北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
语言数据分析

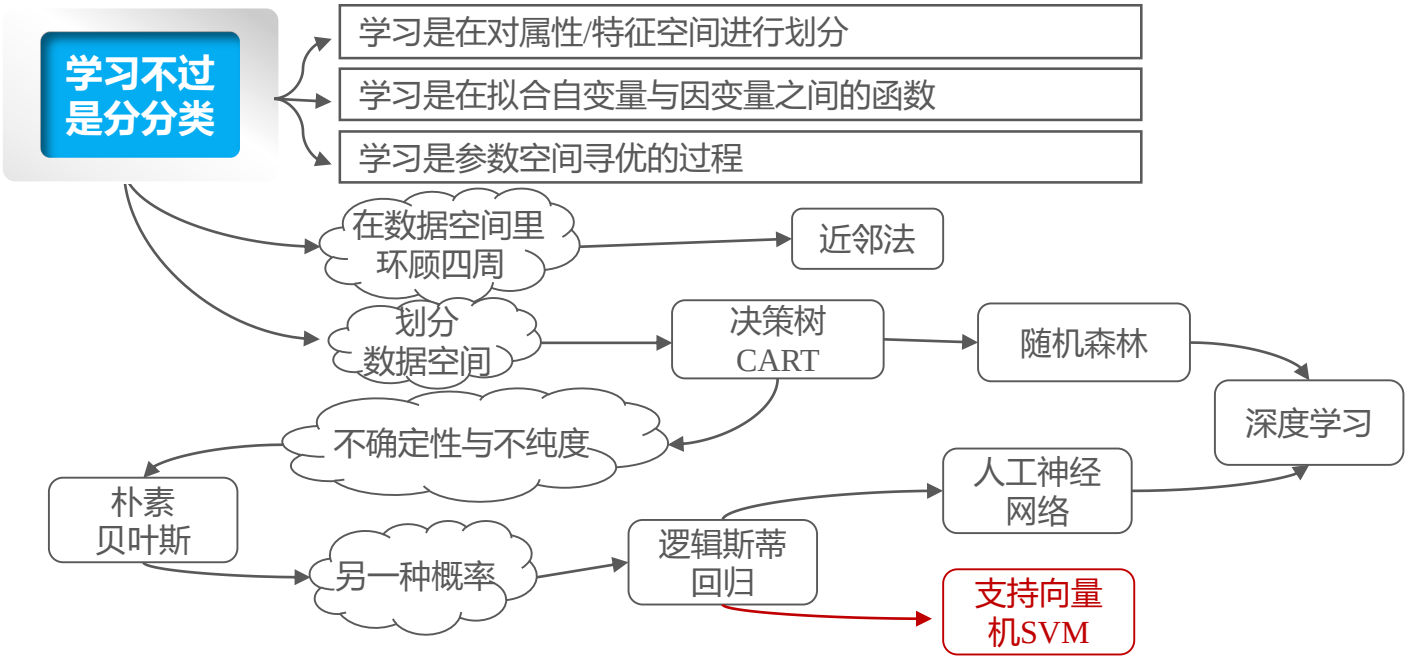
既是世间法、自当有分别

艾新波 / 2018 • 北京

课程体系

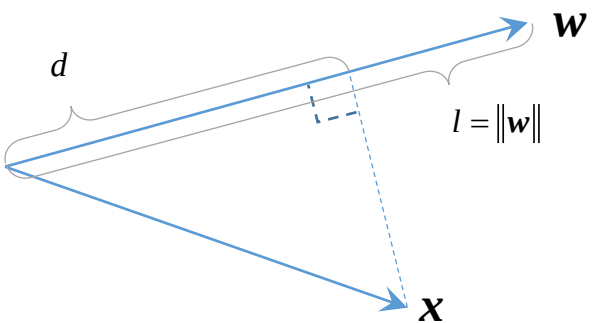
- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

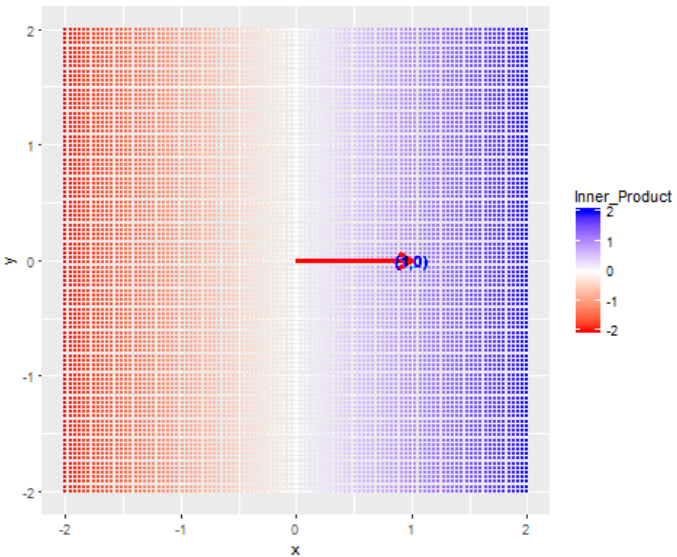
算法模型


学习不过是分分类
学习是在对属性/特征空间进行划分
学习是在拟合自变量与因变量之间的函数
学习是参数空间寻优的过程
在数据空间里环顾四周
近邻法
划分数据空间
决策树CART
随机森林
深度学习
不确定性与不纯度
朴素贝叶斯
人工神经网络
另一种概率
逻辑斯蒂回归
支持向量机SVM

