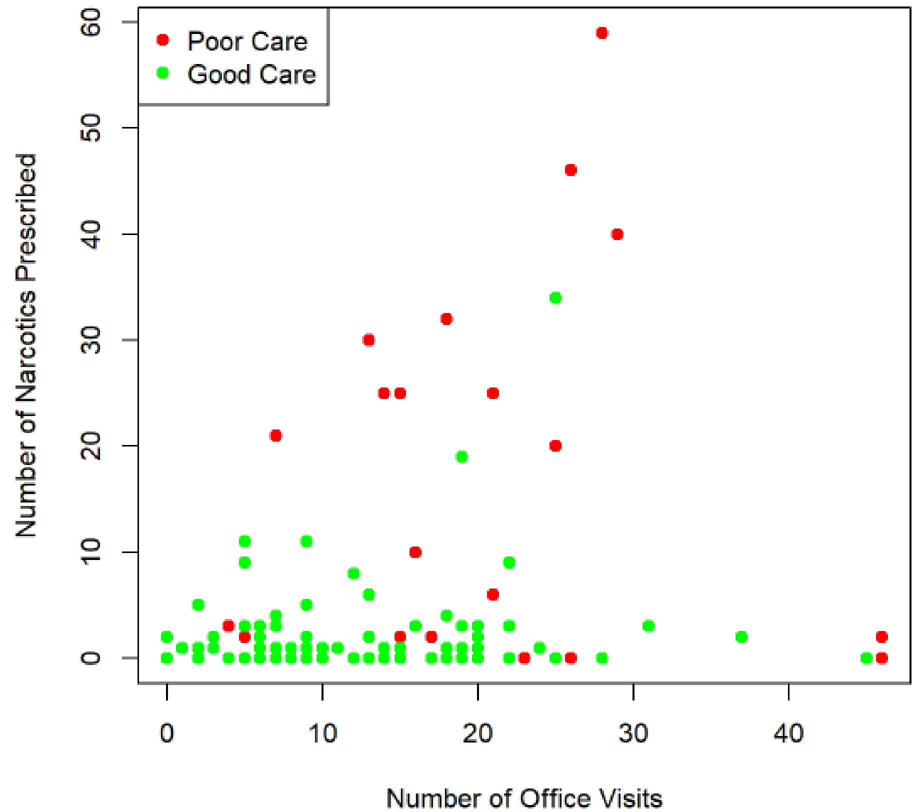
- Not 100% accurate
- Under-reporting is common
- Claims for hospital visits can be vague

Building a model

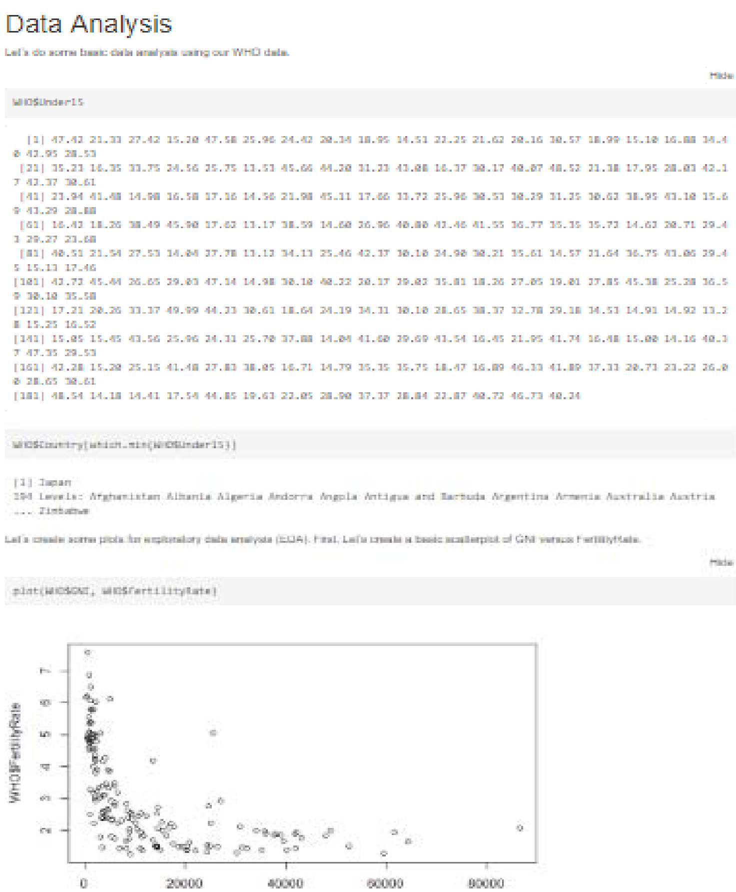
- We use a **SVM**
  - Predicts an outcome variable, or *dependent/response variable*
  - Using a set of *independent/explanatory variables*
- Dependent variable: Poor care or not
  - is equal to 1 if the patient had poor care, and equal to -1 if the patient had good care
- Independent variables:
  - Number of Office Visits (OfficeVisits)
  - Number of Narcotics Prescribed (Narcotics)
  - Etc.

