# Ансамбли моделей

Максим Гончаров
maxgon@microsoft.com
maxim.goncharov@spellabs.ru

# 1. Введение

Под обучением *ансамбля моделей* понимается процедура обучения конечного набора *базовых классификаторов*, результаты прогнозирования которых, затем объединяются и формируют прогноз агрегированного классификатора. Целью создания ансамбля моделей является повышение точности прогноза агрегированного классификатора по сравнению с точностью прогнозирования каждого индивидуального базового классификатора.

Интуитивно понятно, что комбинирование базовых классификаторов даст более точный результат, чем каждый индивидуальный классификатор, если базовые классификаторы с одной стороны достаточно точны, а с другой – если они дают разные результаты, т.е. ошибаются на разных множествах. Проиллюстрируем это следующим искусственным примером:
Пусть у нас есть 25 базовых классификатора, при этом

- вероятность ошибки каждого классификатора 35%
- результаты прогноза классификаторов независимы

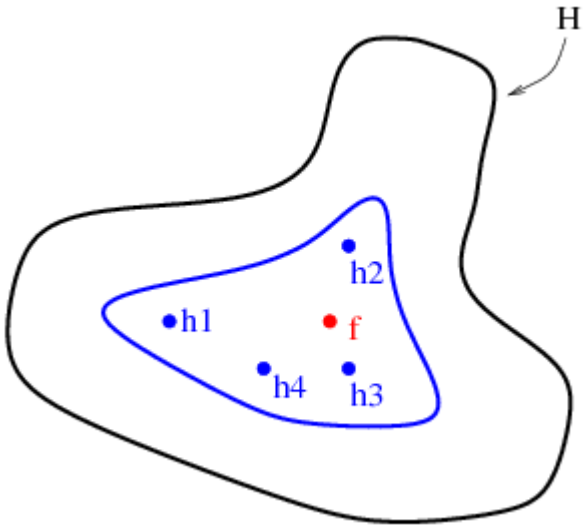
В этом случае, вероятность ошибки объединенного классификатора, прогнозирующего то значение класса, за которое «проголосовало» большинство базовых классификаторов, будет равно:

$$P\left(\text{больше 12 классификаторов ошиблись}\right)=\sum_{i=13}^{25}\binom{25}{i}0,35^{i}\bullet 0,65^{25-i}=0,06$$

Т.е. ошибка ансамбля будет всего 6% по сравнению с 35% ошибки каждого базового классификатора. Рассмотренный пример является нереалистичным, так как на практике результаты прогнозирования базовых классификаторов сильно коррелированы, потому что наибольшая вероятность ошибки любого классификатора происходит на границе классов, а наименьшая – вдали от границ.

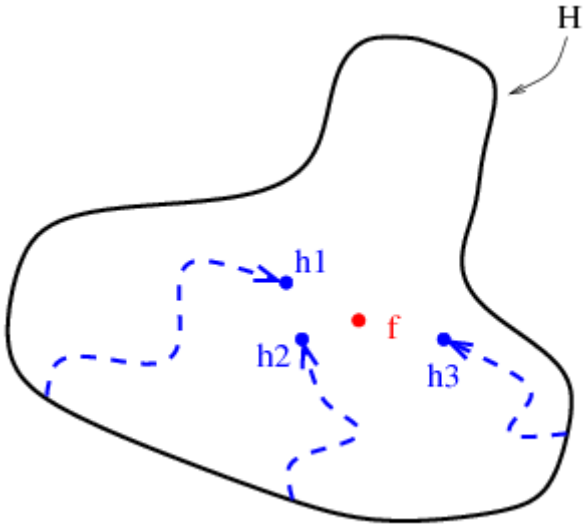
На основании общих рассуждений можно выделить три причины, по которым объединение классификаторов может быть успешным:

1. *Статистическая* причина. Классификационный алгоритм можно рассматривать как процедуру поиска в пространстве гипотез H о распределении данных с целью поиска наилучшей гипотезы. Обучаясь на конечном наборе данных, алгоритм может найти множество различных гипотез одинаково хорошо описывающих обучающую выборку. Строя ансамбль моделей, мы «усредняем» ошибку каждой индивидуальной гипотезы и уменьшаем влияние нестабильностей и случайностей при формировании гипотез.

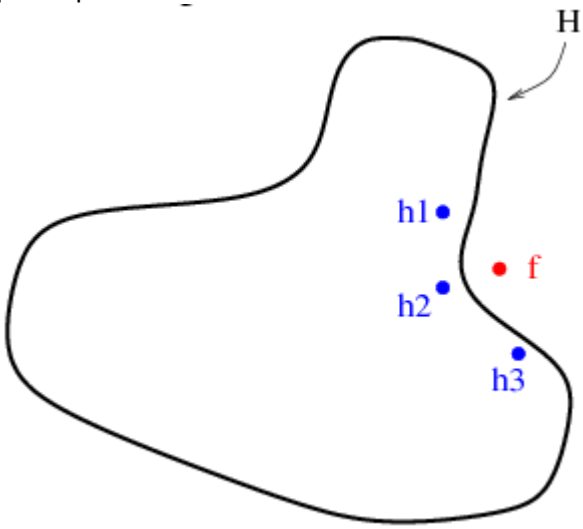



2. *Вычислительная* причина. Большинство обучающих алгоритмов используют методы нахождения экстремума некой целевой функции. Например, нейронные сети используют

методы градиентного спуска для минимизации ошибки прогноза, деревья решений – жадные алгоритмы роста дерева, минимизирующие энтропию данных и т.д. Эти алгоритмы оптимизации могут «застрянуть» в точке локального экстремума. Ансамбли моделей, комбинирующие результаты прогноза базовых классификаторов, обученных на различных подмножествах исходных данных, имеют б**о**льший шанс найти глобальный оптимум, так как ищут его из разных точек исходного множества гипотез.




3. *Репрезентативная* причина. Комбинированная гипотеза может не находиться в множестве возможных гипотез для базовых классификаторов, т.е. строя комбинированную гипотезу, мы расширяем множество возможных гипотез.



## 2. Подходы к формированию ансамблей моделей

Одним из наиболее распространенных подходов к формированию ансамбля моделей является манипулирование обучающим множеством с последующим построением базовых классификаторов на различных его подмножествах. Каждый базовый классификатор использует один и тот же алгоритм, но обучается на различных данных. Затем прогноз ансамбля производится комбинированием результатов прогноза каждого отдельного классификатора при помощи (взвешенного) усреднения для непрерывной целевой переменной или (взвешенного) голосования – для дискретной. Такой подход особенно успешен, когда базовый алгоритм является *нестабильным*, т.е. даёт ощутимо различный результат при небольшом изменении данных в обучающей выборке. Примером нестабильного алгоритма может служить деревья решений, в процессе построения которых небольшие изменения данных могут послужить причиной разбиения узлов дерева по разным атрибутам и границам их значений. Мы рассмотрим два алгоритма, использующих этот подход: *Bagging* и *Boosting*.

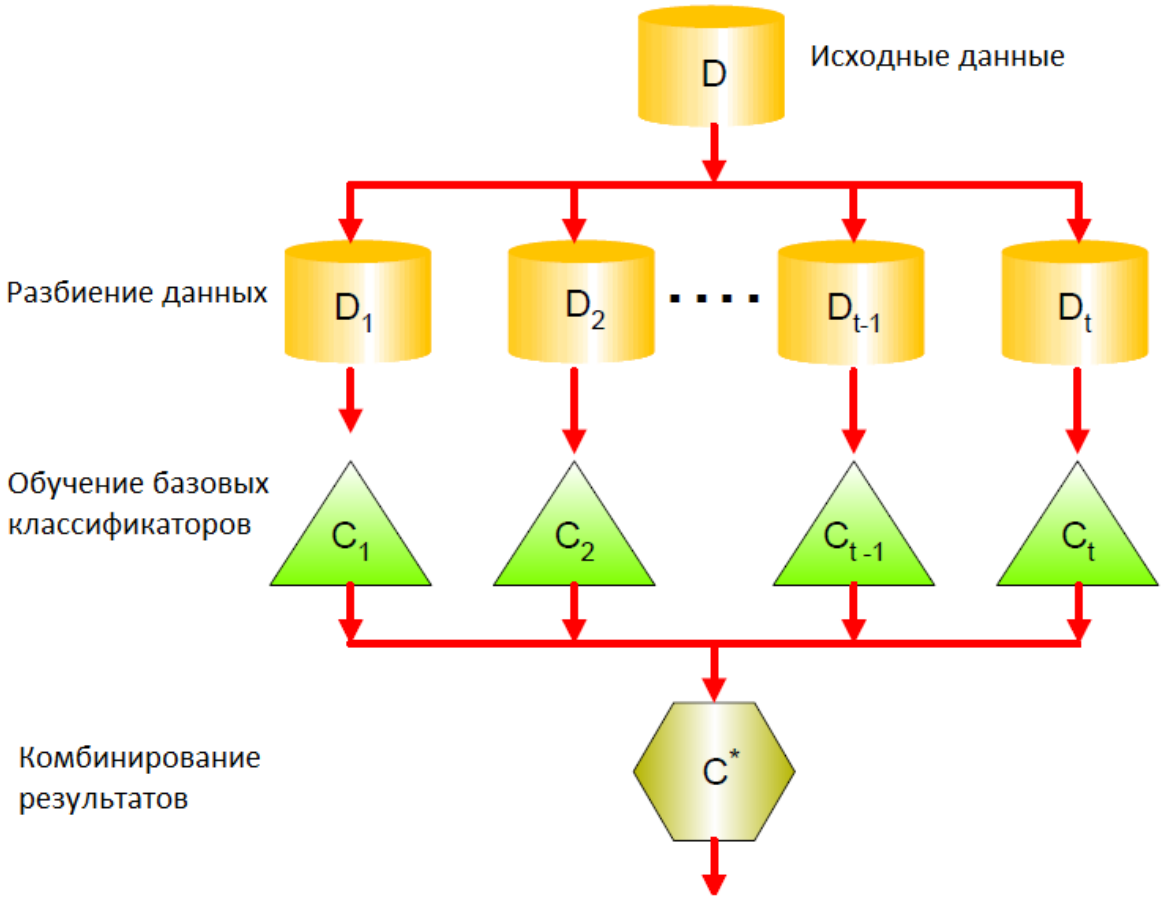
В алгоритме Bagging исходные данные случайно разбиваются на одинаковые по размеру подмножества, каждое из которых используется для обучения одного базового классификатора. Прогноз ансамбля определяется большинством голосов или средним.

Алгоритм Boosting итеративно учится распознавать примеры на границах классов. Каждой записи данных на каждой итерации алгоритма присваивается вес, который соответствует, например, вероятности попадания этой записи в обучающую выборку на следующей итерации или соответствует частоте, с которой эта запись будет «размножена» в обучающей выборке на следующей итерации. Первый базовый классификатор обучается на всех данных с равномерными весами. Затем, на каждой последующей итерации, веса правильно классифицированных примеров уменьшаются, а неправильно классифицированных – увеличиваются. Следующий классификатор, таким образом, будет «уделять больше внимания» неправильно классифицированным примерам, т.е. все больше учиться исправлять ошибки классификатора на прошлой итерации. Прогноз ансамбля представляет собой взвешенное голосование прогнозов базовых классификаторов. Вес, с которым учитывается результат прогноза каждого базового классификатора при голосовании, соответствуют точности его прогнозирования.

Вторым подходом к формированию ансамбля является использование различных алгоритмов в качестве базовых, обучаемых на одинаковых данных. После обучения базовых моделей, исходные данные вместе с результатами прогноза всех базовых моделей используются для обучения мета-алгоритма, который учится распознавать на каких фрагментах данных следует доверять той или иной базовой модели. Такой подход к созданию ансамбля моделей называется *Stacking*.

## 3. Bagging

*Bagging* (**B**ootstrap **Agg**regat**ing**, Leo Breiman) – улучшающее объединение. Исходные данные D, состоящие из N строк, разделяются на t подмножеств $D_1$ ,…, $D_t$ с тем же числом строк в каждом при помощи равномерной случайной выборки с возвратом. Затем, t базовых классификаторов, используя один и тот же алгоритм, обучаются на этих подмножествах. Результаты прогнозирования базовых классификаторов усредняются или выбирается класс на основании большинства голосов.




