Seminar 1. Boosting

Alexander Romanenko

MIPT, SRR

02/09/2015
Moscow

Table of content

Alexey Romanenko

- alexromsput@gmail.com
- Education:

  MIPT, Faculty of Management and Applied Mathematics
- Work experience:
  - Forecsys
  - Svyaznoy Logistics
  - SAS Institute

Alexander Romanenko

- angriff07@gmail.com
- Education:

  MIPT, Faculty of Management and Applied Mathematics
- Work experience:
  Samsung Research Institute Russia

Course programme

- Deep Leaning, Autoencoders, CNN, RNN
- Compositions
- Feature selection
- Unsupervised learning
- Optional (forecasting, reinforcement learning, topic modeling)

Remarks

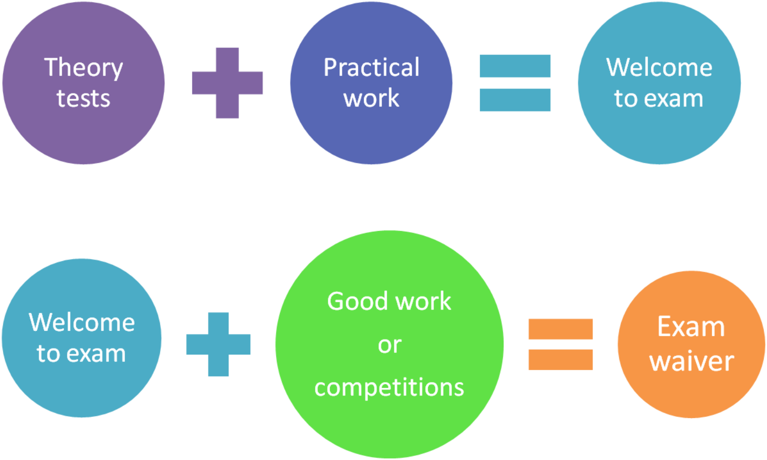
Environment

- Python (notebook)
- R
- Matlab / Octave
- Weka, RapidMiner, Orange
- Mahout, VowpalWabbit

Seminar materials:

Materials on Google drive

Game rules


Theory tests
+
Practical work
=
Welcome to exam
Welcome to exam
+
Good work or competitions
=
Exam waiver

Reminder

- Classification
    - Metric methods;
    - Stochastic methods;
    - SVM;
    - Logit-Regressions;
    - Bayes-classificator
- Regression
    - OLS;
    - Ridge-Reegression;
    - LASSO;
    - Elastic-net;

Reminder

- Clustering
    - Graph based algorithms
    - FOREL
    - k-means
    - EM-algorithm
- Time Series forecasted methods
    - Regression (LAWR)
    - LOWESS (Localy Weighted Scatter plot Smoothing) (непараметрическая регрессия)
    - ES (экспоненципальное сглаживание)
    - ARIMA, ARMA
- NN
    - for classification
    - for regression

Composition

$X^l = (x_i, y_i)_{i=1}^l \in X \times Y$ — обучающая выборка, $y_i = y^*(x_i)$;

$a(x) = C(b(x))$ — алгоритм, где

$b : X \to R$ — базовые алгоритмы,

$C : R \to Y$ — решающее правило,

R — пространство оценок

Определение

Композиция базовых алгоритмов $b_1, \dots, b_T$

$$a(x) = C(F(b_1(x), \dots, b_T(x))),$$

где $F : R^T \to R$ — корректирующий оператор

Зачем вводится R? $\{F : R^T \to R\}$??? $\{F : Y^T \to Y\}$

Questions

R, C, b?