从内积说起

$$\langle w, x \rangle = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

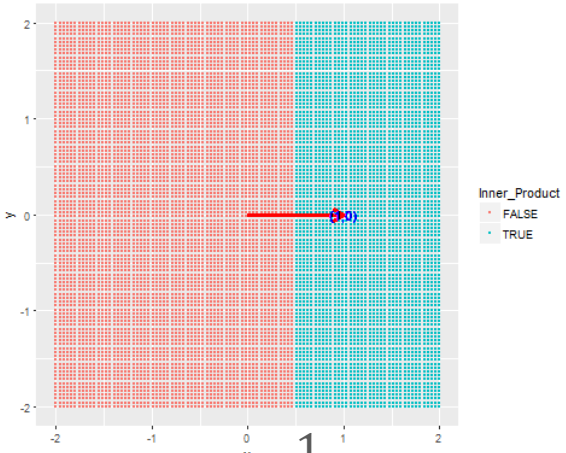




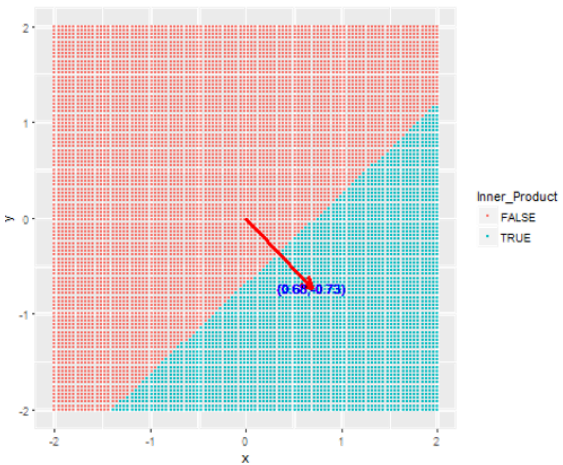
通俗讲：w与x的内积，就是x在w上的投影长度，乘以w的长度

从内积到分类超平面

$w^T x + b = 0$作为一个分类超平面：

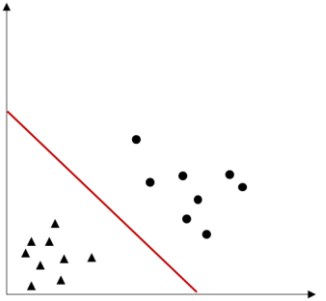


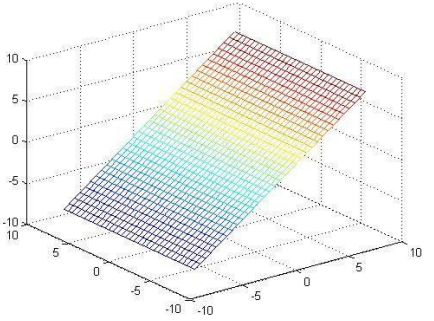
$$b = -\frac{1}{2}$$



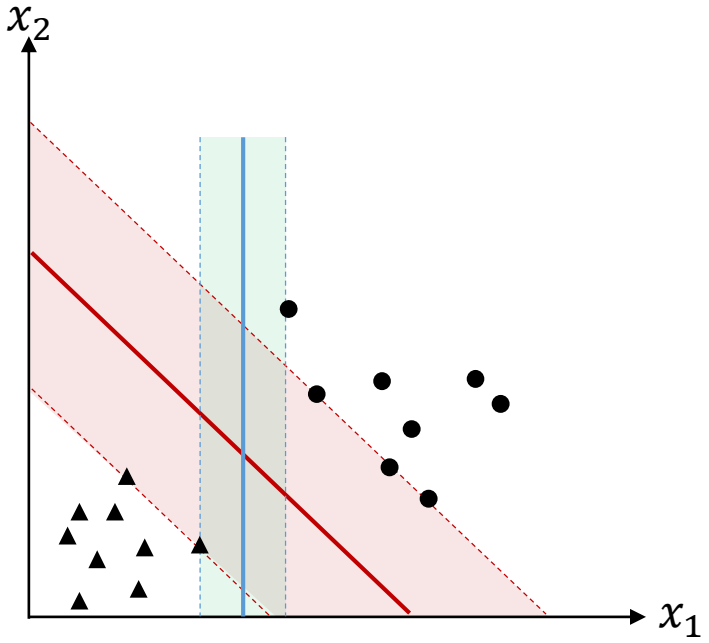
分类超平面

$$w^T x + b = 0$$



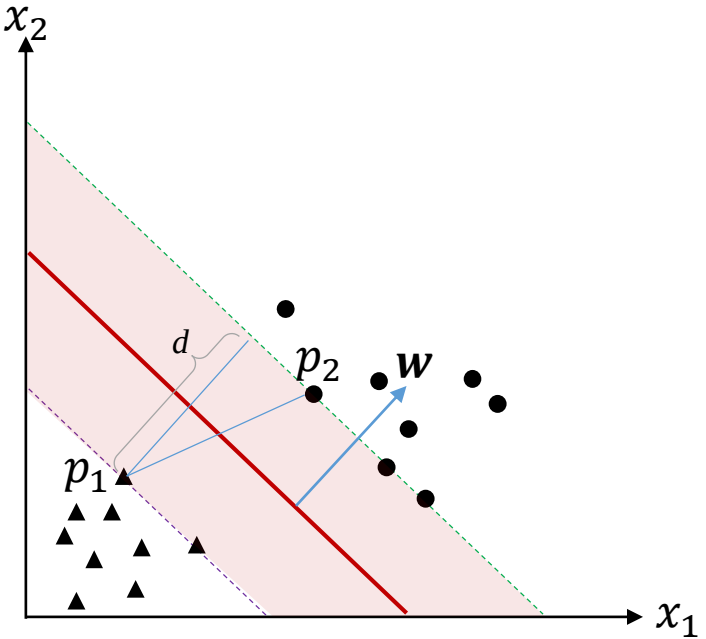


分类超平面



- 要将作图所示的两类点分开，可以有无穷多个超平面
- 相比较而言，红色的部分有更宽的边缘