Каждая запись из D имеет одинаковую вероятность быть выбранной в множество $D_i$, т.е. 1/N, значит, вероятность не быть выбранной равна (1- 1/N). Так как каждое из множеств $D_i$ формируется независимо друг от друга, то вероятность того, что определенная запись из D не попадет ни в какое $D_i$, будет равна $(1- 1/N)^t$, следовательно, вероятность, что запись хотя бы раз будет выбрана, равна $1- (1- 1/N)^t$.

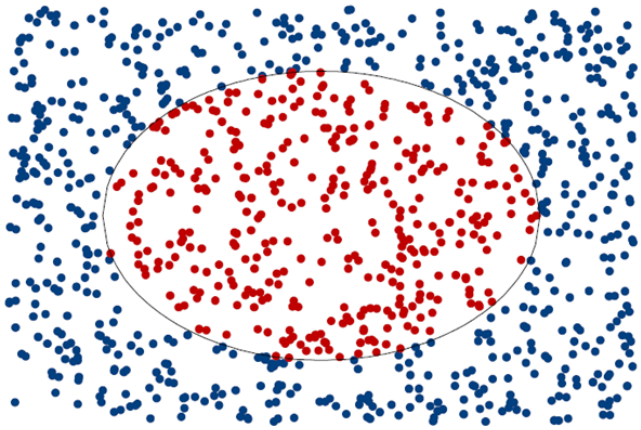
На практике Bagging дает хорошее улучшение точности результатов (5-40% при использовании базового алгоритма CART) по сравнению с индивидуальными базовыми классификаторами в случае, если алгоритм, используемый базовым классификатором достаточно точен, но нестабилен. Улучшение точности прогноза происходит за счет уменьшения разброса нестабильных прогнозов индивидуальных классификаторов.

Преимуществом алгоритма Bagging является простота реализации, а также возможность распараллеливания вычислений по обучению каждого базового классификатора по различным вычислительным узлам. Недостатками являются:

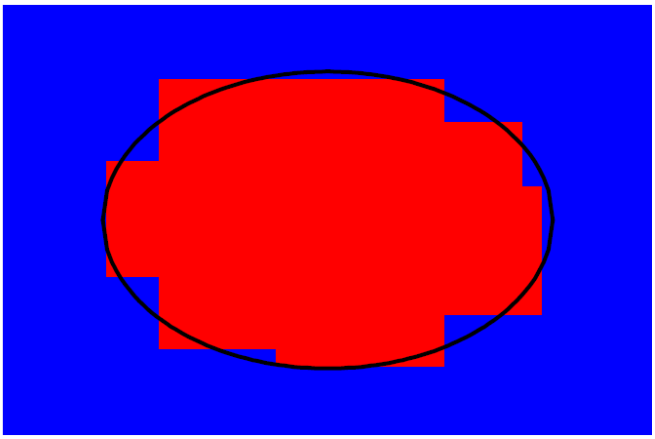
- отсутствие строгого математического обоснования условий улучшения прогноза ансамбля
- недетерминированность результата (обучающие выборки формируются случайно)
- сложность интерпретации результатов по сравнению с индивидуальными моделями

Ниже приведена иллюстрация итеративного улучшения точности прогноза алгоритма Bagging при увеличении числа базовых классификаторов.

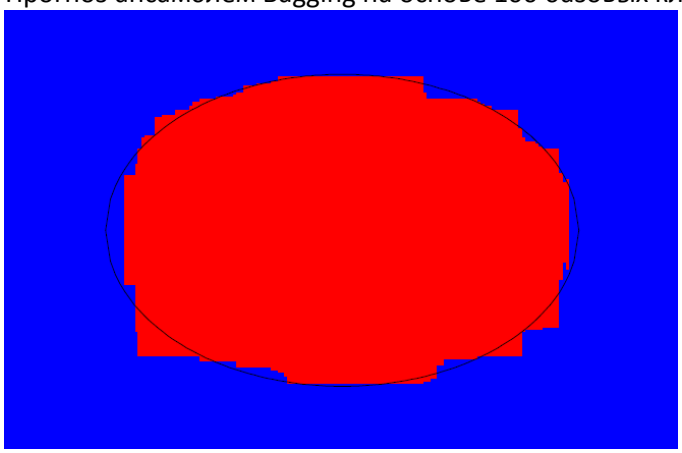
Исходное множество из двух классов:



Прогноз алгоритмом CART:



Прогноз ансамблем Bagging на основе 100 базовых классификаторов CART:



## 3.1 Обоснование работы алгоритма Bagging

Ниже будет приведено некое обоснование улучшения точности прогноза при использовании ансамбля Bagging.

### *3.1.1 Модель*

Пусть X – вектор случайных переменных, описывающий входные атрибуты классификации или регрессии, а Y – выходная (прогнозируемая) переменная. При помощи D мы будем обозначать случайную величину, ставящую в соответствие каждому элементу $\omega \in \Omega$ из множества элементарных событий *базу данных*, т.е. конечную выборку $D(\omega) = \{(X_1(\omega), Y_1(\omega)), ..., (X_N(\omega), Y_N(\omega))\}$ одинаково распределенных и независимых реализаций (X, Y) из некой генеральной совокупности. Мы предполагаем, что D не зависит от (X, Y), т.е., что факт попадания записи в базу данных D не зависит от ее значения (X, Y). Мы будем также предполагать, что все рассматриваемые функции от D являются интегрируемыми.

Пусть задана функция $\varphi(X, D)$, описывающая модель прогноза. Эта модель обучается на базе данных D и ставит в соответствие каждому значению входного вектора $x \in Val(X) = X(\Omega)$ прогнозируемое значение выходной переменной $y \in Val(Y) = Y(\Omega)$.

### *3.1.2 Действительная выходная переменная*

Рассмотрим случай, когда $Val(Y) = \mathbb{R}$. Определим ошибку прогноза некого измеримого по X регрессора $\varphi$ как $R(\varphi) := E\left[(Y-\varphi)^2 | X\right]$. Так как $R(\varphi)$ определяется как условное от X математическое ожидание, то $R(\varphi)$ – случайная величина, измеримая по X, описывающая средний квадрат ошибки прогноза регрессора $\varphi$ в зависимости от входного вектора X.

Байесовская оценка $\hat{\varphi} := E[Y|X]$ является оптимальной с точки зрения минимизации функции ошибки $R(\varphi)$, так как

$$R(\varphi) = E\left[\left((Y-\hat{\varphi})+(\hat{\varphi}-\varphi)\right)^2 | X\right] = \underbrace{E\left[(Y-\hat{\varphi})^2 | X\right]}_{=R(\hat{\varphi})} + \underbrace{E\left[(\varphi-\hat{\varphi})^2 | X\right]}_{\geq 0} + 2E\left[(Y-\hat{\varphi})(\varphi-\hat{\varphi})|X\right] \geq$$

$$\geq R(\hat{\varphi}) + 2(\varphi-\hat{\varphi})\underbrace{E\left[(Y-\hat{\varphi})|X\right]}_{=0} = R(\hat{\varphi}), \forall \varphi$$

Функция ошибки оптимального регрессора $N := R(\hat{\varphi})$ называется *шумом* (noise), так как описывает степень отклонения Y от своего лучшего возможного прогноза, т.е. характеризует разброс в распределении самих данных.

Для любого регрессора $\varphi(X, D)$ определим *регрессор ансамбля* моделей Bagging как $\bar{\varphi} := E_D \varphi(X, D)$, т.е. как средний по всем возможным обучающим базам данных прогноз базового регрессора. Условное среднеквадратическое отклонение регрессора ансамбля $\bar{\varphi}$ от оптимального регрессора $\hat{\varphi}$ будем называть *смещением* (bias), так как эта функция описывает насколько лучший возможный регрессор отличается от усредненного по всем наборам данных регрессора $B(\varphi) := E\left[(\hat{\varphi}-\bar{\varphi})^2 | X\right]$. Условное среднеквадратическое отклонение регрессора ансамбля от базового регрессора мы будем называть *разбросом* (variance), так он описывает дисперсию базового регрессора $V(\varphi) := E\left[(\bar{\varphi}-\varphi)^2 | X\right]$.

Итак, условная среднеквадратическая ошибка прогноза базового регрессора равна:

$$R(\varphi)=E\left[(Y-\varphi)^2|X\right]=E\left[\left((Y-\hat{\varphi})+(\hat{\varphi}-\varphi)\right)^2|X\right]=$$

$$=\underbrace{E\left[(Y-\hat{\varphi})^2|X\right]}_{=R(\hat{\varphi})}+E\left[(\varphi-\hat{\varphi})^2|X\right]+2\underbrace{E\left[(Y-\hat{\varphi})(\varphi-\hat{\varphi})|X\right]}_{=0}=$$

$$=R(\hat{\varphi})+E\left[\left((\varphi-\bar{\varphi})+(\bar{\varphi}-\hat{\varphi})\right)^2|X\right]=R(\hat{\varphi})+E\left[(\varphi-\bar{\varphi})^2|X\right]+E\left[(\bar{\varphi}-\hat{\varphi})^2|X\right]+$$

$$+2E\left[(\varphi-\bar{\varphi})(\bar{\varphi}-\hat{\varphi})|X\right]$$

Функции $\hat{\varphi}=E\left[Y|X\right]$ и $\bar{\varphi}=E_D\varphi(X,D)$ измеримы по X, поэтому

$E\left[(\varphi-\bar{\varphi})(\bar{\varphi}-\hat{\varphi})|X\right]=(\bar{\varphi}-\hat{\varphi})E\left[(\varphi-\bar{\varphi})|X\right]$. Так как (X, Y) и D по условию независимы, то

$E\left[(\varphi-\bar{\varphi})|X\right]=E_{(X,Y,D)}\left[(\varphi-\bar{\varphi})|X\right]=E_{(X,Y)}\underbrace{E_D\left[(\varphi-\bar{\varphi})|X\right]}_{=0}=0$, следовательно

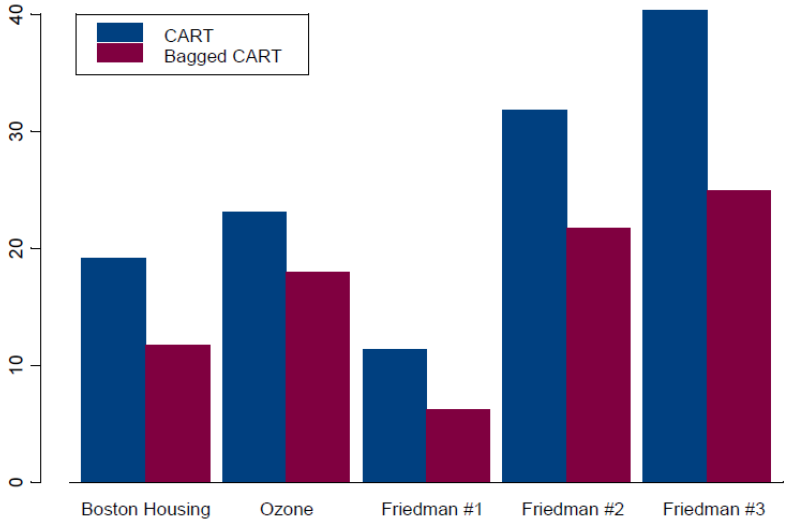
$E\left[(\varphi-\bar{\varphi})(\bar{\varphi}-\hat{\varphi})|X\right]=0$, а значит

$$R(\varphi)=\underbrace{R(\hat{\varphi})}_{=N}+\underbrace{E\left[(\varphi-\bar{\varphi})^2|X\right]}_{=V(\varphi)}+\underbrace{E\left[(\bar{\varphi}-\hat{\varphi})^2|X\right]}_{=B(\varphi)}=N+V(\varphi)+B(\varphi) \qquad \textbf{(1)}$$

Из (1) следует, что среднеквадратическая ошибка прогноза регрессора уменьшается, если он близок к регрессору ансамбля Bagging, т.е. $\varphi\approx\bar{\varphi}$, так как в этом случае уменьшается ошибка, соответствующая разбросу $V(\varphi)$ в правой части равенства (1).