- Task of classification, $Y = -1, +1$
- Task of classification, $Y = 1, \dots, M$
- Task of regression, $Y = \mathbb{R}$

Idea

Quality functional of base algorithms, $\mathcal{L}$ — loss function

$$Q(b, X^l) = \sum_{i=1}^{l} \mathcal{L}(b(x_i), y_i)$$

Greedy iterative process:

$$b_1 = \arg\min_{b} Q(b, X^l) \quad (1)$$

$$b_2 = \arg\min_{b,F} Q(F(b_1, b), X^l)$$

$\dots$

$$b_t = \arg\min_{b,F} Q(F(b_1, \dots, b_{t-1}, b), X^l) \quad (2)$$

Reduce (2) to (1), but with weights of objects and may be another loss function

$$b_t = \arg\min_{b} \sum_{i=1}^{l} w_i \tilde{\mathcal{L}}(b(x_i), y_i).$$

Boosting for binary classification

$Y = -1, +1$, $b_t : X \to -1, 0, +1$, $C(b) = \text{sign}(b)$
Weighted voting

$$a(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t b_t(x)), x \in X.$$

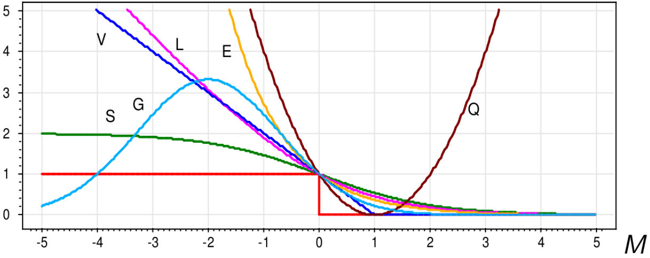
Number of error on X^l:

$$Q_T = \sum_{i=1}^{l} \left[y_i \sum_{t=1}^{T} \alpha_t b_t(x_i) < 0 \right]$$

Heuristics:

- Fix $\alpha_1 b_1(x), \ldots, \alpha_{t-1} b_{t-1}(x)$, then find $\alpha_t b_t(x)$
- Plain approximation of $[M \leqslant 0]$

Loss functions



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LogitBoost);
$Q(M) = (1 - M)^2$ — квадратичная (GentleBoost);
$G(M) = \exp(-cM(M + s))$ — гауссовская (BrownBoost);
$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная;
$V(M) = (1 - M)_+$ — кусочно-линейная (из SVM);

Main theorem. Notation. AdaBoost

Loss function approximation:

$$Q \leqslant \tilde{Q}_T = \sum_{i=1}^{l} \exp\left(-y_i \sum_{t=1}^{T-1} \alpha_t b_t(x_i)\right) \exp(-y_i \alpha_t b_t(x_i))$$

Normalizing weights: $\tilde{W}^l = (\tilde{w}_1, \dots, \tilde{w}_l)$, $\tilde{w}_i = w_i / \sum_{j=1}^{l} w_j$.

Weighted number of positive and negative classification:

$$N(b, \tilde{W}^l) = \sum_{i=1}^{l} \tilde{w}_i [b(x_i) = -y_i],$$

$$P(b, \tilde{W}^l) = \sum_{i=1}^{l} \tilde{w}_i [b(x_i) = y_i]$$

$1 - N - P$ weighted number of rejections

Main theorem. AdaBoost

Theorem (Freund, Schapire, 1995)

1. $P = 1 - N$
2. $\forall U^l : \sum_{u \in U^l} u = 1 \ \exists b : N(b; U^l) < \frac{1}{2}$.

If (1) and (2) then $\tilde{Q}_T$ reaches minimum when

$$b_T = \operatorname*{arg\,min}_{b \in B} N(b, \tilde{W}^l),$$

$$\alpha_T = \frac{1}{2} ln \frac{1 - N(b, \tilde{W}^l)}{N(b, \tilde{W}^l)}$$

Loss functions

Вход: обучающая выборка X^ℓ; параметр T;

Выход: базовые алгоритмы и их веса $\alpha_t b_t$, $t = 1, \dots, T$;

1: инициализировать веса объектов:
$w_i := 1/\ell, \ \ i = 1, \dots, \ell;$

2: **для всех** $t = 1, \dots, T$

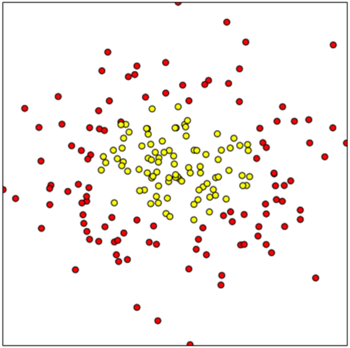
3: обучить базовый алгоритм:
$b_t := \arg\min_b N(b; W^\ell);$