- “道路”更宽、泛化能力更强

分类超平面



两个平行的边界为：

$$w^T x + b = 1$$

$$w^T x + b = -1$$

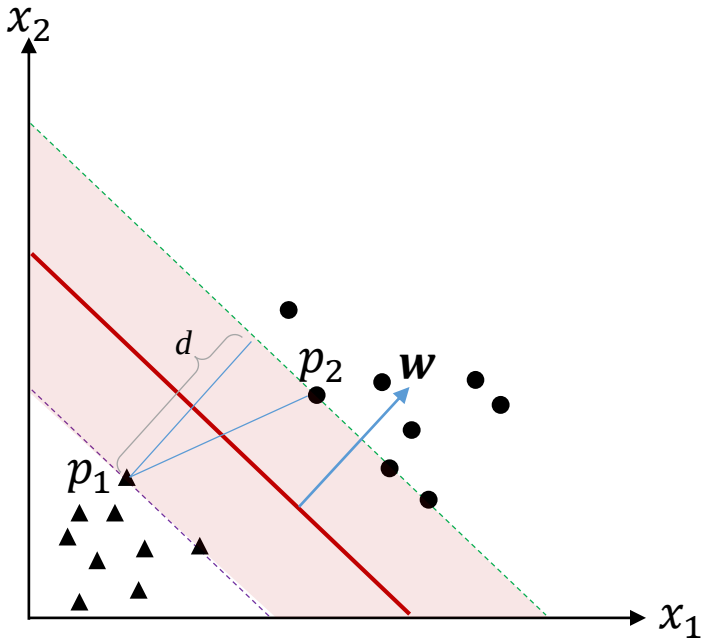
于是有：

$$w^T \left(p_2 - p_1 \right) = 2$$

$$\Rightarrow \|w\| \times d = 2$$

$$\Rightarrow d = \frac{2}{\|w\|}$$

分类超平面



实现间隔最大化，等价于最小化目标函数：

$$\min_{w} \frac{\|w\|^2}{2}$$

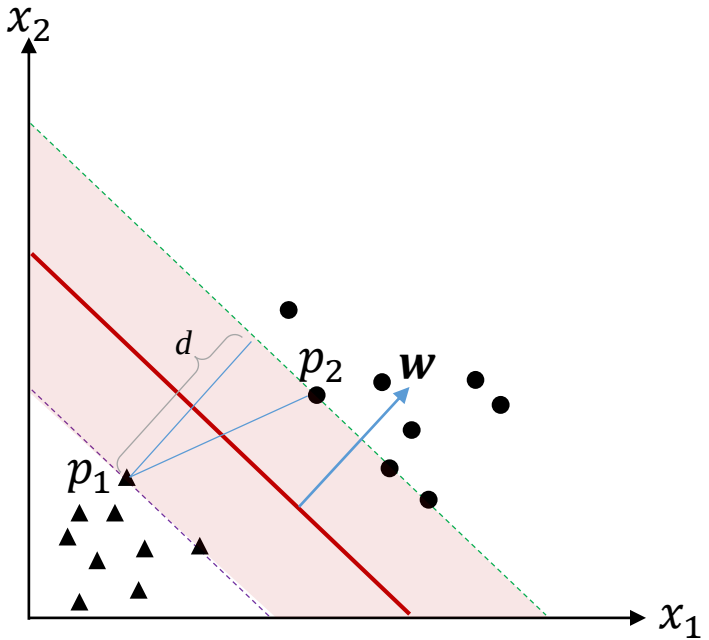
需要满足以下约束：

$$w^T x + b \geq 1, y_i = 1$$

$$w^T x + b \leq -1, y_i = -1$$

即： $y_i \left(w^T x + b \right) \geq 1$

分类超平面



通过拉格朗日法求解

对每个样本引入一个拉格朗日系数：

$$\alpha_i \geq 0, i = 1, 2, \cdots, N$$

