Boosting

Филип Матяш

Ключевая идея

Группируем множество (слабых самих по себе) предикторов
Рассчитываем вес и добавляем их
Получаем сильный предиктор

Что “под капотом”?

Начинаем с набора классификаторов (К1-Kn)

пример - деревья, регрессионые модели, срезы, итд..

Создаем классификатор, который комбинирует функции классификации

Цель — минимизировать ошибку

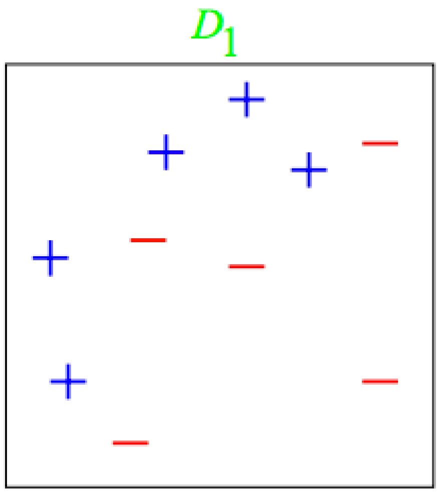
1)Итеративно добавляем классификатор

2)Рассчитываем веса основываясь на ошибке

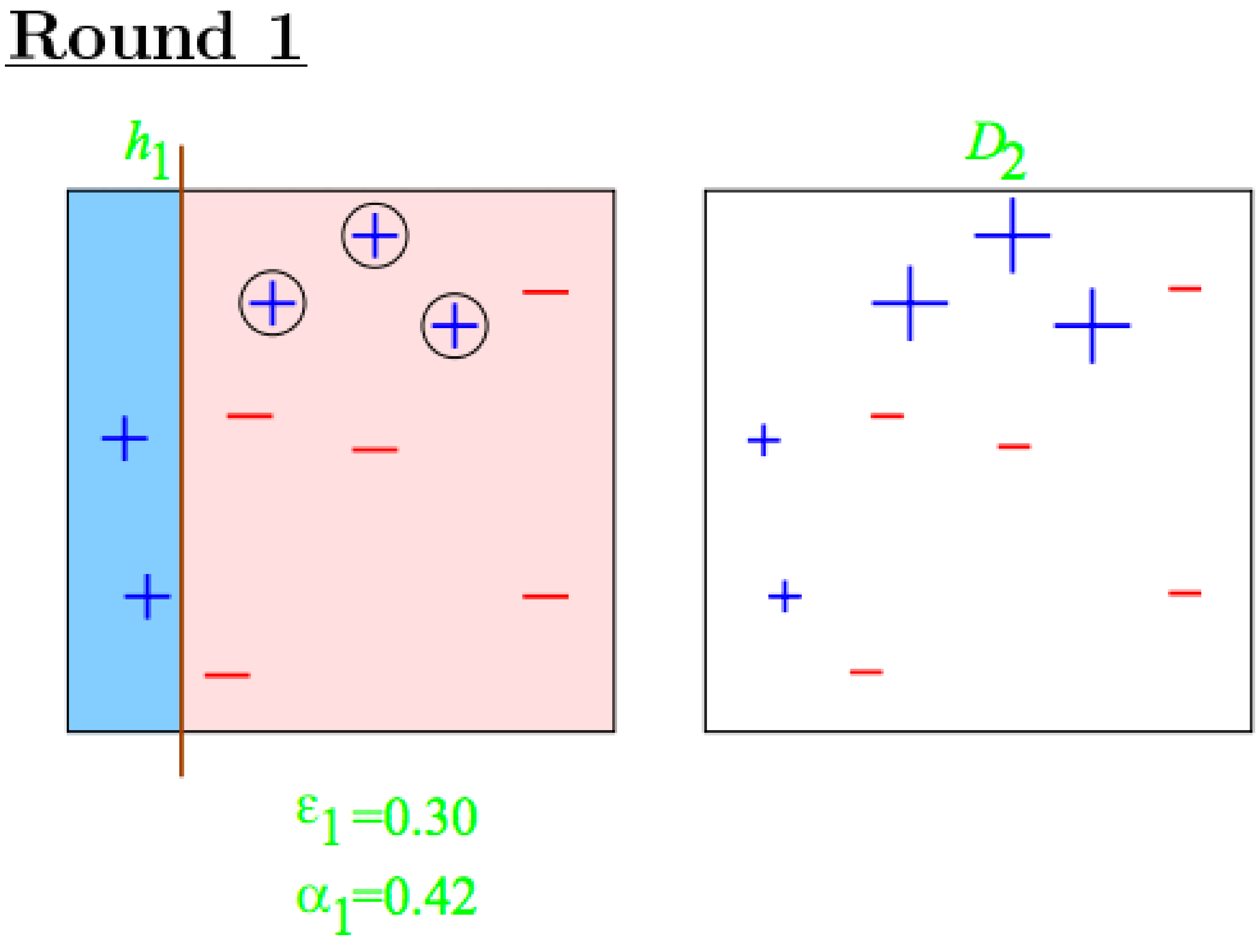
3)Повышаем вес неверно классифицированных значений и предиктор