Model for Healthcare Quality

- Plot of the independent variables
  - Number of Office Visits (OfficeVisits)
  - Number of Narcotics Prescribed (Narcotics)
- Red are poor care
- Green are good care
- Are these variables predictive of good care or poor care?



Let’s get our hands dirty!



Data Analysis

Let's do some basic data analysis using our WHO data.

Hide

```
WHO$Under15
```

```
  [1] 47.42 21.33 27.42 15.20 47.58 25.96 24.42 20.34 18.95 14.51 22.25 21.62 20.16 30.57 18.99 15.10 16.88 34.4
0 42.95 28.53
 [21] 35.23 16.35 33.75 24.56 25.75 13.53 45.66 44.20 31.23 43.08 16.37 30.17 40.07 48.52 21.38 17.95 28.03 42.1
7 42.37 30.61
 [41] 23.94 41.48 14.98 16.58 17.16 14.56 21.98 45.11 17.66 33.72 25.96 30.53 30.29 31.25 30.62 38.95 43.10 15.6
9 43.29 28.88
 [61] 16.42 18.26 38.49 45.90 17.62 13.17 38.59 14.60 26.96 40.80 42.46 41.55 36.77 35.35 35.73 14.62 20.71 29.4
3 29.27 23.68
 [81] 40.51 21.54 27.53 14.04 27.78 13.12 34.13 25.46 42.37 30.10 24.90 30.21 35.61 14.57 21.64 36.75 43.06 29.4
5 15.13 17.46
[101] 42.73 45.44 26.65 29.03 47.14 14.98 30.10 40.22 20.17 29.02 35.81 18.26 27.05 19.01 27.85 45.38 25.28 36.5
9 30.10 35.58
[121] 17.21 20.26 33.37 49.99 44.23 30.61 18.64 24.19 34.31 30.10 28.65 38.37 32.78 29.18 34.53 14.91 14.92 13.2
8 15.25 16.52
[141] 15.05 15.45 43.56 25.96 24.31 25.70 37.88 14.04 41.60 29.69 43.54 16.45 21.95 41.74 16.48 15.00 14.16 40.3
7 47.35 29.53
[161] 42.28 15.20 25.15 41.48 27.83 38.05 16.71 14.79 35.35 35.75 18.47 16.89 46.33 41.89 37.33 20.73 23.22 26.0
0 28.65 30.61
[181] 48.54 14.18 14.41 17.54 44.85 19.63 22.05 28.90 37.37 28.84 22.87 40.72 46.73 40.24
```

```
WHO$Country[which.min(WHO$Under15)]
```