权向量最优解： $\boldsymbol{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \boldsymbol{x}^{(i)}$

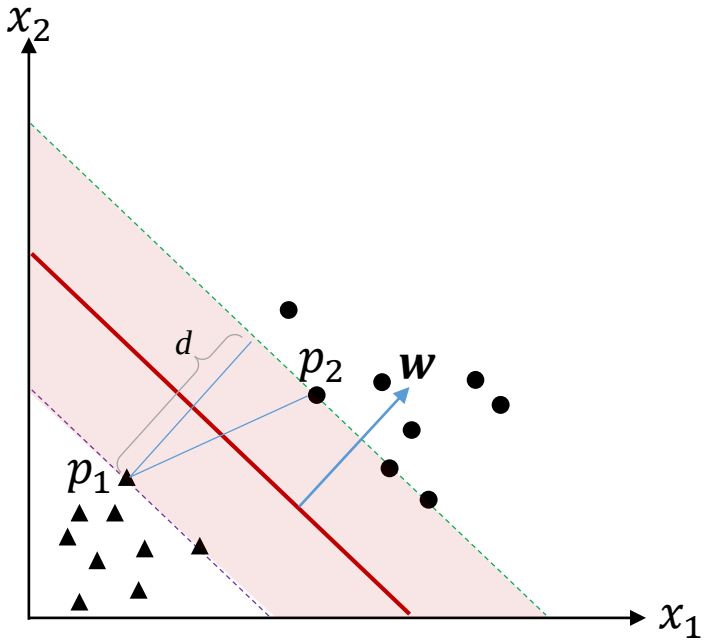
$$\Rightarrow \boldsymbol{w}^T \boldsymbol{x}^{(j)} + b$$

$$= \left(\sum_{i=1}^{N} \alpha_i^* y_i \boldsymbol{x}^{(i)} \right)^T \boldsymbol{x}^{(j)} + b$$

$$= \sum_{i=1}^{N} \alpha_i^* y_i \left(\boldsymbol{x}^{(i)} \right)^T \boldsymbol{x}^{(j)} + b$$

$$= \sum_{i=1}^{N} \alpha_i^* y_i \left\langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \right\rangle + b$$