Итак, Bagging уменьшает ошибку разброса, что ведет к увеличению точности прогноза ансамбля.

Среднеквадратическая ошибка регрессии на открытых базах данных:




### *3.1.3 Дискретная выходная переменная*

Рассмотрим задачу классификации, т.е. случай, когда $Val(Y)=\{1,\ldots,m\}$. Условная по X вероятность правильного прогноза для произвольного классификатора $\varphi$ равна:

$P(Y=\varphi|X)=\sum_{i=1}^{m}P(Y=i,\varphi=i|X)$. При известном значении X, значение прогноза $\varphi$ не зависит от Y, так как $\varphi$ измеримая функция от X, т.е. $\varphi$ зависит от Y только через X. Следовательно,

$P\left(Y=\varphi\middle|X\right)=\sum_{i=1}^{m}P\left(Y=i,\varphi=i\middle|X\right)=\sum_{i=1}^{m}P\left(Y=i\middle|X\right)P\left(\varphi=i\middle|X\right)$. Из-за измеримости классификатора $\varphi$ по X следует $P\left(\varphi=i\middle|X\right)=E\left(1_{\{\varphi=i\}}\middle|X\right)=1_{\{\varphi=i\}}$, и, следовательно,

$$P\left(Y=\varphi\middle|X\right)=\sum_{i=1}^{m}1_{\{\varphi=i\}}P\left(Y=i\middle|X\right) \quad \textbf{(2)}$$

Поэтому байесовский классификатор, минимизирующий условную вероятность ошибки, т.е. функцию $R(\varphi):=E\left(1_{\{Y\neq\varphi\}}\middle|X\right)=P\left(Y\neq\varphi\middle|X\right)=1-P\left(Y=\varphi\middle|X\right)$, максимизирует (2) и равен:

$$\hat{\varphi}:=\arg\max\left\{P\left(Y=i\middle|X\right)\middle|i=1,\ldots,m\right\}, \quad \textbf{(3)}$$

т.е. прогнозирует значение класса, условная вероятность которого максимальна.

Аналогично, если базовый классификатор $\varphi$ - функция как от X, так и от D (обучающей выборки), то $P\left(Y=\varphi\middle|X,D\right)=\sum_{i=1}^{m}P\left(Y=i,\varphi=i\middle|X,D\right)=\sum_{i=1}^{m}P\left(Y=i\middle|X,D\right)P\left(\varphi=i\middle|X,D\right)$

$P\left(Y=i\middle|X,D\right)=P\left(Y=i\middle|X\right)$, так как Y и D независимы

$P\left(\varphi=i\middle|X,D\right)=E\left(1_{\{\varphi=i\}}\middle|X,D\right)=1_{\{\varphi=i\}}$, так как $\varphi$ измерима по X и D, следовательно

$$P\left(Y=\varphi\middle|X,D\right)=\sum_{i=1}^{m}1_{\{\varphi=i\}}P\left(Y=i\middle|X\right) \quad \textbf{(4)}$$

Так как X и D независимы, то $P\left(Y=\varphi\middle|X\right)=E_D P\left(Y=\varphi\middle|X,D\right)$ **(5)**

Действительно: функция $\omega\mapsto E_D P\left(Y=\varphi\middle|X(\omega),D\right)=\int P\left(Y=\varphi\middle|X(\omega),d\right)dP(d)$ измерима по X и ее интеграл по любому измеримому по X множеству $\{X\in A\}$ равен $P\left(Y=\varphi,X\in A\right)$:

$$\int_{X^{-1}(A)}E_D P\left(Y=\varphi\middle|X(\omega),D\right)dP(\omega)=\int_{A}E_D P\left(Y=\varphi\middle|x,D\right)dP_X(x)=$$

$$=\int_{A}\left(\int P\left(Y=\varphi\middle|x,d\right)dP_D(d)\right)dP_X(x)=\int_{A}\int P\left(Y=\varphi\middle|x,d\right)dP_D(d)dP_X(x)=$$

$$\underset{\text{независимость D и X}}{=}\int_{A}\int E\left(1_{\{Y=\varphi\}}\middle|x,d\right)dP_{(D,X)}(d,x)=$$

$$=\int_{X^{-1}(A)}\int E\left(1_{\{Y=\varphi\}}\middle|X,D\right)dP\underset{\text{определение } E\left(1_{\{Y=\varphi\}}\middle|X,D\right)}{=}\int_{X^{-1}(A)}\int 1_{\{Y=\varphi\}}dP=P\left(Y=\varphi,X\in A\right)$$

Итак, интегрируя (4) по D, с учетом (5) получаем:

$P\left(Y=\varphi\middle|X\right)=\sum_{i=1}^{m}E_D\left[1_{\{\varphi=i\}}P\left(Y=i\middle|X\right)\right]=\sum_{i=1}^{m}P\left(Y=i\middle|X\right)E_D\left(1_{\{\varphi=i\}}\right)$ или для любой реализации входного вектора x:

$$P\left(Y=\varphi\middle|x\right)=\sum_{i=1}^{m}P\left(Y=i\middle|x\right)E_D\left(1_{\{\varphi=i\}}\right)=\sum_{i=1}^{m}P\left(Y=i\middle|x\right)P\left(\varphi(x,D)=i\right), \quad \textbf{(6)}$$

где $P\left(\varphi\left(x,D\right)=i\right)$ - вероятность того, что классификатор, обученный на множестве D, даст прогноз i на входном векторе x. Если множество всех возможных обучающих баз данных $D_1,\ldots,D_K$ конечно и равновероятно, то $P\left(\varphi\left(x,D\right)=i\right)=\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(x,D_j\right)=i\right\}}$, следовательно, (6) принимает вид:

$$P\left(Y=\varphi\middle|X\right)=\frac{1}{K}\sum_{i=1}^{m}P\left(Y=i\middle|X\right)\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}} \quad \textbf{(7)}$$

Если в качестве классификатора в (2) подставить классификатор ансамбля моделей Bagging $\bar{\varphi}:=\arg\max\left\{\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\middle|i\right\}=\arg\max\left\{\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\middle|i\right\}$, принимающего значение класса, которое спрогнозировало большинство базовых классификаторов, то вероятность правильного прогноза классификатора ансамбля моделей будет равна:

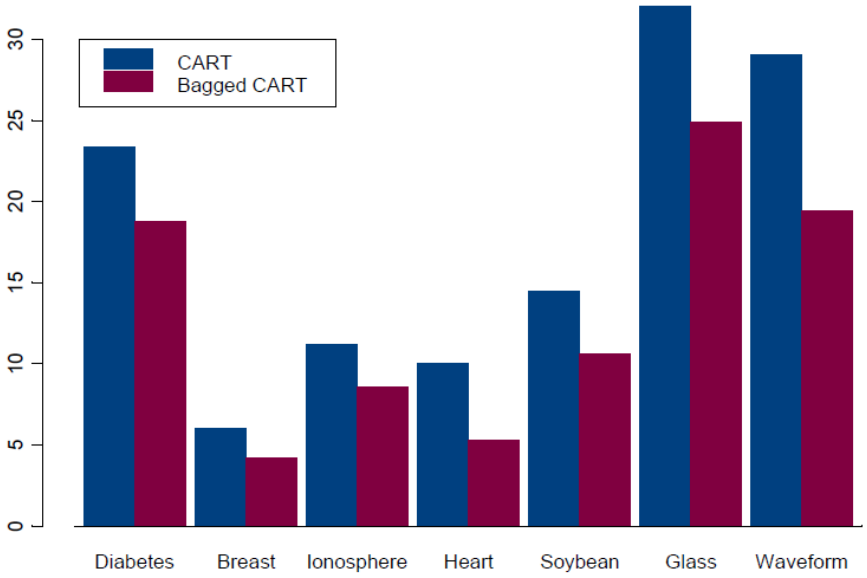
$$P\left(Y=\bar{\varphi}\middle|X\right)=\sum_{i=1}^{m}1_{\{\bar{\varphi}=i\}}P\left(Y=i\middle|X\right)=\sum_{i=1}^{m}1_{\left\{\arg\max\left\{\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\middle|i\right\}=i\right\}}P\left(Y=i\middle|X\right) \quad \textbf{(8)}$$

Если базовые классификаторы достаточно точны и в своем большинстве прогнозируют результат, совпадающий с результатом прогноза оптимального классификатора $\hat{\varphi}$, т.е. если $\bar{\varphi}=\arg\max\left\{\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\middle|i\right\}\approx\arg\max\left\{P\left(Y=i\middle|X\right)\middle|i\right\}=\hat{\varphi}$, то, очевидно, условная вероятность правильного прогноза (8) будет близка максимально возможной. С другой стороны, если базовые классификаторы при этом нестабильны, т.е. если всегда находится хотя бы один классификатор, прогнозирующий результат отличный, от прогноза оптимального классификатора $\forall x\exists i_0:\varphi\left(x,D_{i_0}\right)\neq\hat{\varphi}\left(x\right)=\arg\max\left\{P\left(Y=i\middle|X\right)\middle|i=1,\ldots,m\right\}$, то среднее число «правильно» проголосовавших базовых классификаторов будет всегда строго меньше единицы $\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=\arg\max\left\{P\left(Y=i\middle|X\right)\middle|i\right\}\right\}}<1$, и поэтому, для условной вероятности правильного прогноза базового классификатора согласно (7) будет справедливо

$$P\left(Y=\varphi\middle|X\right)=\sum_{i=1}^{m}P\left(Y=i\middle|X\right)\left(\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\right)=\sum_{i=1,i\neq i_0}^{m}P\left(Y=i\middle|X\right)\left(\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i\right\}}\right)+$$
$$+\underbrace{P\left(Y=i_0\middle|X\right)}_{<\max\left\{P\left(Y=i\middle|X\right)\middle|i\right\}}\underbrace{\left(\frac{1}{K}\sum_{j=1}^{K}1_{\left\{\varphi\left(X,D_j\right)=i_0\right\}}\right)}_{>0}<\max\left\{P\left(Y=i\middle|X\right)\middle|i\right\}=P\left(Y=\hat{\varphi}\middle|X\right)\approx P\left(Y=\bar{\varphi}\middle|X\right) \quad \textbf{(9)}$$

Таким образом, Bagging улучшает точность классификации, если базовые модели достаточно точны, чтобы в большинстве давать оптимальные результаты, но недостаточно точны, чтобы оптимальные результаты были выбраны ими единогласно.