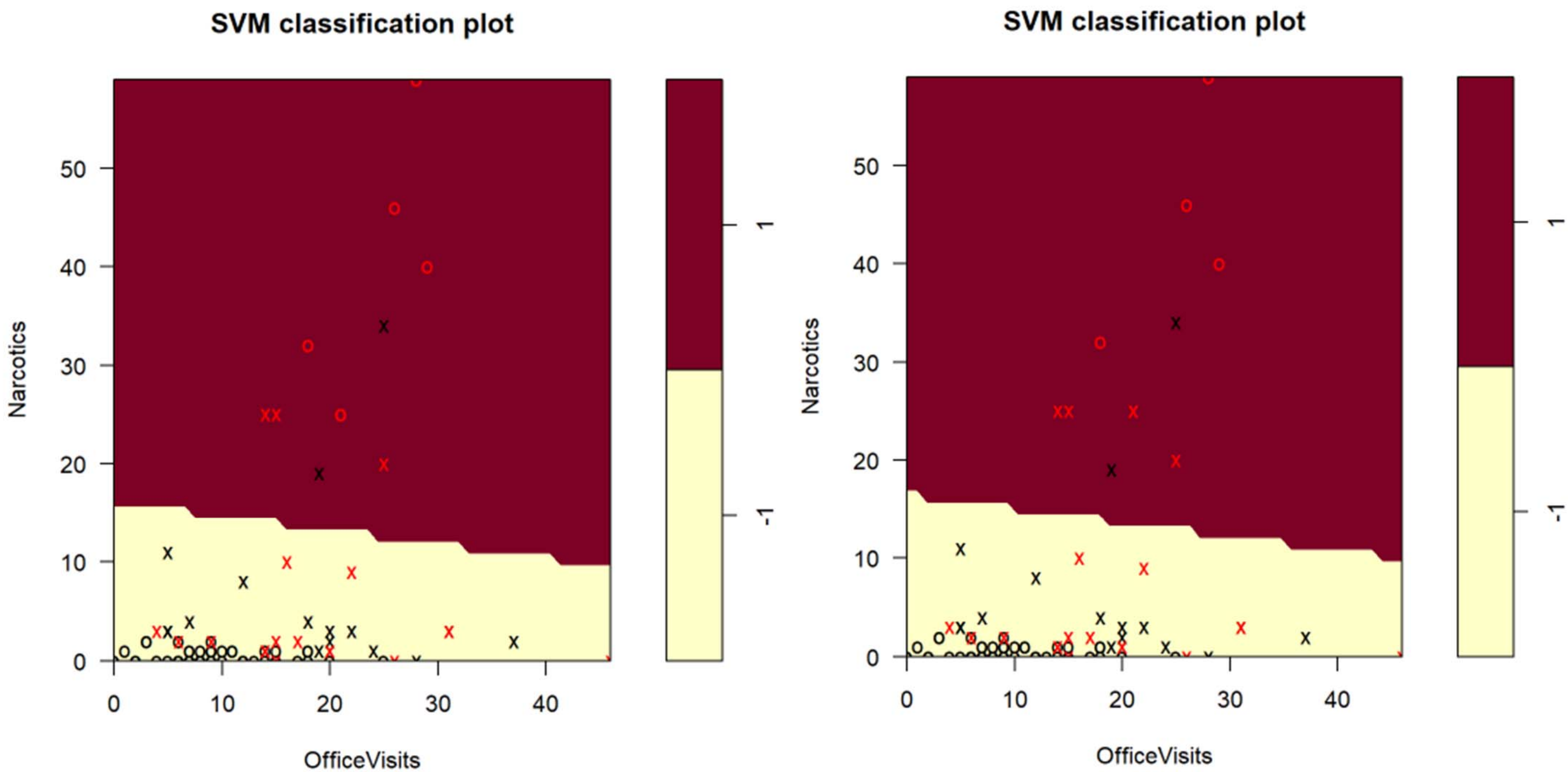
```
[1] Japan
194 Levels: Afghanistan Albania Algeria Andorra Angola Antigua and Barbuda Argentina Armenia Australia Austria
... Zimbabwe
```

Let's create some plots for exploratory data analysis (EDA). First, Let's create a basic scatterplot of GNI versus FertilityRate.

Hide

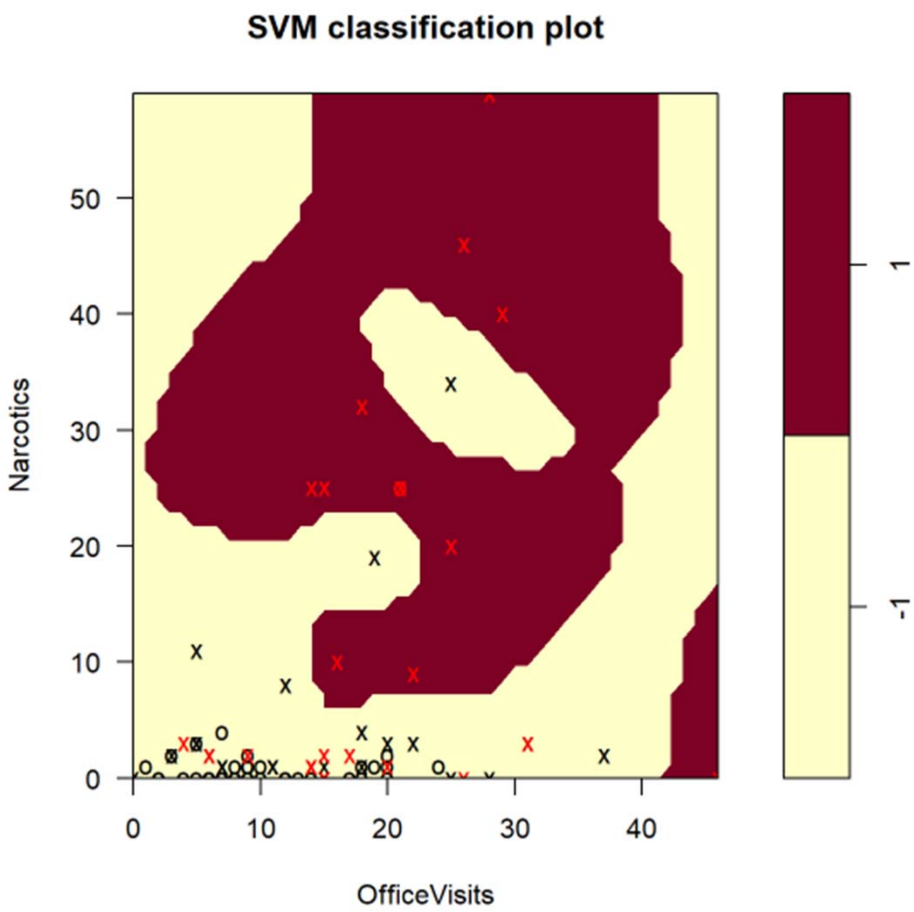
```
plot(WHO$GNI, WHO$FertilityRate)
```