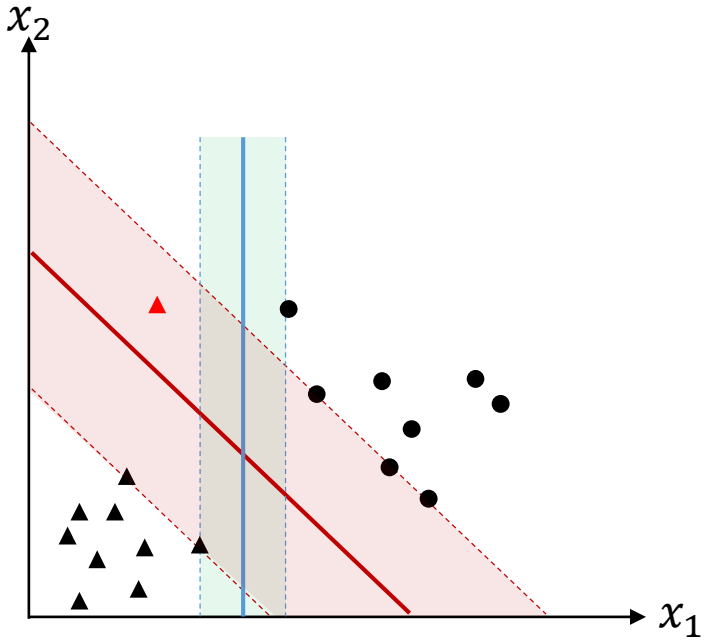
分类超平面



从决策函数可以看出：

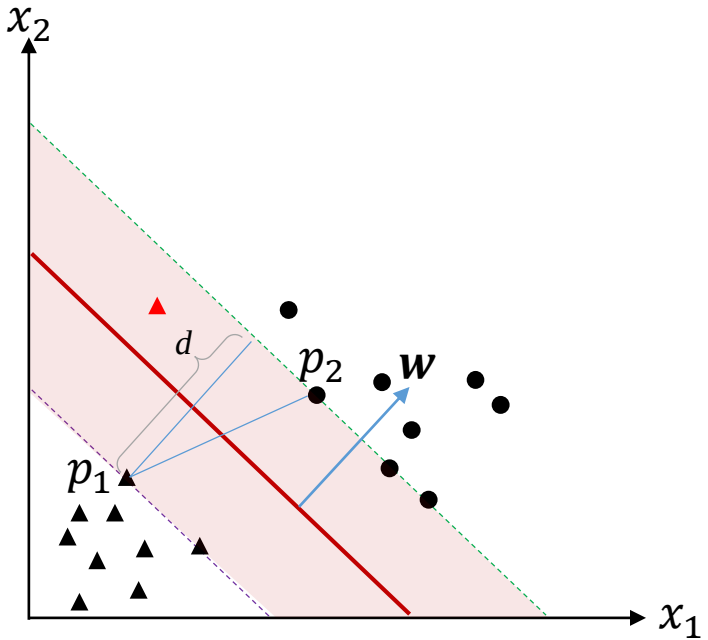
- 对于测试集的预测，只需要计算它与训练数据点的内积即可
- 准确地讲，是**只需要计算它和支持向量的内积即可**，因为非支持向量的α均为零

噪声与软间隔



- 红色的超平面，错分了一个样本，蓝色的超平面完全正确
- 但是，红色道路依然比蓝色的道路更可取
- 软间隔soft margin学习允许一定训练错误的决策边界

噪声与软间隔



引入松弛变量：

$$w^T x^{(i)} + b = 1 - \xi_i, y_i = 1$$

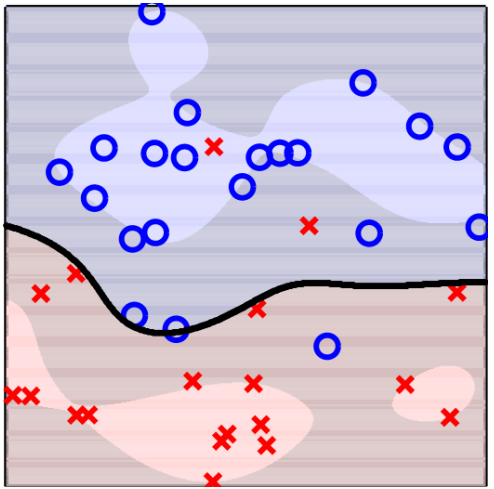
$$w^T x^{(i)} + b = -1 + \xi_i, y_i = -1$$

修正目标函数：

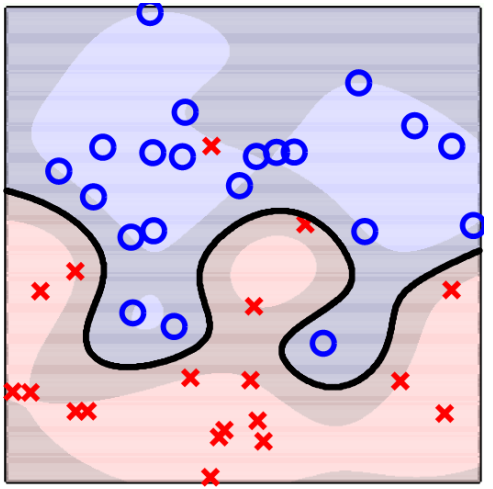
$$\frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k$$