4: $\alpha_t := \dfrac{1}{2} \ln \dfrac{1 - N(b_t; W^\ell)}{N(b_t; W^\ell)};$

5: обновить веса объектов:
$w_i := w_i \exp\bigl(-\alpha_t y_i b_t(x_i)\bigr), \ \ i = 1, \dots, \ell;$

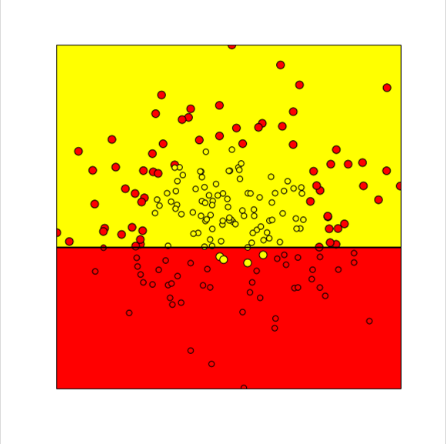
6: нормировать веса объектов:
$w_0 := \sum_{j=1}^{\ell} w_j;$
$w_i := w_i / w_0, \ \ i = 1, \dots, \ell;$

AdaBoost on decision stumps



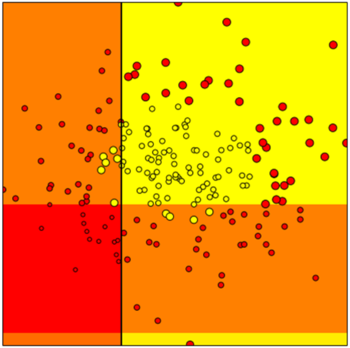
AdaBoost on decision stumps

Iter 1: $Q = 200.61$, $Errors = 64$



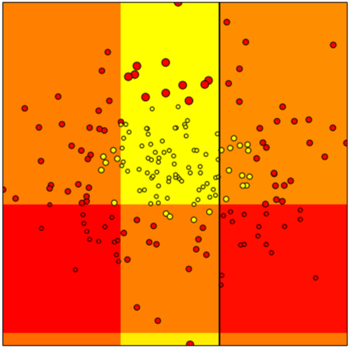
AdaBoost on decision stumps

Iter 3: $Q = 190.93$, $Errors = 95$



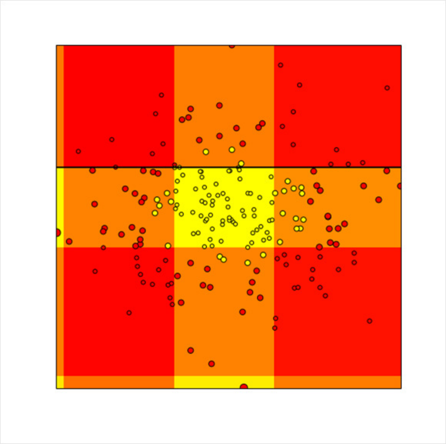
AdaBoost on decision stumps

Iter 4: $Q = 179.31$, $Errors = 78$



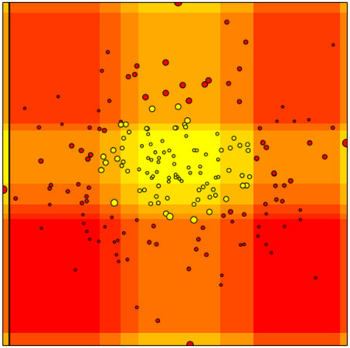
AdaBoost on decision stumps

Iter 6: $Q = 168.20$, $Errors = 27$



AdaBoost on decision stumps

Iter 20: $Q = 129.4$, $Errors = 10$



AdaBoost on decision stumps

AdaBoost Decision Map