$$f(x) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Простой пример

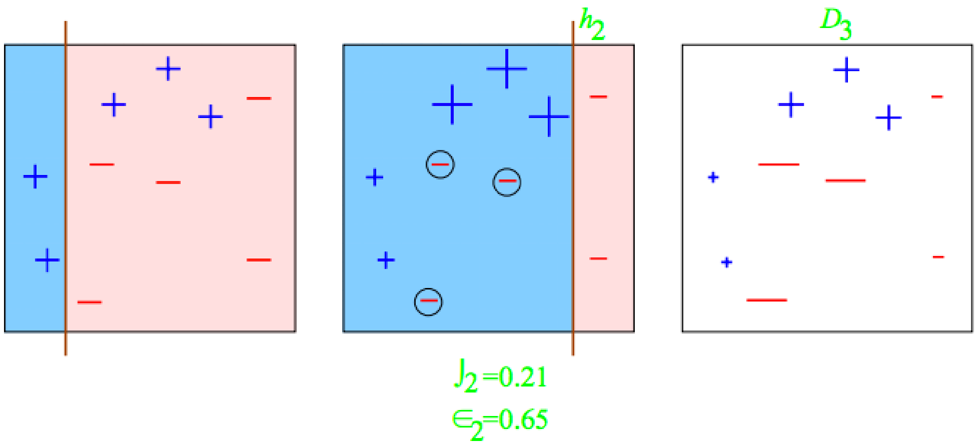


Пример

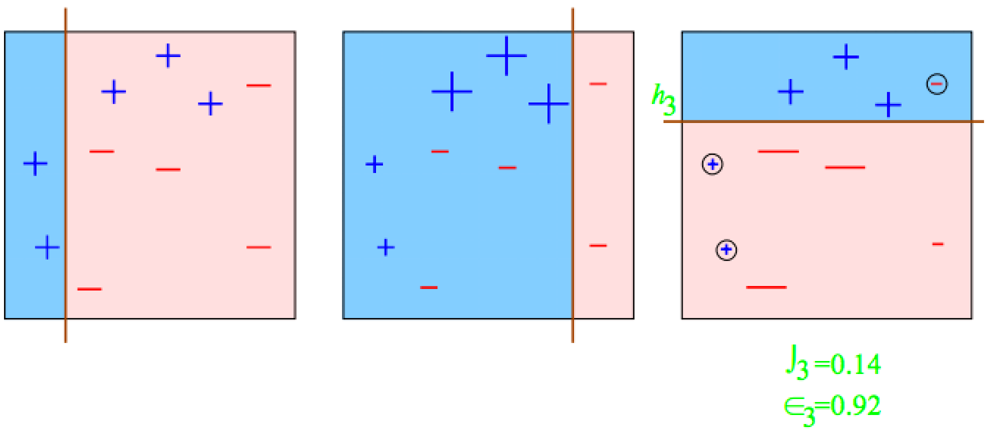


Пример

Round 2

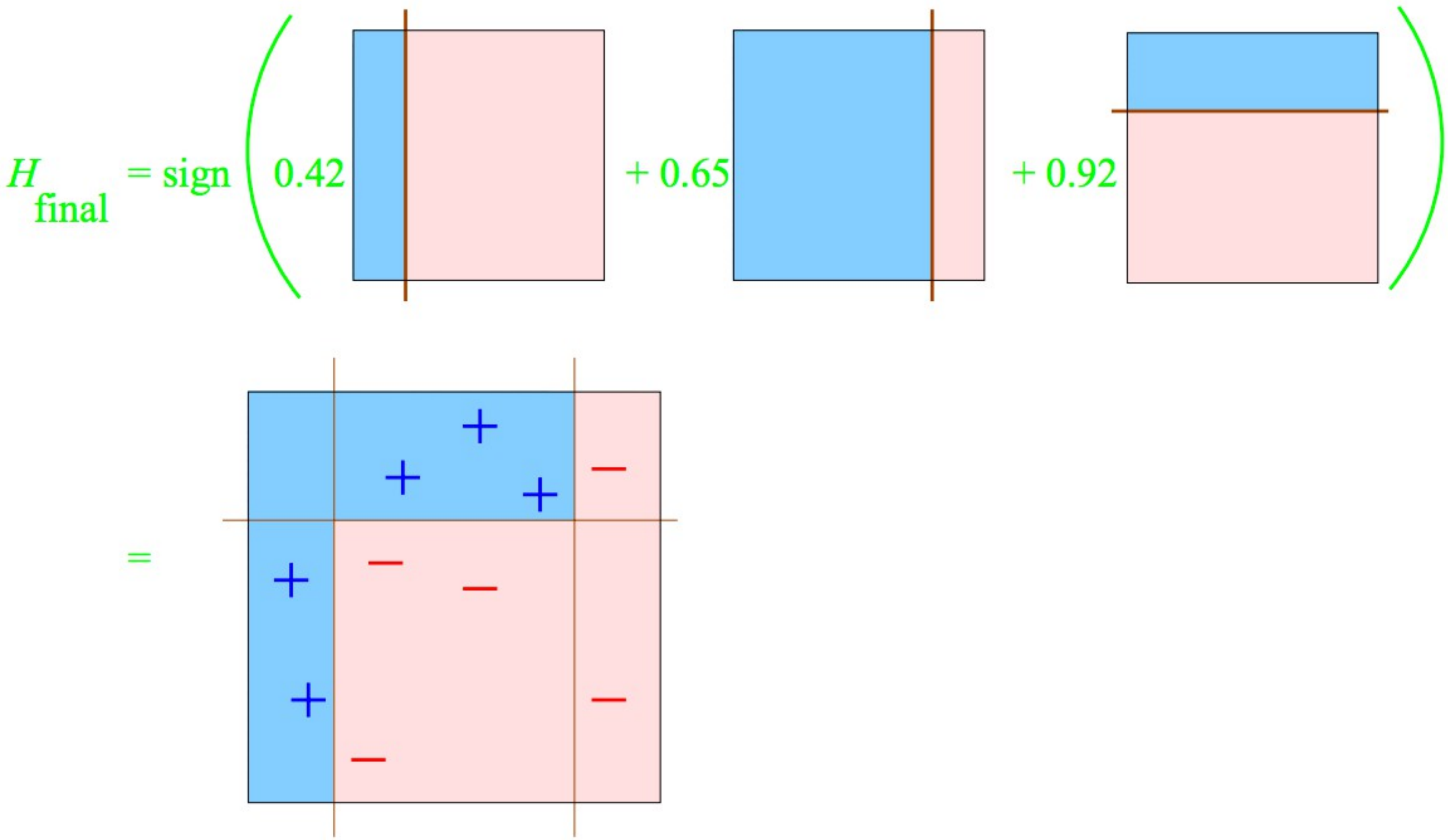


Round 3



Пример

Final Hypothesis



Boosting in R

Boosting можно использовать с любым нобором классификаторов

Отдельный саб-класс gradient boosting

R имеет множество пакетов и методов

- gbm - boosting with trees.
- mboost - model based boosting
- ada - statistical boosting based on additive logistic regression
- gamBoost for boosting generalized additive models

Большинство есть в caret пакете

Еще пример

```
library(ISLR); data(Wage); library(ggplot2);
library(caret);
Wage <- subset(Wage,select=-c(logwage))
inTrain <- createDataPartition(y=Wage$wage,
                              p=0.7, list=FALSE)
training <- Wage[inTrain,]; testing <- Wage[-
inTrain,]
#Fit the model

modFit <- train(wage ~ .,
method="gbm",data=training,verbose=FALSE
)
```

Other

http://en.wikipedia.org/wiki/AdaBoost
http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf
http://www.cc.gatech.edu/~thad/6601-gradAI-fall2013/boosting.pdf

Model based prediction

Basic idea

Предпологает, что данные следуют вероятностной модели

Использует Теорему Баеса для поиска оптимальных классификаторов

Плюсы:

- Дополнительные знания из структуры данных
- Не требовательно к вычислительным мощностям
- Достаточно точны в реальных условиях

Cons:

- Дополнительные предположения о данных (в случае ложных могут негативно повлиять на модель)

Model based approach

Цель — построить параметрическую модель для условного (вероятностного) распределения $P(Y = k \mid X = x)$

Обычно используют

Теорему Баеса:

$$\Pr(Y = k \mid X = x) = \frac{\Pr(X = x \mid Y = k)\Pr(Y = k)}{\sum_{\ell=1}^{K} \Pr(X = x \mid Y = \ell)\Pr(Y = \ell)}$$

$$\Pr(Y = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

Обычно априорная вероя

Для $f(x)$ чаще всего испол

параметры которого мы получаем из данных.

Признак принадлежности классу — при максимальном $P(Y = k \mid X = x)$

$$\frac{1}{\sigma_k\sqrt{2\pi}} e^{-\frac{(x-\mu_k)^2}{\sigma_k^2}}$$

Классификационные модели

Linear discriminant analysis assumes f(x) is multivariate Gaussian with same covariances
Quadratic discrimant analysis assumes f(x) is multivariate Gaussian with different covariances
Model based prediction assumes more complicated versions for the covariance matrix
Naive Bayes assumes independence between features for model building

LDA

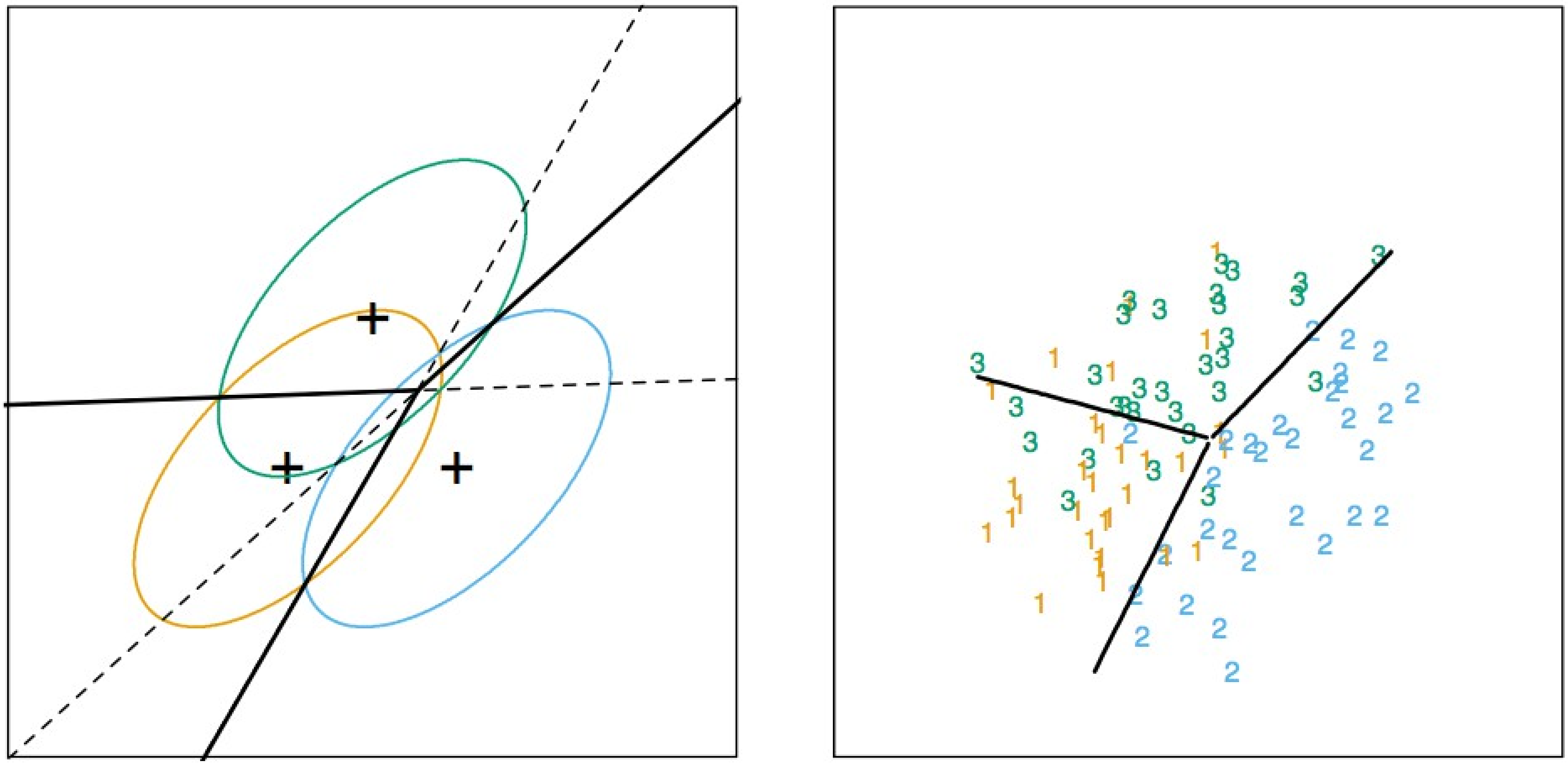
$$\log \frac{\Pr(Y = k|X = x)}{\Pr(Y = j|X = x)}$$