Ошибки классификации на открытых базах данных:

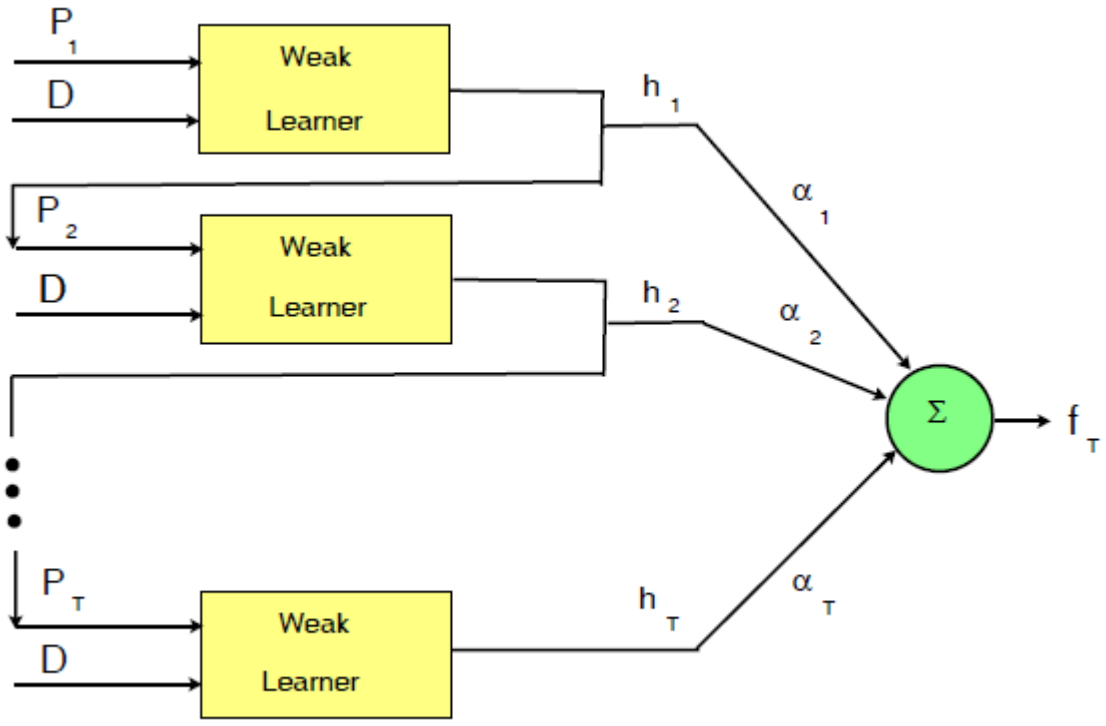

CART
Bagged CART
0
5
10
15
20
25
30
Diabetes
Breast
Ionosphere
Heart
Soybean
Glass
Waveform

## 4. Boosting

В алгоритме Boosting (Yoav Freund, Robert Schapire, Jerry Friedman) базовые классификаторы обучаются последовательно, а не параллельно, как в алгоритме Bagging. Набор данных, на которых обучается каждый последующий базовый алгоритм в Boosting, зависит от точности прогнозирования предыдущего базового классификатора. Важным понятием в Boosting является *вес* строки данных, который обновляется после каждой итерации (выполнения обучения базового классификатора). Значение веса строки описывает ее важность для обучения следующего базового классификатора и основывается на величине ошибки прогноза предыдущего классификатора на этой строке. Вес строки можно интерпретировать как вероятность ее выбора для обучения следующего классификатора или как число, пропорциональное относительной доли этой строки в следующей обучающей выборке. Второй подход представляется предпочтительней, так как является детерминистическим (мы «размножаем» строки для обучения на следующей итерации пропорционально их весу).

Различные модификации алгоритма Boosting имеют одинаковую структуру:

1. Присвоить всем N строкам обучающей выборки одинаковые веса 1/N.
2. В цикле от m=1 до M:
   2.1. Обучить базовый классификатор $f_m$ на данных, распределение строк в которых соответствует весам этих строк.
   2.2. Вычислить взвешенную ошибку прогноза $f_m$.
   2.3. Пересчитать веса каждой строки в соответствии с ошибкой прогноза на этой строке: для правильно классифицированных строк вес уменьшается, для неправильных – увеличивается.
3. Прогноз ансамбля вычисляется по (взвешенному) среднему или (взвешенному) большинству прогнозов базовых классификаторов. В случае использования весов, их значения больше у более точных базовых классификаторов.




Непосредственно из описания шаблона алгоритма Boosting видны его недостатки:

1. Так как на каждой итерации алгоритм уделяет все большее внимание неправильно классифицированным строкам, то в случае зашумления данных, алгоритм на больших

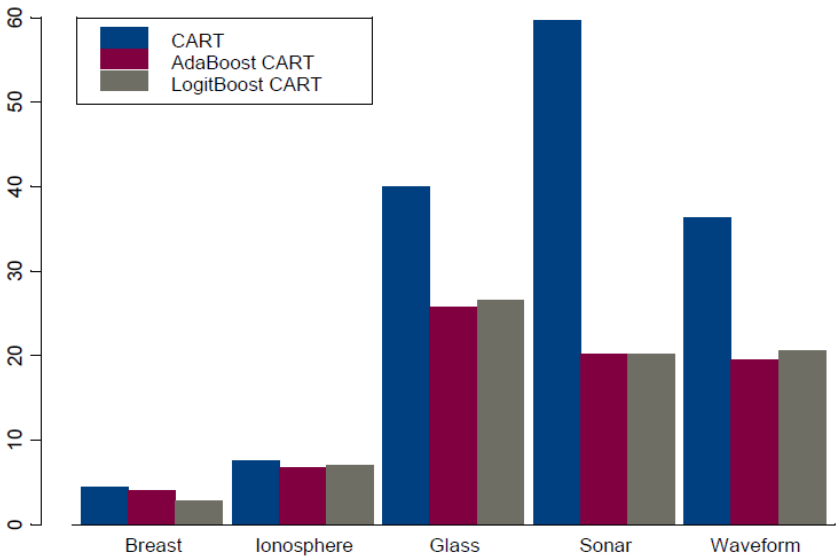
итерациях будет полностью сконцентрирован на попытках получить прогноз на имеющихся ошибочных записях, а не на выявлении объективных закономерностей.
2. Последовательность выполнения обучения базовых классификаторов не позволяет использовать распараллеливание вычислительных мощностей.
3. Результаты классификации ансамбля трудны для интерпретации.

Однако алгоритм Boosting обладает существенными преимуществами по сравнению с другими алгоритмами, работающими с ансамблями моделей:
1. Алгоритм Boosting основывается на вероятностной модели и является аналогом хорошо исследованной логистической регрессии.
2. Каждая итерация любой модификации алгоритма Boosting представляет собой шаг по минимизации функции взвешенного штрафа ошибки или шаг по максимизации функции условного ожидаемого правдоподобия данных. Экстремумы этих функций являются истинными значениями классификационной функции модели. Т.е. алгоритм Boosting имеет статистическое обоснование.
3. Ошибка классификационной функции ансамбля на обучающей выборке при достаточно слабых условиях на точность базовых классификаторов стремится к нулю при увеличении числа итераций.
4. Boosting уменьшает не только ошибку разброса, но и ошибку смещения в терминологии равенства (1).

Все это делает Boosting наиболее теоретически исследованным и обоснованным методом из существующих. Дополнительные факты:

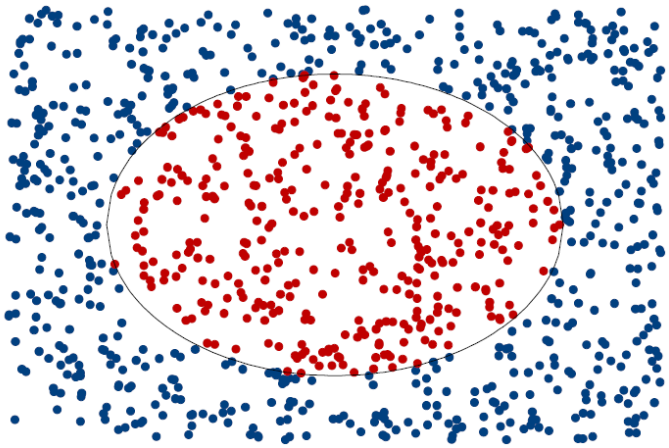
- Leo Breiman назвал в 1996 году алгоритм Boosting с деревьями решений в качестве базового алгоритма лучшим «коробочным» классификационным алгоритмом.
- Boosting с базовым алгоритмом Naïve Bayes оказался самым точным в конкурсе классификационных алгоритмов KDD Cup, 1997 по всем наборам тестовых данных.
- «Boosted naïve Bayes is a scalable, interpretable classifier», Ridgeway, et al 1998.

Ошибки классификации двух модификаций алгоритма Boosting на открытых базах данных:

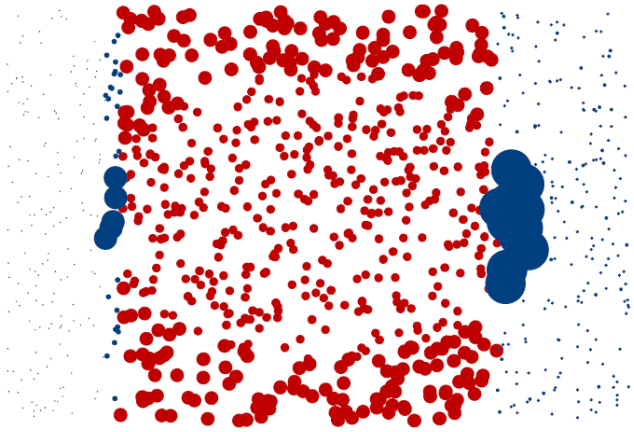



Благодаря тому, что Boosting на каждой итерации концентрируется на задаче классификации «трудных» примеров, то при увеличении числа итераций данные с большими весами находятся на границе классов. Проиллюстрируем результаты работы Boosting в зависимости от числа итераций.

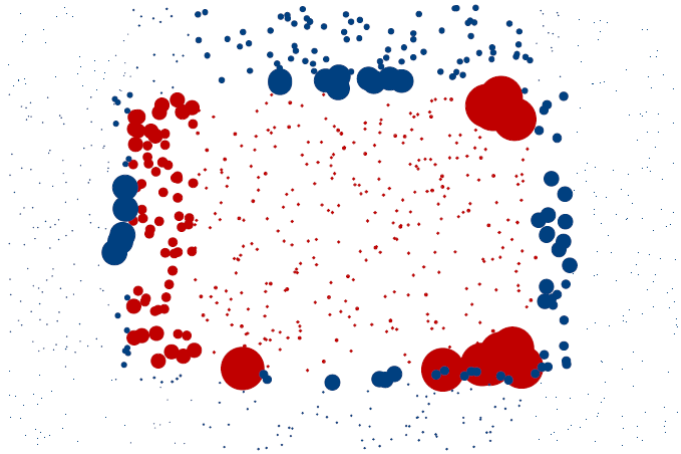
Исходные данные для двух классов:



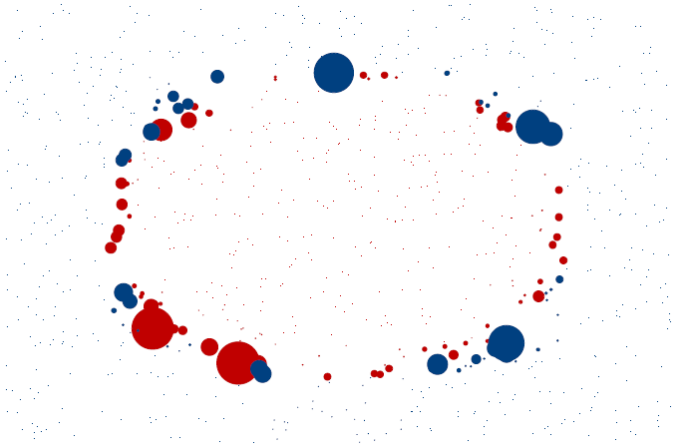
Одна итерация CART:



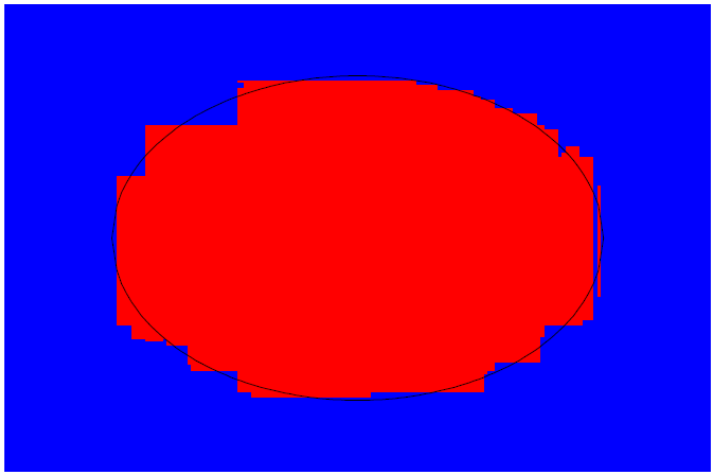
Прогноз CART после трех итераций:



Прогноз CART после 20 итераций:

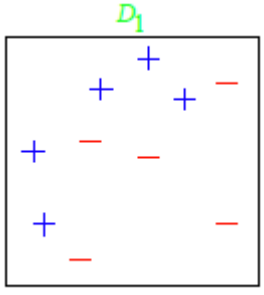


Прогноз ансамбля Boosting после 100 итераций:

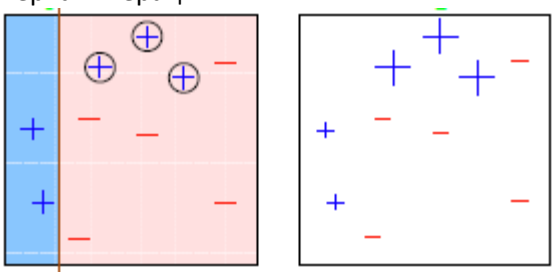


Для иллюстрации работы  алгоритма Boosting рассмотрим следующий «игрушечный» пример. Пусть на плоскости мы хотим построить классификационную функцию, разделяющую + и – при помощи базовых классификаторов, представляющих собой горизонтальные или вертикальные отсечения. Тогда итерации Boosting будут выглядеть следующим образом:

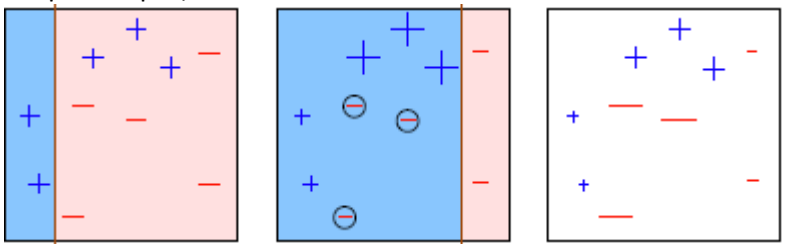
Исходные данные:



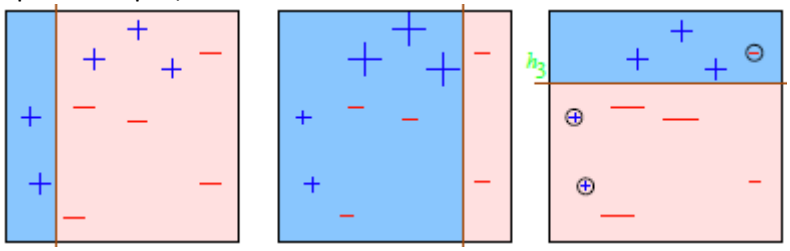

Первая итерация:



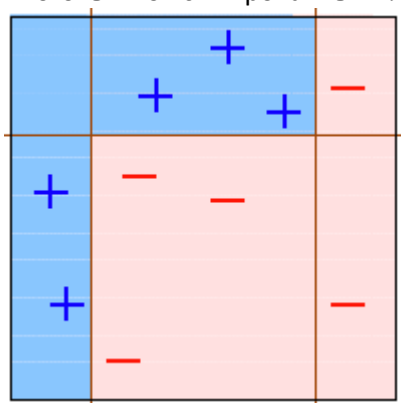
Вторая итерация:



Третья итерация:




Итоговый комбинированный классификатор:



## 4.1 Обоснование работы алгоритма Boosting

В этом разделе будет приведено обоснование применимости алгоритма Boosting для увеличения точности прогнозирования. В качестве модели данных будет рассмотрена модификация бинарной логистической регрессии.

### *4.1.1 Модель данных*

Пусть X – случайный вектор входных переменных, а Y – выходная (прогнозируемая) дискретная случайная переменная, принимающая значение -1 или 1. Мы предполагаем, что вектор данных $(X,Y)$ имеет плотность совместного распределения $p_{X,Y}(x,y)$ относительно меры $\mu \otimes N$, где $\mu$ – некая $\sigma$-конечная мера на $\sigma$-алгебре области значений X (например, произведение лебеговских мер для непрерывных компонент и счетных мер для дискретных компонент вектора X), а N – счетная мера на $\sigma$-алгебре области значений Y: $\wp(\{-1,1\})$. Т.е.

$$P(X \in A, Y = i) = \int_A \int_{\{i\}} p_{X,Y}(x,y)\, dN(y)\, d\mu(x) = \int_A p_{X,Y}(i,y) \underbrace{N(\{i\})}_{=1} d\mu(x) = \int_A p_{X,Y}(i,y)\, d\mu(x)$$

Обозначим маржинальные плотности X и Y:

$$p_X(x) = \int p_{X,Y}(x,y)dN(y) = p_{X,Y}(x,-1) + p_{X,Y}(x,1)$$

$$p_Y(y) = \int p_{X,Y}(x,y)d\mu(x)$$

Обозначим далее условную плотность вероятности Y по X: $p_{Y|X}(y \mid x) = \frac{p_{X,Y}(x,y)}{p_X(x)}$. Т.е.

$$P(Y = i|X) = \int_{\{i\}} p_{Y|X}(y \mid X)\, dN(y) = p_{Y|X}(i \mid X)$$

Рассмотрим семейство $\Im$ плотностей $p^F_{X,Y}(x,y)$ совместного распределения $(X,Y)$ относительно меры $\mu \otimes N$. Это семейство параметризовано действительной функцией $F : X(\Omega) \to \mathbb{R}$, определенной на области значений X. Параметризация плотности распределения имеет следующий вид:

$$p^F_{X,Y}(x,y) = p^F_{Y|X}(y \mid x)\, p_X(x)$$
$$p^F_{Y|X}(1 \mid x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \qquad \textbf{(10)}$$

Равенство (10) описывает модель, являющуюся вариантом логистической регрессии, так как из (10) непосредственно следует $\operatorname{logit} P(Y = 1|X) = \log \frac{P(Y=1|X)}{P(Y=-1|X)} = 2F(X)$.

Мы предполагаем, что истинное значение плотности распределения (X,Y) принадлежит семейству $\Im$, т.е. найдется функция $\hat{F} : X(\Omega) \to \mathbb{R}$, такая, что

$$p_{X,Y}(x,y) = p^{\hat{F}}_{X,Y}(x,y) = p^{\hat{F}}_{Y|X}(y \mid x)\, p_X(x)$$
$$p_{Y|X}(1 \mid x) = p^{\hat{F}}_{Y|X}(1 \mid x) = \frac{e^{\hat{F}(x)}}{e^{\hat{F}(x)} + e^{-\hat{F}(x)}} \qquad \textbf{(11)}$$

### *4.1.2 Вспомогательные результаты*

Получим некоторые вспомогательные результаты, которые потребуются в дальнейшем.

Получим выражение для условного по X математического ожидания логарифма правдоподобия функции-параметра F (Conditional Expected Log-Likelihood)

**Утверждение 1**:

$$E\left[\log p^F_{X,Y}(X,Y)|X\right] = F(X)\left(1 + E(Y|X)\right) - \log\left(1 + e^{2F(X)}\right) + \log p_X(X) =$$
$$= 2F(X)P(Y = 1|X) - \log\left(1 + e^{2F(X)}\right) + \log p_X(X)$$

**Доказательство**:

Логарифм условной плотности распределения Y по X из семейства $\Im$ имеет вид:

$$\log p^F_{Y|X}(y\,|\,x) = \log\left[\, p^F_{Y|X}(1\,|\,x)^{\frac{1+y}{2}}\; p^F_{Y|X}(-1\,|\,x)^{\frac{1-y}{2}} \right] =$$

$$= \frac{1+y}{2}\log p^F_{Y|X}(1\,|\,x) + \frac{1-y}{2}\log p^F_{Y|X}(-1\,|\,x) =$$

$$= \frac{1+y}{2}\log\frac{e^{F(x)}}{e^{F(x)}+e^{-F(x)}} + \frac{1-y}{2}\log\frac{e^{-F(x)}}{e^{F(x)}+e^{-F(x)}} =$$

$$= \frac{1+y}{2}\left(F(x) - \log\left(e^{F(x)}+e^{-F(x)}\right)\right) + \frac{1-y}{2}\left(-F(x) - \log\left(e^{F(x)}+e^{-F(x)}\right)\right) =$$

$$= yF(x) - \log\left(e^{F(x)}+e^{-F(x)}\right) = yF(x) - \log e^{-F(x)}\left(1+e^{2F(x)}\right) =$$

$$= yF(x) + F(x) - \log e^{-F(x)}\left(1+e^{2F(x)}\right) = (1+y)F(x) - \log\left(1+e^{2F(x)}\right)$$

Итак,

$$\log p^F_{Y|X}(y\,|\,x) = (1+y)F(x) - \log\left(1+e^{2F(x)}\right) \qquad \textbf{(12)}$$

Возьмем условное по X математическое ожидание $\log p^F_{X,Y}(X,Y)$, используя (12):

$$E\left[\log p^F_{X,Y}(X,Y)\middle|X\right] \underset{(10)}{=} E\left[\log p^F_{Y|X}(Y\,|\,X)\middle|X\right] + E\left[\log p_X(X)\middle|X\right] =$$

$$\underset{(12)}{=} E\left[(1+Y)F(X) - \log\left(1+e^{2F(X)}\right)\middle|X\right] + \log p_X(X) =$$

$$= F(X)E(1+Y\,|\,X) - \log\left(1+e^{2F(X)}\right) + \log p_X(X) =$$

$$= F(X)\left(1+E(Y\,|\,X)\right) - \log\left(1+e^{2F(X)}\right) + \log p_X(X)$$

То есть

$$E\left[\log p^F_{X,Y}(X,Y)\middle|X\right] = F(X)\left(1+E(Y\,|\,X)\right) - \log\left(1+e^{2F(X)}\right) + \log p_X(X) \qquad \textbf{(13)}$$

Далее, так как

$$E(Y\,|\,X) = E\left(1_{\{Y=1\}} - 1_{\{Y=-1\}}\,|\,X\right) = P(Y=1\,|\,X) - P(Y=-1\,|\,X) =$$

$$= P(Y=1\,|\,X) - \left(1 - P(Y=-1\,|\,X)\right) = 2P(Y=1\,|\,X) - 1$$

то $1+E(Y\,|\,X) = 2P(Y=1\,|\,X)$ и (13) принимает вид:

$$E\left[\log p^F_{X,Y}(X,Y)\middle|X\right] = 2P(Y=1\,|\,X)F(X) - \log\left(1+e^{2F(X)}\right) + \log p_X(X) \qquad \textbf{(14)}$$

□

Далее, покажем, что Conditional Expected Log-Likelihood принимает максимальное значение на истинном значении функции-параметра $\hat{F}$.

**Утверждение 2**:

$$\hat{F} = \frac{1}{2}\log\frac{P(Y=1\,|\,X)}{P(Y=-1\,|\,X)} = \arg\max\left\{E\left[\log p^F_{X,Y}(X,Y)\middle|X\right]\middle|F\right\}$$

**Доказательство**:

Рассмотрим функцию $H : \mathbb{R}\times(0,1) \to \mathbb{R}$ $H(u,v) := 2uv - \log\left(1+e^{2u}\right)$

$$\frac{\partial H(u,v)}{\partial u} = 2v - \frac{2e^{2u}}{1+e^{2u}} = 0 \Leftrightarrow v\left(1+e^{2u}\right) = e^{2u} \Leftrightarrow$$

$$\Leftrightarrow e^{2u}(1-v) = v \Leftrightarrow e^{2u} = \frac{v}{1-v} \Leftrightarrow u = \frac{1}{2}\log\frac{v}{1-v}$$

$$\frac{\partial^2 H(u,v)}{\partial u^2} = \frac{\partial H\left(2v - \frac{2e^{2u}}{1+e^{2u}}\right)}{\partial u} = -\frac{4e^{2u}\left(1+e^{2u}\right) - 2e^{2u}2e^{2u}}{\left(1+e^{2u}\right)^2} = -\frac{4e^{2u}}{\left(1+e^{2u}\right)^2} < 0, \forall v, u$$

Таким образом, $u(v) := \frac{1}{2}\log\frac{v}{1-v}$ является единственной точкой глобального максимума $u \mapsto H(u,v)$ для всех v, и, следовательно,

$$2F(X)P(Y=1|X) - \log\left(1+e^{2F(X)}\right) = H\left(\underbrace{F(X)}_{u}, \underbrace{P(Y=1|X)}_{v}\right) \leq$$

$$\leq H\left(\underbrace{u\left(P(Y=1|X)\right)}_{u(v)}, \underbrace{P(Y=1|X)}_{v}\right) = H\left(\underbrace{\frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)}}_{=:\bar{F}(X)}, P(Y=1|X)\right) =$$

$$= 2\bar{F}(X)P(Y=1|X) - \log\left(1+e^{2\bar{F}(X)}\right), \forall F$$

Т.е.

$$\bar{F}(X) = \frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)} =$$

$$= \arg\max\left\{2F(X)P(Y=1|X) - \log\left(1+e^{2F(X)}\right)|F\right\} =$$

$$= \arg\max\left\{\underbrace{2F(X)P(Y=1|X) - \log\left(1+e^{2F(X)}\right) + \log p_X(X)}_{=E\left[\log p^F_{X,Y}(X,Y)|X\right](14)}|F\right\} = \quad \textbf{(15)}$$

$$= \arg\max\left\{E\left[\log p^F_{X,Y}(X,Y)|X\right]|F\right\}$$

С другой стороны, из (11) следует

$$p^{\hat{F}}_{Y|X}(1|X) = p_{Y|X}(1|X) = P(Y=1|X) = \frac{e^{\hat{F}}}{e^{\hat{F}} + e^{-\hat{F}}} \Rightarrow \hat{F} = \frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)} \quad \textbf{(16)}$$

Из (15) и (16) следует, таким образом, что

$$\hat{F} = \frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)} = \arg\max\left\{E\left[\log p^F_{X,Y}(X,Y)|X\right]|F\right\}$$

□

Из утверждения 2 следует, что стратегией поиска истинной функции-параметра модели $\hat{F}$ может служить поиск максимума по F выражения (13) или (14). Эта стратегия реализована в алгоритме LogitBoost, каждая итерация которого представляет собой шаг минимизации (13) методом Ньютона.