C越大，对于噪声的容忍就越小

噪声与软间隔

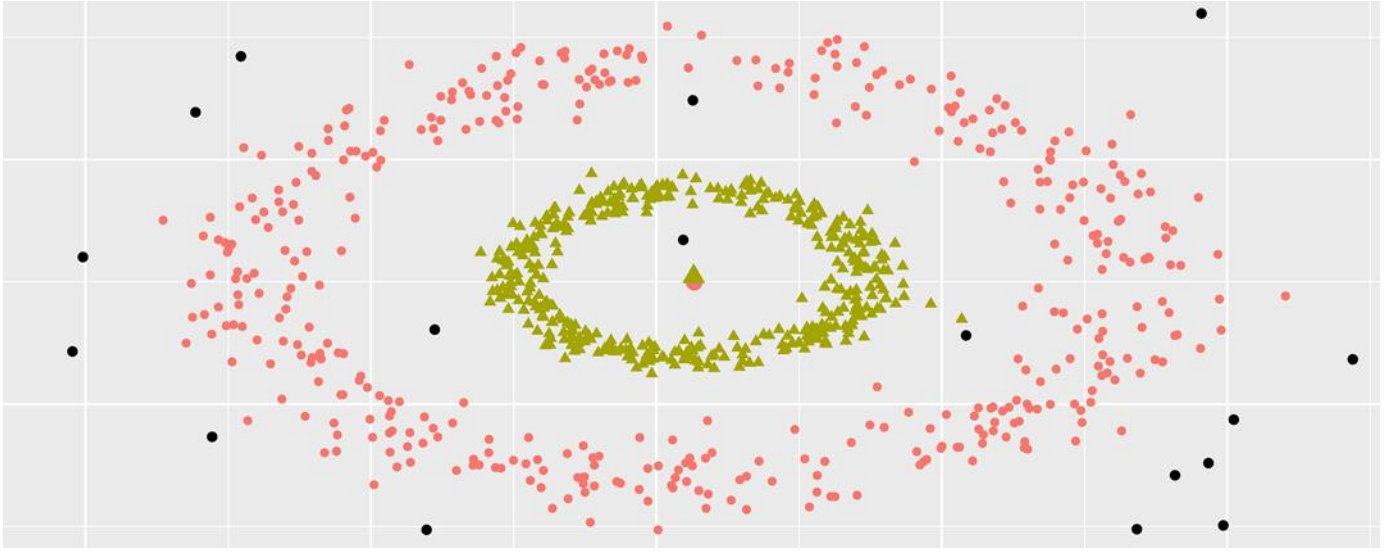


C=1

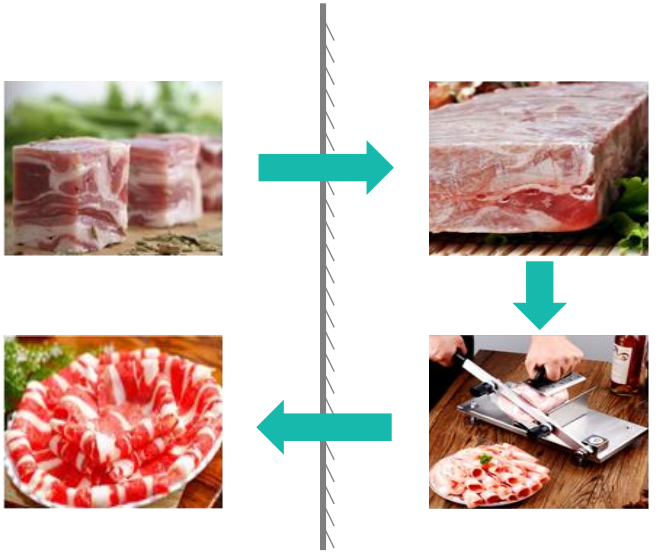


C=10

非线性可分问题

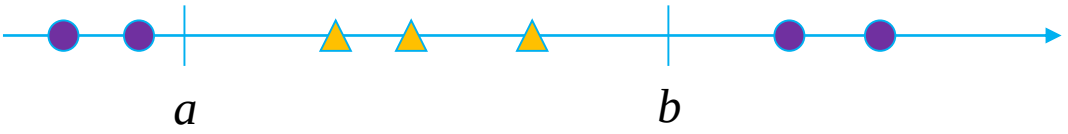


非线性可分：映射与变换



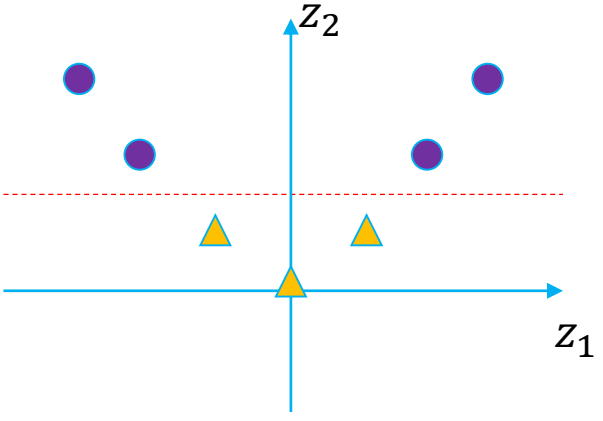
当问题不好解决时，变换到另一个空间，问题往往可以得到解决

支持向量机：线性不可分



$$g(x)=(x-a)(x-b)$$
$$=x^2-(a+b)x+ab$$

$$z_1=x, z_2=x^2$$



支持向量机：线性不可分

对于非线性问题，需要做到：

第一，将属性空间映射到特征空间；

第二，在特征空间中采用线性分类器进行分类。

决策规则为：$f(\boldsymbol{x})=\sum_{i=1}^{l}\alpha_i y_i\left\langle\phi\left(\boldsymbol{x}^{(i)}\right),\phi(\boldsymbol{x})\right\rangle+b$