$$= \log \frac{f_k(x)}{f_j(x)} + \log \frac{\pi_k}{\pi_j}$$

$$= \log \frac{\pi_k}{\pi_j} - \frac{1}{2} (\mu_k + \mu_j)^T \Sigma^{-1} (\mu_k + \mu_j)$$

$$+ x^T \Sigma^{-1} (\mu_k - \mu_j)$$

Границы классификации



Discriminant function

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k \Sigma^{-1} \mu_k + \log(\mu_k)$$

Решение о принадлежности к классу принимается путем максимизации функции

Naive Bayes

Используя Т. Баеса получаем:

$$P(Y = k|X_1, \dots, X_m) = \frac{\pi_k P(X_1, \dots, X_m|Y = k)}{\sum_{\ell=1}^{K} P(X_1, \dots, X_m|Y = k)\pi_\ell}$$

$$\propto \pi_k P(X_1, \dots, X_m|Y = k)$$

Развернутый вид:

$$P(X_1, \dots, X_m, Y = k) = \pi_k P(X_1|Y = k)P(X_2, \dots, X_m|X_1, Y = k)$$

$$= \pi_k P(X_1|Y = k)P(X_2|X_1, Y = k)P(X_3, \dots, X_m|X_1, X_2, Y = k)$$

Naive Bayes

$$= \pi_k P(X_1|Y = k)P(X_2|X_1, Y = k) \dots P(X_m|X_1 \dots, X_{m-1}, Y = k)$$

$$\approx \pi_k P(X_1|Y = k)P(X_2|Y = k) \dots P(X_m|, Y = k)$$

Пример

Example: Iris Data
data(iris); library(ggplot2)
names(iris)
table(iris$Species)
Create training and test sets
inTrain <- createDataPartition(y=iris$Species,
p=0.7, list=FALSE)
training <- iris[inTrain,]
testing <- iris[-inTrain,]
dim(training); dim(testing)

Пример

Build predictions

```
modlda = train(Species ~
.,data=training,method="lda")
modnb = train(Species ~ .,
data=training,method="nb")
plda = predict(modlda,testing); pnb =
predict(modnb,testing)
table(plda,pnb)
```

Comparison of results

```
equalPredictions = (plda==pnb)
qplot(Petal.Width,Sepal.Width,colour=equalPre
```

other

http://statweb.stanford.edu/~tibs/ElemStatLearn

https://en.wikipedia.org/wiki/Linear_discriminant_analysis

http://www.stat.washington.edu/raftery/Research/PDF/fraley2002.pdf