**Утверждение 3**: Истинное значение функции-параметра $\hat{F}$ минимизирует функционалы $F \mapsto E\left[e^{-YF(X)}|X\right]$ и $F \mapsto E\left[e^{-YF(X)}\right]$, т.е.

$$\hat{F} = \arg\min\left\{E\left[e^{-YF(X)}|X\right]|F : X(\Omega) \to \mathbb{R}\right\} = \arg\min\left\{E\left[e^{-YF(X)}\right]|F : X(\Omega) \to \mathbb{R}\right\}$$

**Доказательство**:

Рассмотрим функцию

$$H:\mathbb{R}\times(0,1)\to\mathbb{R}\quad H(u,v):=e^{-u}v+e^{u}(1-v)$$

$$\frac{\partial H(u,v)}{\partial u}=-e^{-u}v+e^{u}(1-v)=0\Leftrightarrow u=\frac{1}{2}\log\frac{v}{1-u}$$

$$\frac{\partial^2 H(u,v)}{\partial u^2}=e^{-u}v+e^{u}(1-v)>0,\forall u,\forall v\in(0,1)$$

Таким образом, $u(v):=\frac{1}{2}\log\frac{v}{1-v}$ является единственной точкой глобального минимума $u\mapsto H(u,v)$ для всех v, и, следовательно,

$$E\left[e^{YF(X)}\middle|X\right]=E\left[e^{-F(X)}1_{\{Y=1\}}+e^{F(X)}1_{\{Y=-1\}}\middle|X\right]=$$

$$=e^{-F(X)}E\left[1_{\{Y=1\}}\middle|X\right]+e^{F(X)}E\left[1_{\{Y=-1\}}\middle|X\right]=$$

$$=e^{-F(X)}P(Y=1|X)+e^{F(X)}P(Y=-1|X)=$$

$$=H\left(\underset{v}{e^{-F(X)}},\underbrace{P(Y=1|X)}_{v}\right)\geq H\left(\underbrace{\frac{1}{2}\log\frac{P(Y=1|X)}{P(Y=-1|X)}}_{u(v)=\hat{F}(X)},\underbrace{P(Y=1|X)}_{v}\right)=$$

$$=H\left(\hat{F}(X),P(Y=1|X)\right)=E\left[e^{y\hat{F}(X)}\middle|X\right]$$

Итак, $E\left[e^{YF(X)}\middle|X\right]\geq E\left[e^{Y\hat{F}(X)}\middle|X\right],\forall F:X(\Omega)\to\mathbb{R}$ **(17)**

Интегрируя (17), получаем $E\left[e^{YF(X)}\right]\geq E\left[e^{Y\hat{F}(X)}\right],\forall F:X(\Omega)\to\mathbb{R}$ **(18)**

□

Из утверждения 3 следует, что минимизация $F\mapsto E\left[e^{-YF(X)}\middle|X\right]$ или $F\mapsto E\left[e^{-YF(X)}\right]$ может являться стратегией поиска истинной функции-параметра $\hat{F}$ . На этой стратегии основаны алгоритмы Discrete AdaBoost, Real AdaBoost, Gentle AdaBoost и другие.

**Утверждение 4**: Квадратичная аппроксимация $t\mapsto E\left[e^{-Y(F(X)+t)}\middle|X\right]$ имеет вид:

$$E\left[e^{-Y(F(X)+t)}\middle|X\right]\approx E\left[e^{-YF(X)}\left(1-ty+\frac{1}{2}t^2\right)\middle|X\right]$$

**Доказательство**:

Разложим функцию $t\mapsto E\left[e^{-Y(F(X)+t)}\middle|X\right]$ в ряд Тейлора второго порядка в точке t=0:

$$E\left[e^{-Y(F(X)+t)}\middle|X\right]=E\left[e^{-F(X)}1_{\{Y=1\}}+e^{F(X)}1_{\{Y=-1\}}\middle|X\right]=$$

$$=e^{-F(X)-t}P\left(Y=1\middle|X\right)+e^{F(X)+t}P\left(Y=-1\middle|X\right)\approx\left\langle \textit{разложение Тейлора } 2\text{-ого порядка}\right\rangle$$

$$\approx e^{-F(X)}P\left(Y=1\middle|X\right)-e^{-F(X)}P\left(Y=1\middle|X\right)t+\frac{1}{2}e^{-F(X)}P\left(Y=1\middle|X\right)t^2+$$

$$+e^{F(X)}P\left(Y=1\middle|X\right)+e^{F(X)}P\left(Y=1\middle|X\right)t+\frac{1}{2}e^{F(X)}P\left(Y=1\middle|X\right)t^2= \qquad \textbf{(19)}$$

$$=E\left[e^{-F(X)}\left(1-t+\frac{1}{2}t^2\right)1_{\{Y=1\}}+e^{F(X)}\left(1+t+\frac{1}{2}t^2\right)1_{\{Y=-1\}}\middle|X\right]=$$

$$=E\left[e^{-yF(X)}\left(1-Yt+\frac{1}{2}t^2\right)1_{\{Y=1\}}+e^{-yF(X)}\left(1-Yt+\frac{1}{2}t^2\right)1_{\{Y=-1\}}\middle|X\right]=$$

$$=E\left[e^{-yF(X)}\left(1-Yt+\frac{1}{2}t^2\right)\middle|X\right]$$

□

**Утверждение 5**: Классификационная функция $\hat{f}:X\left(\Omega\right)\to\left\{0,1\right\}$, минимизирующая на множестве всех классификационных функций $f:X\left(\Omega\right)\to\left\{0,1\right\}$ взвешенную по $w\left(x,y\right):=\frac{e^{-yF(x)}}{Ee^{-yF(x)}}$ условную вероятность ошибки классификации $P_w\left(Y\neq f\middle|X\right)$, также минимизирует взвешенную условную среднеквадратическую ошибку $E_w\left(\left(Y-f\right)^2\middle|X\right)$, а также минимизирует квадратичную аппроксимацию $E\left[e^{-Y\left(F(X)+cf(X)\right)}\middle|X\right]$ при любом c>0. Т.е.

$$\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ P_w\left(Y\neq f\middle|X\right)=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left(\left(Y-f\right)^2\middle|X\right)=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E\left[e^{-YF(X)}\left(1-Ycf\left(X\right)+\frac{1}{2}\left(cf\left(X\right)\right)^2\right)\middle|X\right] \qquad \textbf{(20)}$$

**Доказательство**:

По утверждению 4 квадратичная аппроксимация $cf\mapsto E\left[e^{-Y\left(F(X)+cf(X)\right)}\middle|X\right]$ имеет вид: $E\left[e^{-YF(X)}\left(1-cf\left(X\right)Y+\frac{1}{2}c^2f\left(X\right)^2\right)\middle|X\right]$. Следовательно, для функции, на которой она принимает минимальное значение, справедливо

$$\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E\left[e^{-YF(X)}\left(1-cf(X)Y+\frac{1}{2}c^2 f(X)^2\right)\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E\left[\underbrace{\frac{e^{-YF(X)}}{E\left(e^{-YF(X)}\right)}}_{w(X,Y)}\left(1-cf(X)Y+\frac{1}{2}c^2 f(X)^2\right)\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[1-cf(X)Y+\frac{1}{2}c^2\underbrace{f(X)^2}_{=1}\middle|X\right]=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[1-cf(X)Y+\frac{1}{2}c^2\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[-cf(X)Y\middle|X\right]=\underbrace{\underset{c>0}{\arg}\underset{f:X(\Omega)\to\{0,1\}}{\min}}\ E_w\left[-f(X)Y\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[-1_{\{Y=f\}}+1_{\{Y\neq f\}}\middle|X\right]=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(P_w\left[Y\neq f\middle|X\right]-P_w\left[Y=f\middle|X\right]\right)=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(P_w\left[Y\neq f\middle|X\right]-\left(1-P_w\left[Y\neq f\middle|X\right]\right)\right)=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(2P_w\left[Y\neq f\middle|X\right]-1\right)=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ P_w\left[Y\neq f\middle|X\right]$$

С другой стороны

$$\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[\left(Y-f(X)\right)^2\middle|X\right]=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[\underbrace{Y^2}_{1}+\underbrace{f(X)^2}_{1}-2Yf(X)\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[-Yf(X)\middle|X\right]=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ E_w\left[-1_{\{Y=f\}}+1_{\{Y\neq f\}}\middle|X\right]=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(P_w\left[Y\neq f\middle|X\right]-P_w\left[Y=f\middle|X\right]\right)=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(P_w\left[Y\neq f\middle|X\right]-\left(1-P_w\left[Y\neq f\middle|X\right]\right)\right)=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\left(2P_w\left[Y\neq f\middle|X\right]-1\right)=$$

$$=\underset{f:X(\Omega)\to\{0,1\}}{\arg\min}\ P_w\left[Y\neq f\middle|X\right]$$

□

**Утверждение 6**:

Пусть дана $f:X(\Omega)\to\{0,1\}$ некая классификационная функция, для которой вероятность ошибки прогноза не более 0,5, т.е. выполняется $P_w(Y\neq f)\le\frac{1}{2}$. Тогда справедливо:

$$\hat{c}:=\frac{1}{2}\log\frac{1-P_w(Y\neq f)}{P_w(Y\neq f)}=\underset{c\ge 0}{\arg\min}\,E\left(e^{-Y(F(X)+cf(X))}\right)>0 \qquad \textbf{(21)}$$

**Доказательство**:

$$E\left(e^{-Y(F(X)+cf(X))}\right)=E_w\left(e^{-cYf(X)}\right)=e^{-c}E_w\left(1_{\{Y=f(X)\}}\right)+e^{c}E_w\left(1_{\{Y\neq f(X)\}}\right)=$$

$$=e^{-c}P_w\left(Y=f(X)\right)+e^{c}P_w\left(Y\neq f(X)\right)=$$

$$=e^{-c}\left(1-P_w\left(Y\neq f(X)\right)\right)+e^{c}P_w\left(Y\neq f(X)\right)$$

Функция $h(c) := e^{-c}(1-u) + e^{c}u, u \in (0,1)$ выпуклая по c>0 функция

$h'(c) = -e^{-c}(1-u) + e^{c}u = 0 \Leftrightarrow c = \frac{1}{2}\log\frac{1-u}{u}$, следовательно, c – глобальный минимум h(c), следовательно

$$\arg\min_{c\geq 0} E\left(e^{-Y\left(F(X)+cf(X)\right)}\right) = \arg\min_{c\geq 0}\left[e^{-c}\left(1-P_w\left(Y \neq f(X)\right)\right) + e^{c}P_w\left(Y \neq f(X)\right)\right] =$$

$$= \arg\min_{c\geq 0} h\left(P_w\left(Y \neq f(X)\right)\right) = \frac{1}{2}\log\frac{1-P_w\left(Y \neq f(X)\right)}{P_w\left(Y \neq f(X)\right)}$$

□

**Утверждение 7**: Функция, определяемая как знак истинной функции-параметра модели $\hat{F}$, является оптимальной с точки зрения минимизации вероятности ошибки прогноза, т.е.

$$sign\left(\hat{F}(X)\right) = \arg\min_{f:X(\Omega)\to\{0,1\}} P\left(f \neq Y \middle| X\right) \quad \textbf{(22)}$$

**Доказательство**:

Так как $\hat{F}(X) = \frac{1}{2}\log\frac{P\left(Y=1\middle|X\right)}{P\left(Y=-1\middle|X\right)}$, то $\hat{F}(X) \geq 0 \Leftrightarrow P\left(Y=1\middle|X\right) \geq P\left(Y=-1\middle|X\right)$, аналогично $\hat{F}(X) < 0 \Leftrightarrow P\left(Y=1\middle|X\right) < P\left(Y=-1\middle|X\right)$, следовательно $sign\left(\hat{F}(X)\right) = \arg\max_{i\in\{-1,1\}} P\left(Y=i\middle|X\right)$.