Linear SVM

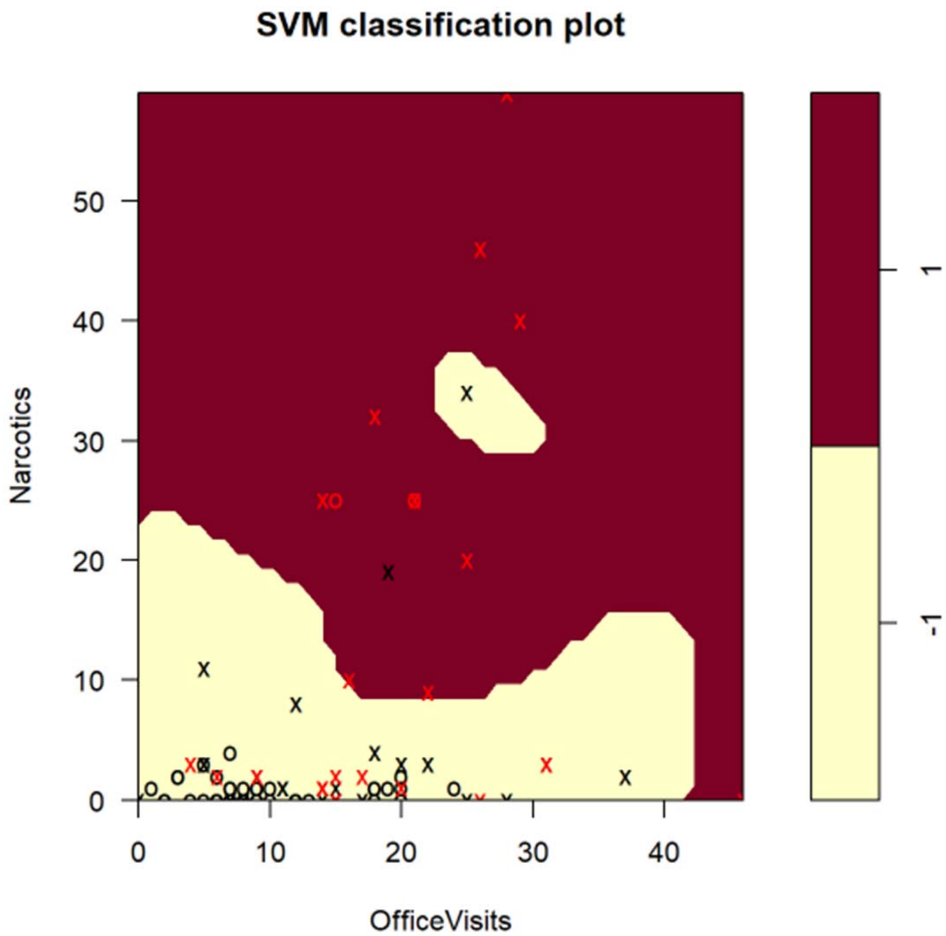


Large C | Small C

SVM with RBF kernel



Large C



Small C

K-fold cross validation and testing

- Best model after tuning
  - Sigmoid kernel

- In-sample accuracy
  - 82.16%

- Out-of-sample
  - 79.49%

Lecture 6 wrap-up

- ✓Linear SVM
  - ✓Model
  - ✓Strategy
  - ✓Algorithm
- ✓Regularization
- ✓Kernels
- ✓Application: diabetes care revisit

Next lecture

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

Questions?

Shan Wang (王杉)

https://wangshan731.github.io/