Introduction to Machine Learning and Data Mining
Ensemble classifiers

Dmitry Ignatov

National Research University Higher School of Economics
Faculty of Computer Science
Department of Data Analysis and Artificial Intelligence

2019

Outline

Outline

Bias-variance decomposition

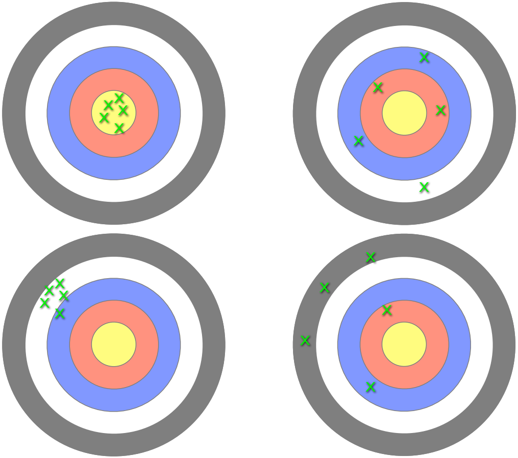


Рис. 1: *Source: Machine Learning – The Art and Science of Algorithms that Make Sense of Data*

Bias-variance decomposition

- Regression case ($x_1, \ldots, x_n$ and $y_i \in \mathbb{R}$)

$$\mathbf{E}[(y - \hat{f}(x))^2] = (\mathbf{Bias}[\hat{f}(x)])^2 + \mathbf{Var}[\hat{f}(x)] + \sigma^2, \text{ under condition}$$

$y = f(x) + \varepsilon$, where $\mathbf{E}(\varepsilon) = 0$ and $\mathbf{Var}(\varepsilon) = \sigma^2$ (not necessary normal distribution)

$$\mathbf{Bias}[\hat{f}(x)] = \mathbf{E}[\hat{f}(x) - f(x)]$$

$$\mathbf{Var}[\hat{f}(x)] = \mathbf{E}[\hat{f}(x)^2] - (\mathbf{E}[\hat{f}(x)])^2$$

Outline

Bagging (BootstrapAggregation)

(Breiman, 1996), ссылка

Let us consider a training set $\mathcal{L} = \{(\mathbf{x}_i, y_i), 1, \dots n\},$ где
$y_i \in \mathbb{R}$ (regression problem) or $y_i \in \{1, \dots k\}$ (classification problem).

Let us generate B training sets $\mathcal{L}^b$ of length n with replacements $\mathcal{L}$.

$$\mathcal{L}^b = \{(\mathbf{x}_i^b, y_i^b), i = 1, 2, \dots, n\}, \text{ где}$$

$b = 1, 2, \dots, B$, and $p_i = 1/n$ is the probability of each pair (x_i, y_i) to be chosen from $\mathcal{L}$.

37 % $\left(1-\frac{1}{n}\right)^n \quad n \to \infty \quad = e^{-1}$ 1/e

Bagging over classification trees

Let us build a tree $\mathcal{T}^b$ for each set $\mathcal{L}^b$.
If $(\mathbf{x}, y) \in \mathcal{L} \setminus \mathcal{L}^b$, then the pair $(\mathbf{x}, y)$ is out-of-bag.
37% of examples for each $\mathcal{L}^b$ do not contribute to the tree training process
For $\mathbf{x}_i \notin \mathcal{L}^b$ we predict its class by applying the tree $\mathcal{T}^b$: $\mathcal{T}^b(\mathbf{x})$.
Let us assume that for $n_i \leq B$ trees, the example $\mathbf{x}_i$ is out-of-bag, then its class distribution vector is as follows:

$$\hat{p}(x_i) = (\hat{p_1}(x_i), \hat{p_2}(x_i), \ldots, \hat{p_K}(x_i))^T$$

OOB-classifier

$$C_{bag}(x_i) = \arg\max_k \hat{p}_k(x_i)$$

Error-rate:

$$PE = \frac{1}{n}\sum_{i=1}^{n}[C_{bag}(\mathbf{x}_i) \neq y_i]$$

Example

Spambase: 57 features for 4601 messages, two classes: spam (1813 messages) or e-mail (2788 messages). Imbalance ratio $1813/4601 = 0{,}394$.

Decision tress: stumps, trees with 4 nodes, 8 nodes and fully-grown trees. Bagging for B form 10 to 200 with the step size 25.