其中，ϕ 为属性空间到特征空间的映射：$\phi:\mathcal{X}\to\mathcal{F}$

引入核方法：

一对向量的核函数等于变换后空间中计算这对向量的内积：$K(\boldsymbol{x},\boldsymbol{z})=\left\langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\right\rangle$

也就是说，有了核函数，我们甚至不需要知道如何具体映射到高维空间

支持向量机：来自libsvm的一些建议

Tips on practical use

- Note that SVMs may be very sensible to the proper choice of parameters, so allways check a range of parameter combinations, at least on a reasonable subset of your data.
- For classification tasks, you will most likely use C-classification with the RBF kernel (default), because of its good general performance and the few number of parameters (only two: C and γ). The authors of libsvm suggest to try small and large values for C—like 1 to 1000—first, then to decide which are better for the data by cross validation, and finally to try several γ's for the better C's.

https://cran.r-project.org/web/packages/e1071/ Vignettes:

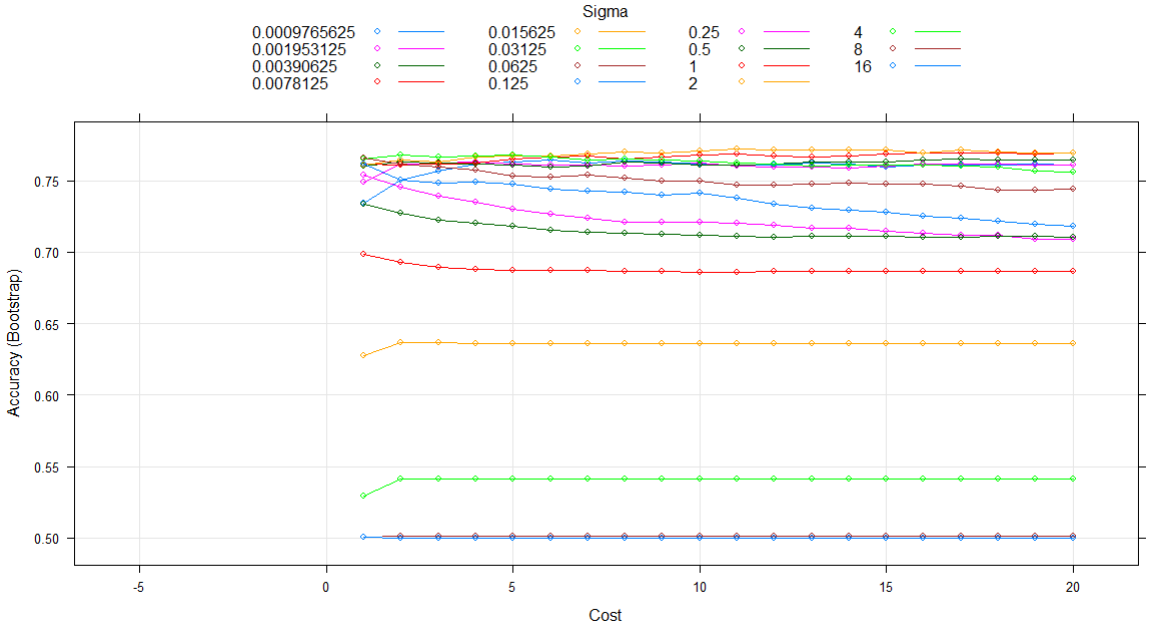
R语言实现

```
library(kernlab)
imodel <- ksvm(wlfk~., data = cjb[train_set_idx, ])
predicted_train <- predict(
  imodel, newdata = cjb[train_set_idx,], type = "response")
Metrics::ce(cjb$wlfk[train_set_idx], predicted_train)
#> [1] 0.1497227
predicted_test <- predict(imodel,
  newdata = cjb[-train_set_idx,], type = "response")
Metrics::ce(cjb$wlfk[-train_set_idx], predicted_test)
#> [1] 0.1802575
```

R语言实现

```
library(caret)
svm_grid <- expand.grid(
  sigma = 2^(-10:4), C= -5:20)
set.seed(2012)
imodel <- train(wlfk~., data = cjb[train_set_idx, ],
    method = "svmRadial", preProc = c("center", "scale"),
    tuneGrid = svm_grid)
imodel$bestTune
#>       sigma  C
# 121 0.015625 11
```

R语言实现


Sigma
0.0009765625
0.001953125
0.00390625
0.0078125
0.015625
0.03125
0.0625
0.125
0.25
0.5
1
2
4
8
16
Accuracy (Bootstrap)
0.75
0.70
0.65
0.60
0.55
0.50
-5
0
5
10
15
20
Cost