Из этого с учетом (3) следует, что функция $sign\left(\hat{F}(X)\right)$ минимизирует вероятность ошибки прогноза в выражении (2).

□

Ниже мы подробно рассмотрим алгоритм Discrete AdaBoost.

### *4.1.3 Алгоритм Discrete AdaBoost*

Идея алгоритма Discrete AdaBoost основывается на результате утверждения 3, заключающееся в том, что истинное значение функции-параметра $\hat{F}$ модели минимизирует функционалы $F \mapsto E\left(e^{-YF(X)}\middle|X\right)$ и $F \mapsto E\left(e^{-YF(X)}\right)$. В процессе выполнения алгоритма Discrete AdaBoost строится последовательность приближений $(F_m)_{m=1,\ldots,M}$ функции $\hat{F}$ путем итеративной минимизации функционалов $F \mapsto E\left(e^{-YF(X)}\middle|X\right)$ и $F \mapsto E\left(e^{-YF(X)}\right)$.

Пусть $F_m$ – текущее приближение $\hat{F}$. Будем искать следующее приближение в виде $F_{m+1} = F_m + c_m f_m, c_m \geq 0, f_m : X(\Omega) \to \{0,1\}$. Сначала значению каждой записи поставим в соответствие ее вес $w_m(x,y) := \frac{e^{-yF_m(x)}}{Ee^{-yF_m(x)}}$. Затем определим $f_m$ как классификационную функцию, минимизирующую взвешенную по w вероятность ошибки прогноза $P_{w_m}\left(Y \neq f\middle|X\right)$ или минимизирующую взвешенное среднеквадратичное отклонение прогноза $E_{w_m}\left((Y-f)^2\middle|X\right)$.

Согласно утверждению 5, $f_m$ минимизирует квадратичную аппроксимацию $f \to E\left(e^{-Y\left(F_m(X)+cf(X)\right)}\middle|X\right)$ при любом c>0. Затем определим вероятность ошибки прогноза $f_m$:

$P_{w_m}\left(Y \neq f_m\left(X\right)\right)$ и вычислим $c_m$ как $\frac{1}{2}\log\frac{1-P_{w_m}\left(Y \neq f_m\left(X\right)\right)}{P_{w_m}\left(Y \neq f_m\left(X\right)\right)}$. Согласно утверждению 6, $c_m$ минимизирует $c \to E\left(e^{-Y\left(F_m(X)+cf_m(X)\right)}\right)$. Затем, пересчитаем веса записей:

$$w_{m+1}\left(x,y\right)=\frac{e^{-yF_{m+1}(x)}}{Ee^{-yF_{m+1}(x)}}=\frac{e^{-yF_m(x)}}{\underbrace{Ee^{-yF_m(x)}}_{=w_m(x,y)}}\underbrace{\frac{Ee^{-yF_m(x)}}{Ee^{-yF_{m+1}(x)}}}_{1/K_m}e^{-yc_m f_m(x)}=\frac{1}{K_m}w_m\left(x,y\right)e^{-yc_m f_m(x)}, \quad \textbf{(23)}$$

где $\frac{1}{K_m}$ - нормирующий коэффициент, обеспечивающий равенство

$$\frac{1}{K_m}E\left(w_m\left(x,y\right)e^{-yc_m f_m(x)}\right)=E\left(w_{m+1}\left(x,y\right)\right)=1. \quad \textbf{(24)}$$

После M итераций мы получим в качестве оценки $\hat{F}$ функцию $F_M$. Так как по утверждению 7 $sign\left(\hat{F}\left(X\right)\right)$ является оптимальной с точки зрения минимизации вероятности ошибки классификационной функцией, то в качестве ее оценки в алгоритме Discrete AdaBoost берется $sign\left(F_M\left(X\right)\right)$.

Итак, сформулируем алгоритм Discrete AdaBoost:

1. На первой итерации, m=1 всем N записям базы данных присваиваем одинаковые веса w(x,y)=1/N.
2. Формируем обучающую выборку $D_m$, состоящую из исходной базы данных D, где каждая запись «размножена» пропорционально своему весу.
3. На основании базового классификационного алгоритмы строится модель $f_m$ на данных $D_m$, минимизирующая вероятность ошибки прогноза или величину среднеквадратической ошибки прогноза.
4. Вычисляем оценку вероятности ошибки $P_{w_m}\left(Y \neq f_m\left|X\right.\right)$ базового классификатора $f_m$ как
$$e_m=\sum_{i=1}^{N}w_m\left(x_i,y_i\right)1_{\left\{f_m(x_i)\neq y_i\right\}}. \quad \textbf{(25)}$$
5. Если $e_m$>0,5, то останавливаемся (в этом случае бинарный классификационный алгоритм $f_m$ нельзя считать адекватным, и мы не можем получить $c_m$>=0).
6. Получаем $c_m$ как $\frac{1}{2}\log\frac{1-e_m}{e_m}$.
7. Обновляем $F_{m+1} := F_m+c_mf_m$.
8. Обновляем веса каждой записи согласно (23):
$$w_{m+1}\left(x_i,y_i\right)=\frac{1}{K_m}w_m\left(x_i,y_i\right)e^{-y_ic_m f_m(x_i)} \quad \textbf{(26)}$$
(вес увеличивается у неправильно классифицированных записей на $\frac{1}{K_m}e^{c_m}$ и уменьшается у правильно классифицированных записей на ту же величину). С учетом (24), оценка нормирующего коэффициент $K_m=E\left(w_m\left(x,y\right)e^{-yc_m f_m(x)}\right)$ на обучающих данных вычисляется как
$$K_m=\sum_{i=1}^{N}w_m\left(x_i,y_i\right)e^{-y_ic_m f_m(x_i)} \quad \textbf{(27)}$$
9. Если m < M, goto 2, else вычисляем итоговый классификатор ансамбля как $sign\left(F_M\left(X\right)\right)$.

Далее получим верхнюю границу ошибки прогнозирования алгоритма Discrete AdaBoost на обучающих данных.

**Утверждение 8**: Для обучающей базы данных $D=\left\{\left(x_1,y_1\right),...,\left(x_N,y_N\right)\right\}$, параметра $\theta$, и функции $h_M : X\left(\Omega\right)\to\left[0,1\right], h_M := \frac{F_M}{\sum_{m=1}^{M} c_m} = \frac{\sum_{m=1}^{M} c_m f_m}{\sum_{m=1}^{M} c_m}$ справедлива оценка:

$$\frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right)\le\theta\right\}} \le e^{\theta\sum_{m=1}^{M} c_m}\left(\prod_{m=1}^{M} K_m\right) \quad \textbf{(28)}$$

**Доказательство**:

Из (27):

$$K_m = \sum_{i=1}^{N} w_m\left(x_i,y_i\right)e^{-y_i c_m f_m\left(x_i\right)} = \sum_{i:y_i=f_m\left(x_i\right)} w_m\left(x_i,y_i\right)e^{-c_m} + \sum_{i:y_i\ne f_m\left(x_i\right)} w_m\left(x_i,y_i\right)e^{c_m} =$$

$$= e^{-c_m}\sum_{i:y_i=f_m\left(x_i\right)} w_m\left(x_i,y_i\right) + e^{c_m}\sum_{i:y_i\ne f_m\left(x_i\right)} w_m\left(x_i,y_i\right)$$

Из (26) и (27) получаем

$$\sum_{i=1}^{N} w_m\left(x_i,y_i\right)=1, \quad \textbf{(29)}$$

следовательно $\sum_{i:y_i=f_m\left(x_i\right)} w_m\left(x_i,y_i\right)=1-\sum_{i:y_i\ne f_m\left(x_i\right)} w_m\left(x_i,y_i\right)$, и, значит, с учетом (25)

$$K_m = e^{-c_m}\underbrace{\sum_{i:y_i=f_m\left(x_i\right)} w_m\left(x_i,y_i\right)}_{=1-e_m} + e^{c_m}\underbrace{\sum_{i:y_i\ne f_m\left(x_i\right)} w_m\left(x_i,y_i\right)}_{=e_m} = e^{-c_m}\left(1-e_m\right)+e^{c_m}e_m \quad \textbf{(30)}$$

Далее,

$$y_i h_M\left(x_i\right)\le\theta \Leftrightarrow y_i\frac{\sum_{m=1}^{M} c_m f_m\left(x_i\right)}{\sum_{m=1}^{M} c_m}\le\theta \Leftrightarrow y_i\sum_{m=1}^{M} c_m f_m\left(x_i\right)\le\theta\sum_{m=1}^{M} c_m \Leftrightarrow$$

$$\Leftrightarrow \theta\sum_{m=1}^{M} c_m - y_i\sum_{m=1}^{M} c_m f_m\left(x_i\right)\ge 0 \Leftrightarrow \exp\left(\theta\sum_{m=1}^{M} c_m - y_i\sum_{m=1}^{M} c_m f_m\left(x_i\right)\right)\ge 1$$

Следовательно $1_{\left\{y_i h_M\left(x_i\right)\le\theta\right\}} \le \exp\left(\theta\sum_{i=1}^{M} c_m - y_i\sum_{i=1}^{M} c_m f_m\left(x_i\right)\right)$ **(31)**

Так как согласно 1-ому шагу алгоритма $w_1\left(x_i,y_i\right)=\frac{1}{N}$, а согласно 7-ому шагу алгоритма $w_{m+1}\left(x_i,y_i\right)=\frac{1}{K_m}w_m\left(x_i,y_i\right)e^{-y_i c_m f_m\left(x_i\right)}$, то по индукции получим:

$$w_M\left(x_i,y_i\right)=\frac{1}{K_M}w_{M-1}\left(x_i,y_i\right)e^{-y_ic_{M-1}f_{M-1}\left(x_i\right)}=\frac{e^{-y_ic_{M-1}f_{M-1}\left(x_i\right)}e^{-y_ic_{M-2}f_{M-2}\left(x_i\right)}}{K_MK_{M-1}}w_{M-2}\left(x_i,y_i\right)=$$

$$=\ldots=\frac{1}{N}\frac{\exp\left(\sum_{j=1}^{M}-y_ic_jf_j\left(x_i\right)\right)}{\prod_{m=1}^{M}K_m}$$

Следовательно $\frac{1}{N}\exp\left(\sum_{j=1}^{M}-y_ic_jf_j\left(x_i\right)\right)=w_M\left(x_i,y_i\right)\prod_{m=1}^{M}K_m$ **(32)**

Из (31) получаем с учетом (32):

$$\frac{1}{N}\sum_{i=1}^{N}1_{\left\{y_ih_M\left(x_i\right)\le\theta\right\}}\le\frac{1}{N}\sum_{i=1}^{N}\exp\left(\theta\sum_{m=1}^{M}c_m-y_i\sum_{m=1}^{M}c_mf_m\left(x_i\right)\right)=$$

$$=\exp\left(\theta\sum_{m=1}^{M}c_m\right)\frac{1}{N}\sum_{i=1}^{N}\exp\left(-y_i\sum_{m=1}^{M}c_mf_m\left(x_i\right)\right)=$$

$$=\exp\left(\theta\sum_{m=1}^{M}c_m\right)\sum_{i=1}^{N}\underbrace{\frac{1}{N}\exp\left(-y_i\sum_{m=1}^{M}c_mf_m\left(x_i\right)\right)}_{=w_M\left(x_i,y_i\right)\prod_{m=1}^{M}K_m,(32)}=$$

$$=\exp\left(\theta\sum_{m=1}^{M}c_m\right)\sum_{i=1}^{N}w_M\left(x_i,y_i\right)\prod_{m=1}^{M}K_m=$$

$$=\exp\left(\theta\sum_{m=1}^{M}c_m\right)\prod_{m=1}^{m}K_m\underbrace{\sum_{i=1}^{N}w_M\left(x_i,y_i\right)}_{=1,(29)}=\exp\left(\theta\sum_{m=1}^{M}c_m\right)\prod_{m=1}^{M}K_m$$

□

Далее покажем, что при достаточно слабых требованиях к точности базовых классификаторов, ошибка классификации алгоритма Discrete AdaBoost стремится к нулю при увеличении числа итераций.

**Утверждение 9**:

Если для взвешенной ошибки каждого базового классификатора справедливо

$$e_m=\sum_{i=1}^{N}w_m\left(x_i,y_i\right)1_{\left\{f_m\left(x_i\right)\ne y_i\right\}}=\frac{1}{2}-\gamma_m,\text{ где }0<\gamma_m<\frac{1}{2},\sum_{m=1}^{\infty}{\gamma_m}^2=\infty,\quad\textbf{(33)}$$

то средняя ошибка классификатора $sign\left(F_M\right)$ ансамбля Discrete AdaBoost на обучающих данных стремится к нулю, т.е.

$$\lim_{M\to\infty}\frac{1}{N}\sum_{i=1}^{N}1_{\left\{y_i\ne sign\left(F_M\left(x_i\right)\right)\right\}}=0\quad\textbf{(34)}$$

**Доказательство**:

Так как на 6-ом шаге алгоритма мы выбираем $c_m = \frac{1}{2}\log\frac{1-e_m}{e_m}$, то, с учетом (30), получаем

$$K_m = e^{-c_m}\left(1-e_m\right)+e^{c_m}e_m = e^{-\frac{1}{2}\log\frac{1-e_m}{e_m}}\left(1-e_m\right)+e^{\frac{1}{2}\log\frac{1-e_m}{e_m}}e_m =$$

$$= e^{\log\left(\frac{1-e_m}{e_m}\right)^{-\frac{1}{2}}}\left(1-e_m\right)+e^{\log\left(\frac{1-e_m}{e_m}\right)^{\frac{1}{2}}}e_m = \left(\frac{e_m}{1-e_m}\right)^{\frac{1}{2}}\left(1-e_m\right)+\left(\frac{1-e_m}{e_m}\right)^{\frac{1}{2}}e_m =$$

$$= \frac{\sqrt{e_m}}{\sqrt{1-e_m}}\left(1-e_m\right)+\frac{\sqrt{1-e_m}}{\sqrt{e_m}}e_m = \sqrt{e_m}\sqrt{1-e_m}+\sqrt{1-e_m}\sqrt{e_m} = 2\sqrt{1-e_m}\sqrt{e_m}$$

Т.е. $K_m = 2\sqrt{1-e_m}\sqrt{e_m}$ **(35)**

Подставляя (35) в (28), получаем:

$$\frac{1}{N}\sum_{i=1}^{N}1_{\{y_i h_M(x_i)\le\theta\}} \le e^{\theta\sum_{m=1}^{M}c_m}\left(\prod_{m=1}^{M}K_m\right) = e^{\theta\sum_{m=1}^{M}c_m}\prod_{m=1}^{M}\left(2\sqrt{1-e_m}\sqrt{e_m}\right) \quad \textbf{(36)}$$

При $\theta := 0$ (36) принимает вид:

$$\frac{1}{N}\sum_{i=1}^{N}1_{\{y_i h_M(x_i)\le 0\}} \le \prod_{m=1}^{M}\left(2\sqrt{1-e_m}\sqrt{e_m}\right) \quad \textbf{(37)}$$

С учетом (33) получаем из (37):

$$\frac{1}{N}\sum_{i=1}^{N}1_{\{y_i h_M(x_i)\le 0\}} \le \prod_{m=1}^{M}\left(2\sqrt{1-\left(\frac{1}{2}-\gamma_m\right)}\sqrt{\left(\frac{1}{2}-\gamma_m\right)}\right) =$$

$$= \prod_{m=1}^{M}\left(2\sqrt{\left(\frac{1}{2}+\gamma_m\right)\left(\frac{1}{2}-\gamma_m\right)}\right) = \prod_{m=1}^{M}\left(2\sqrt{\left(\frac{1}{4}-{\gamma_m}^2\right)}\right) =$$

$$= \prod_{m=1}^{M}\sqrt{4\left(\frac{1}{4}-{\gamma_m}^2\right)} = \prod_{m=1}^{M}\sqrt{1-4{\gamma_m}^2} \quad \textbf{(38)}$$

Так как $\log x \le x-1$, то $\log\left(1-4{\gamma_m}^2\right) \le \left(1-4{\gamma_m}^2\right)-1 = -4{\gamma_m}^2$, следовательно

$$\prod_{m=1}^{M}\sqrt{1-4{\gamma_m}^2} = \exp\left(\log\prod_{m=1}^{M}\sqrt{1-4{\gamma_m}^2}\right) = \exp\left(\frac{1}{2}\sum_{m=1}^{M}\log\left(1-4{\gamma_m}^2\right)\right) \le$$

$$\le \exp\left(\frac{1}{2}\sum_{m=1}^{M}\left(-4{\gamma_m}^2\right)\right) = \exp\left(-2\sum_{m=1}^{M}{\gamma_m}^2\right) \quad \textbf{(39)}$$

Подставляем (39) в (38):

$$\frac{1}{N}\sum_{i=1}^{N}1_{\{y_i h_M(x_i)\le 0\}} \le \exp\left(-2\sum_{m=1}^{M}{\gamma_m}^2\right) \quad \textbf{(40)}$$

С учетом (33) $\sum_{m=1}^{\infty}{\gamma_m}^2 = \infty$, переходя к пределу, получаем для правой части (40)

$$\lim_{M\to\infty}\frac{1}{N}\sum_{i=1}^{N}1_{\{y_i h_M(x_i)\le 0\}} \le \lim_{M\to\infty}\exp\left(\underbrace{-2\sum_{m=1}^{M}{\gamma_m}^2}_{\to-\infty}\right) = 0 \quad \textbf{(41)}$$

С другой стороны, $y_i \neq sign\left(F_M\left(x_i\right)\right) \Leftrightarrow y_i F_M\left(x_i\right) < 0 \Leftrightarrow y_i \frac{F_M\left(x_i\right)}{\sum_{m=1}^{M} c_m} < 0 \Leftrightarrow y_i h_M\left(x_i\right) < 0$,

следовательно

$$1_{\left\{y_i \neq sign\left(F_M\left(x_i\right)\right)\right\}} = 1_{\left\{y_i h_M\left(x_i\right) < 0\right\}} \Rightarrow \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i \neq sign\left(F_M\left(x_i\right)\right)\right\}} = \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right) < 0\right\}} \Rightarrow$$

$$\Rightarrow \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i \neq sign\left(F_M\left(x_i\right)\right)\right\}} \leq \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right) < 0\right\}} + \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right) = 0\right\}} = \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right) \leq 0\right\}}$$

и с учетом (41):

$$\lim_{M\to\infty} \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i \neq sign\left(F_M\left(x_i\right)\right)\right\}} \leq \lim_{M\to\infty} \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{y_i h_M\left(x_i\right) \leq 0\right\}} = 0$$

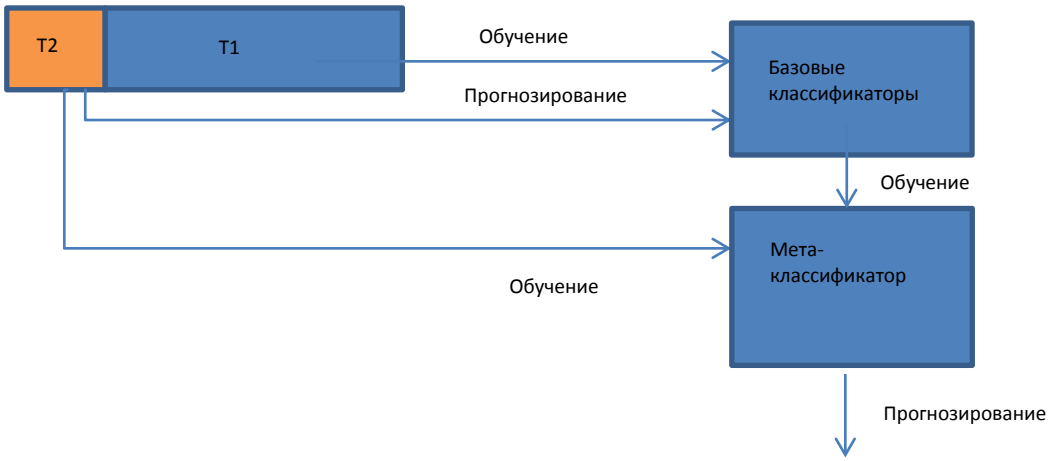
□

## 5. Stacking

Алгоритм Stacking не основывается на математической модели. Его идея заключается в использовании в качестве базовых моделей различных классификационных алгоритмов, обучаемых на одинаковых данных. Затем, мета-классификатор обучается на исходных данных, дополненных результатами прогноза базовых алгоритмов. Иногда мета-классификатор использует при обучении не результаты прогноза базовых алгоритмов, а полученные ими оценки параметров распределения, например, оценки вероятностей каждого класса.

Идея Stacking заключается в том, что мета-алгоритм учится различать какому из базовых алгоритмов следует «доверять» на каких областях входных данных.

Алгоритм Stacking обучается с использованием перекрестной валидации (cross-validation): данные случайным образом n раз делятся на две части – T1 и T2, содержащие каждый раз (например, при n=10) 90% и 10% данных соответственно. Базовые алгоритмы обучаются на данных из T1, а затем применяются к данным из T2. Данные из T2 объединяются с прогнозами базовых классификаторов и обучают мета-классификатор. Так как после n разбиений все имеющиеся данные попадут в итоге в какой-то из T1 и в какой-то из T2, то мета-классификатор будет обучен на полном объеме данных.



## 6. Литература


1. Robert E. Schapire , “The strength of weak learnability”, 1990
2. David H. Wolpert, “Stacked Generalization”, 1992
3. **Leo Breiman, “Bagging Predictors”, 1994**
4. Yoav Freund, “Boosting a weak learning algorithm by majority”, 1995
5. Yoav Freund, Robert E. Schapire “Experiments with a New Boosting Algorithm”, 1996
6. Leo Breiman, “Combining predictors”, 1998
7. **Jerome Friedman, Trevor Hastie, Robert Tibshirani , “Additive Logistic Regression: a Statistical View of Boosting”, 1998**
8. Eric Bauer, Ron Kohavi, “An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”, 1998
9. Nikunj C. Oza, “Ensemble Data Mining Methods”, 1998
10. Robert E. Schapire, Yoram Singer, “Improved Boosting Algorithms Using Confidence-rated Predictions”, 1999
11. **John F. Elder, Greg Ridgeway, “Combining Estimators to Improve Performance”, 1999**
12. Greg Ridgeway, “State of Boosting”, 1999
13. Jerome Friedman, Peter Hall, “On bagging and nonlinear estimation”, 1999
14. “Stochastic Gradient Boosting”, 1999
15. Peter Bühlmann, Bin Yu, “Analyzing Bagging”, 2002
16. **Ron Meir, “Boosting Tutorial”, 2002**
17. Xiangliang Zhang, “Classification Ensemble Methods”
18. S. B. Kotsiantis, P. E. Pintelas, “Combining Bagging and Boosting”, 2005
19. Claudia Perlich, Grzegorz Swirszcz, “On CrossValidation and Stacking”
20. Thomas Dietterich, “Ensemble methods in Machine Learning”
21. Andrew Fast , David Jensen, “Why Sacked Models Perform Effective Collective Classification”, 2008