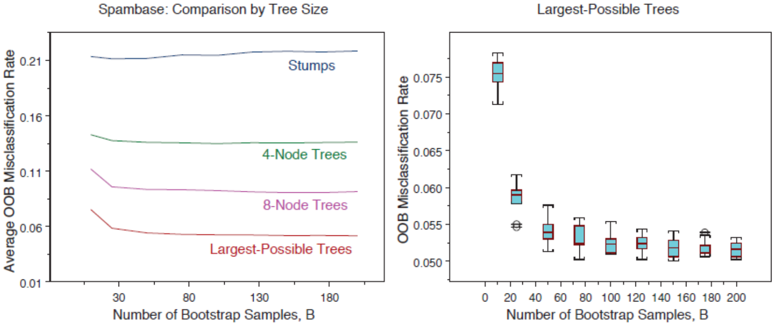


Рис. 2: *Source: Modern Multivariate Statistical Techniques*

Bagging over decision trees

Let us build a decision tree $\mathcal{T}^b$ for each bagging sample $\mathcal{L}^b$.
For $\mathbf{x}$ we obtain the prediction by the tree $\mathcal{T}^b$ as follows: $\hat{\mu}^b(\mathbf{x})$.

OOB-estimate

$$\hat{\mu}_{bag}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}\hat{\mu^b}(x)$$

Errors rate:

$$PE_{bag} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_{bag}(\mathbf{x}_i))^2$$

Outline

Boosting

[Schapire (1990) link], Freund (1995) link]

Weak (or base) classifier correctly classifies examples as $\{+1, -1\}$ more than in 50% cases.
Boosting of the algorithms combines M base classifiers $C_1, C_2, \ldots, C_M$.
For an object $\mathbf{x}$, "boosted" classifier is as follows:

$$C_\alpha(\mathbf{x}) = sign\{f_\alpha(\mathbf{x})\}, \text{где}$$

$f_\alpha(\mathbf{x}) = \sum\limits_{j=1}^{M} \left(\frac{\alpha_j}{\sum_k \alpha_k}\right) C_j(\mathbf{x})$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ is the coefвектор коэффициентов.

Example

$M = 4$

$$C_1(\text{e-mail}) = \begin{cases} +1 & \text{if the message contains the word “money”} \\ -1 & \text{otherwise} \end{cases}$$

$$C_2(\text{e-mail}) = \begin{cases} +1 & \text{if the message contains the word “free”} \\ -1 & \text{otherwise} \end{cases}$$

$$C_3(\text{e-mail}) = \begin{cases} +1 & \text{if the message contains the word “order”} \\ -1 & \text{otherwise} \end{cases}$$

$$C_4(\text{e-mail}) = \begin{cases} +1 \text{ if the message contains the word “credit”} \\ -1 \text{ otherwise} \end{cases}$$

$$f(\text{e-mail}) = 0,2C_1(\text{e-mail}) + 0,1C_2(\text{e-mail}) + 0,4C_3(\text{e-mail}) + 0,3C_4(\text{e-mail})$$

The message contains words “money”, “order”, and “credit”

$$f(\text{e-mail}) = 0,2 - 0,1 + 0,4 + 0,3 = 0,8$$

$sign\{f(\text{e-mail})\} = sign\{0,8\} = +1 \Rightarrow \text{spam}$

AdaBoost

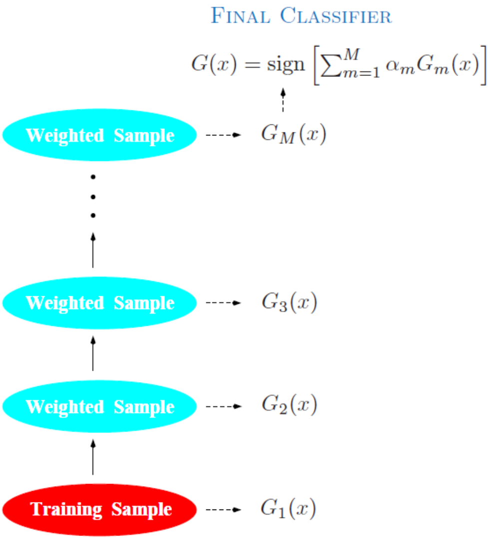


Рис. 3: *Source: Modern Multivariate Statistical Techniques*

AdaBoost.M1

1. Input: $\mathcal{L} = \{(\mathbf{X}_i, Y_i), i = 1, 2, \ldots, n\}$, $Y_i \in \{-1, +1\}$, $i = 1, 2, \ldots, n$, $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$, T = number of iterations.
2. Initialize the weight vector: Set $\mathbf{w}_1 = (w_{11}, \cdots, w_{n1})^\tau$, where $w_{i1} = 1/n$, $i = 1, 2, \ldots, n$.
3. For $t = 1, 2, \ldots, T$:
   - Select a weak classifier $C_{j_t}(\mathbf{x}) \in \{-1, +1\}$ from $\mathcal{C}$, $j_t \in \{1, 2, \ldots, M\}$, and train it on the learning set $\mathcal{L}$, where the ith observation $(\mathbf{X}_i, Y_i)$ has (normalized) weight w_{it}, $i = 1, 2, \ldots, n$.
   - Compute the weighted prediction error:
   $$PE_t = PE(\mathbf{w}_t) = \mathrm{E}_w\{I_{[Y_i \neq C_{j_t}(\mathbf{X}_i)]}\} = \left(\frac{\mathbf{w}_t^\tau}{\mathbf{1}_n^\tau \mathbf{w}_t}\right)\mathbf{e}_t,$$
   where E_w indicates taking expectation with respect to the probability distribution of $\mathbf{w}_t = (w_{1t}, \cdots, w_{nt})^\tau$, and $\mathbf{e}_t$ is an n-vector with ith entry $[\mathbf{e}_t]_i = I_{[Y_i \neq C_{j_t}(\mathbf{X}_i)]}$.
   - Set $\beta_t = \frac{1}{2}\log\left(\frac{1-PE_t}{PE_t}\right)$.
   - Update weights:
   $$w_{i,t+1} = \frac{w_{it}}{W_t}\exp\{2\beta_t I_{[Y_i \neq C_{j_t}(\mathbf{X}_i)]}\}, \quad i = 1, 2, \ldots, n,$$
   where W_t is a normalizing constant needed to ensure that the vector $\mathbf{w}_{t+1} = (w_{1,t+1}, \cdots, w_{n,t+1})^\tau$ represents a true weight distribution over $\mathcal{L}$; that is, $\mathbf{1}_n^\tau \mathbf{w}_{t+1} = 1$.
4. Output: $\mathrm{sign}\{f(\mathbf{x})\}$, where $f(\mathbf{x}) = \sum_{t=1}^T \beta_t C_{j_t}(\mathbf{x}) = \sum_{j=1}^M \alpha_j C_j(\mathbf{x})$, and $\alpha_j = \sum_{t=1}^T \beta_t I_{[j_t = j]}$.

Рис. 4: *Source: Modern Multivariate Statistical Techniques*

AdaBoost.M1

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\mathrm{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \mathrm{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Рис. 5: *Source: Elements of Statistical Learning*

AdaBoost: Example

Данные: Solubility data 5631×71, 2 classes (soluble and insoluble compounds)

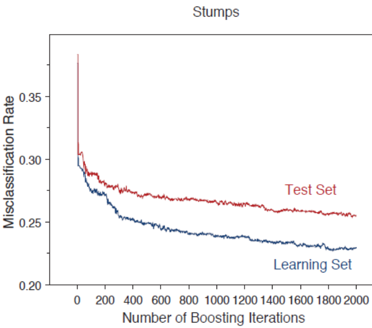


Рис. 6: *Source: Modern Multivariate Statistical Techniques*

Boosting for regression

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{0}$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{1}$$
3. Output the boosted model,
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{2}$$

Рис. 7: *Source: Introduction to Statistical Learning*

Gradient boosting

Greedy Function Approximation: A Gradient Boosting Machine, Friedman (1999-2001)

Idea:

use the loss gradient by the added function during the training phase

Implementation with the regularisation: XGBoost library

Outline

Random Forest

Breiman (2001), статья

Idea:

The average of B i.i.d. random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$. If the variables are i.d. (but not necessarily independent) with positive pair-wise correlation ρ, the variance of their mean:

$$\rho\sigma^2 + \frac{(1-\rho)}{B}\sigma^2.$$

What is happening when B grows?

Random Forest

Breiman (2001), статья

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\rm rf}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\rm rf}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Рис. 8: *Source: Elements of Statistical Learning*

Comparison of boosting, bagging, and random forest

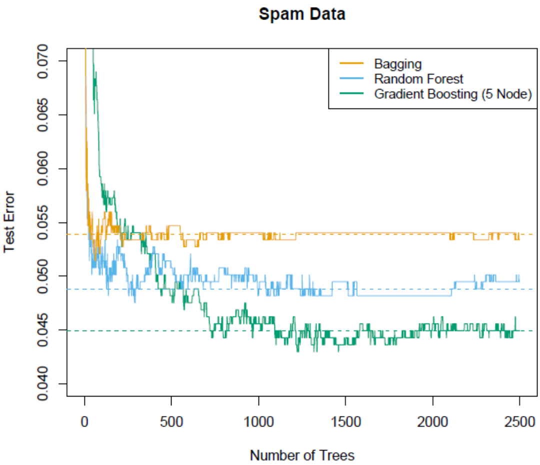


FIGURE 15.1. *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting, 5-node trees were used, and the number of trees were chosen by 10-fold cross-validation (2500 trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of 1536).*

Outline

Classic variant

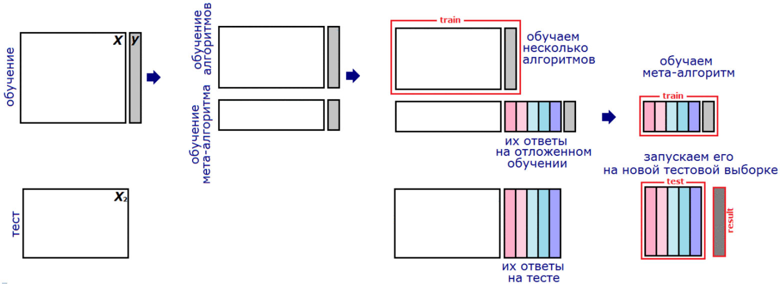


Рис. 10: *Source: Blog of Alexander Diakonov*

Blending
Modification

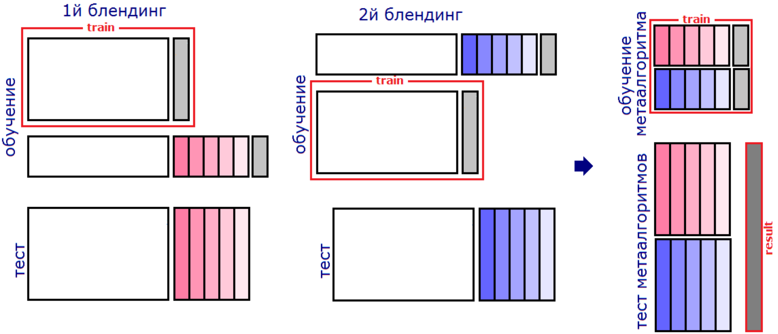


Рис. 11: *Source: Blog Alexander Diakonov*

Stacking or Stacked Generalization

David H. Wolpert: Stacked generalization. Neural Networks 5(2): 241-259 (1992)

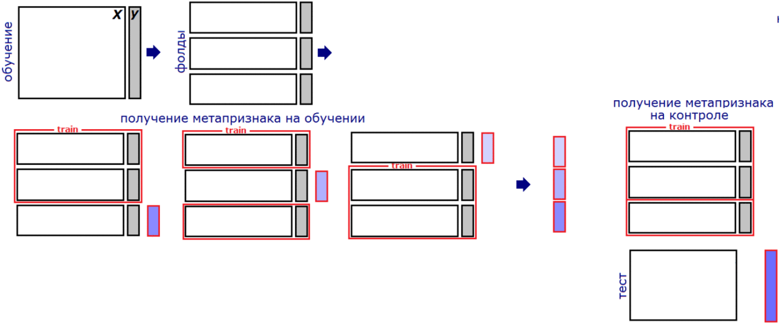


Рис. 12: *Source: Blog Alexander Diakonov*

Stacking or Stacked Generalization

Data source: http://mlbootcamp.ru/

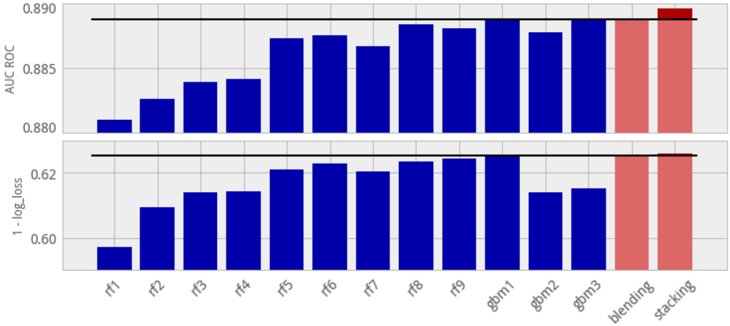


Рис. 13: *Source: Blog Alexander Diakonov*

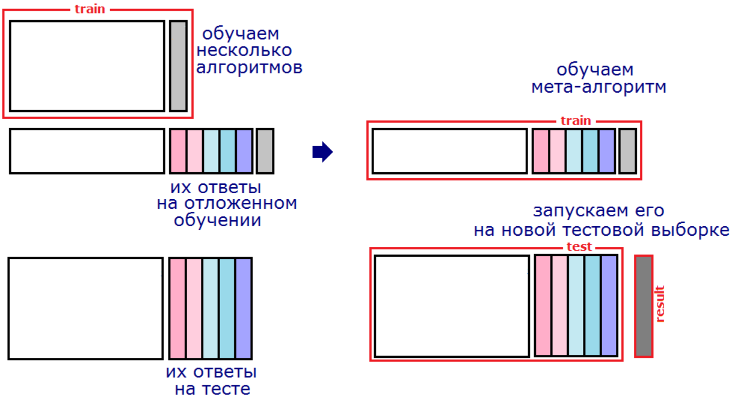


Рис. 14: *Source: Blog Alexander Diakonov*

Stacking or Stacked Generalization

Typical scheme

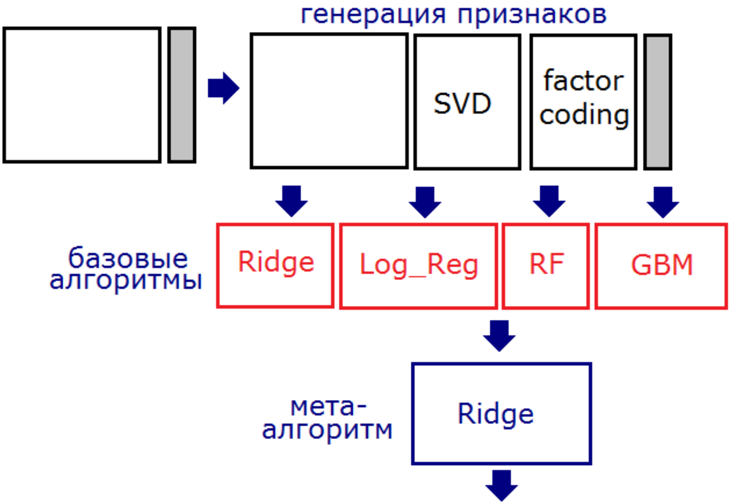


Рис. 15: *Source: Блог Alexander Diakonov*

Yury Kashnitsky, Dmitry I. Ignatov. Can FCA-based Recommender System Suggest a Proper Classifier? ECAI 2014, workshop FCA4AI

Таблица 1: *A sample data set of 10 objects with 4 attributes and 1 binary target class*

G/M	m_1	m_2	m_3	m_4	Label
1	×	×		×	1
2	×			×	1
3		×	×		0
4	×		×	×	1
5	×	×	×		1
6		×	×	×	0
7	×	×	×		1
8			×	×	0
9	×	×	×	×	?
10		×		×	?

Таблица 2: *A classification context*

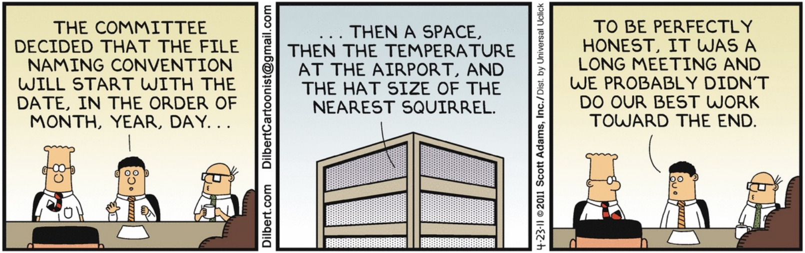
G/C	cl_1	cl_2	cl_3	cl_4
1	×		×	×
2		×	×	
3	×			×
4		×	×	
5	×	×		
6	×	×		×
7		×		×
8		×	×	×

Yury Kashnitsky, Dmitry I. Ignatov. Can FCA-based Recommender System Suggest a Proper Classifier? ECAI 2014, workshop FCA4AI

Таблица 3: *Recommending classifiers for objects from* G_{test}

G_{test}	1^{st} nearest neighbor	2^{nd}	3^{rd}	Neighbors	Classification concept	Recommended classifier
9	4	5	7	$\{4,5,7\}$	$(\{2,4,5,6,7,8\},\{cl_2\})$	cl_2
10	1	6	8	$\{1,6,8\}$	$(\{1,3,6,7,8\},\{cl_4\})$	cl_4

Just for fun или шутки ради



References

- A.J. Izenman, Modern Multivariate Statistical Techniques, Chapter 14

Question and contacts

www.hse.ru/staff/dima

Thank you!

dmitrii.ignatov[at]gmail.com