R语言实现

```
sp <- Sys.time() #记录开始时间
cat("\n[Start at:", as.character(sp))
for (i in 1:length(kfolds)) {
  curr_fold <- kfolds[[i]] #当前这一折
  train_set <- cjb[-curr_fold,] #训练集
  test_set <- cjb[curr_fold,] #测试集
  imodel_kfold <-  ksvm(wlfk~., data = train_set
      C = imodel$bestTune$C, gamma = imodel$bestTune$sigma)
  predicted_train <-  predict(imodel_kfold,
           train_set, type = "response")
```

R语言实现

```
  imetrics("ksvm", "Train",
           predicted_train, train_set$wlfk)
  predicted_test <- predict(imodel_kfold,
           test_set, type = "response")
  imetrics("ksvm", "Test",
           predicted_test, test_set$wlfk)
}
ep <- Sys.time()
cat("\tFinised at:", as.character(ep), "]\n")
cat("[Time Ellapsed:\t",
    difftime(ep, sp, units = "secs"), " seconds]\n")
```

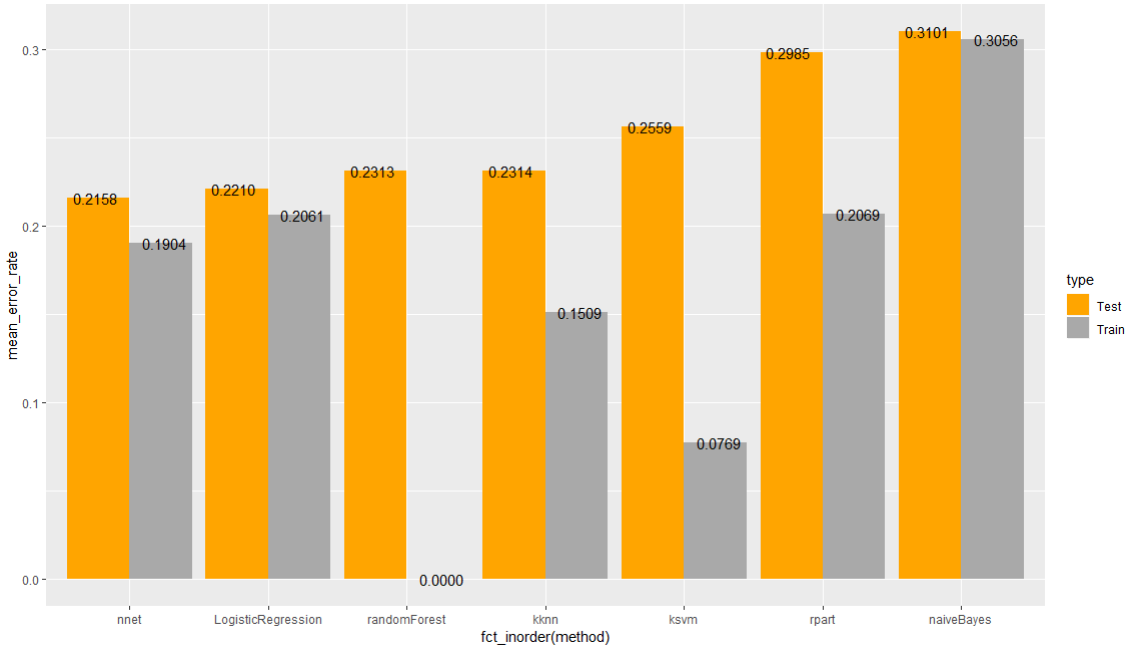
R语言实现

```
#> method  type  accuracy error_rate
#> 1    ksvm Train 0.9181034 0.08189655
#> 2    ksvm  Test 0.7307692 0.26923077
#> 3    ksvm Train 0.9252874 0.07471264
#> 4    ksvm  Test 0.7692308 0.23076923
#> 5    ksvm Train 0.9181034 0.08189655
#> 6    ksvm  Test 0.8205128 0.17948718
.....
#> 19   ksvm Train 0.9167862 0.08321377
#> 20   ksvm  Test 0.7662338 0.23376623
```

R语言实现

```
global_performance %>%
  group_by(method, type) %>%
  summarise(mean_error_rate = mean(error_rate)) %>%
  arrange(type, mean_error_rate) %>%
  ggplot(aes(x = fct_inorder(method), y = mean_error_rate,
             fill = type)) +
  geom_bar(stat= "identity", position = "dodge") +
  geom_text(aes(label = format(mean_error_rate, digits = 3)),
            position = position_dodge(width = 1))+
  scale_fill_manual(values=c("orange","darkgrey")) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

模型综合评估


mean_error_rate
0.3
0.2
0.1
0.0
0.2158
0.1904
0.2210
0.2061
0.2313
0.0000
0.2314
0.1509
0.2559
0.0769
0.2985
0.2069
0.3101
0.3056
nnet
LogisticRegression
randomForest
kknn
ksvm
rpart
naiveBayes
fct_inorder(method)
type
Test
Train

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ： 23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics