Композиции решающих функций

Неделько В. М.

Институт математики СО РАН, г. Новосибирск
nedelko@math.nsc.ru

Спецкурс «Теория статистических решений».
Лекция 5.

Основные понятия

$X = R^n$ – пространство значений прогнозирующих переменных,
$Y = \{-1, 1\}$ – прогнозируемая переменная,
$D = X \times Y$.

Решающая функция (алгоритм классификации)

$$f : X \to Y.$$

$V = \left((x^i, y^i) \in D \,|\, i = \overline{1, N}\right)$ – случайная независимая выборка, $V \in D^N$.

$Q \colon D^N \to \Phi$ — метод построения решающих функций,
Φ — заданный класс решающих функций.

Композиции классификаторов

Обобщение решающей функции: $\lambda\colon X \to [0, +\infty)$ — вводится пространство оценок.
Пусть имеются T решающих функций $\lambda_1(x)$, ..., $\lambda_T(x)$.
Композиция есть решение в виде

$$\lambda(x) = C(\lambda_1(x), ..., \lambda_T(x)),$$

где $C(\cdot, ..., \cdot)$ – монотонна по всем аргументам.

Функции $\lambda_t(x)$ принимают значения из прострнства оценок, значения функции $\lambda(x)$ – из множества Y.

Линейные композиции

Линейная композиция

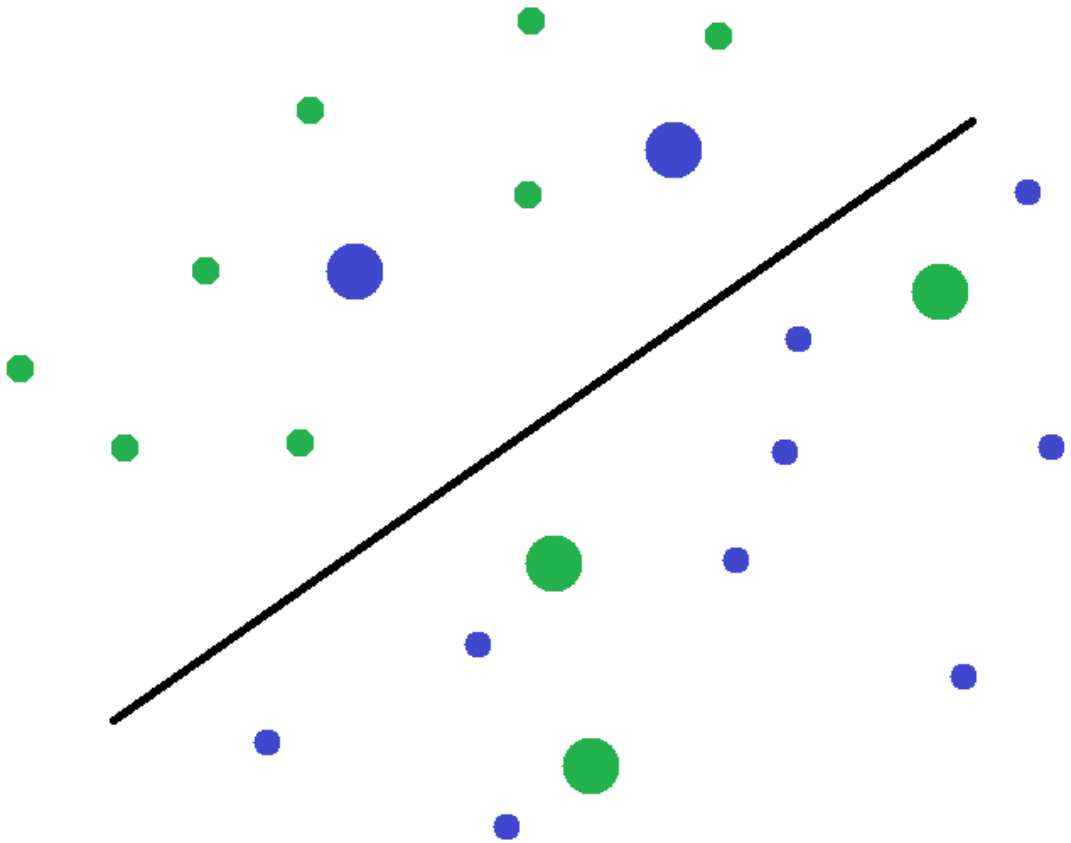
$$\lambda(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \lambda_t(x))\right), \quad \alpha_t \geq 0.$$

Методы построения композиций

- Бустинг (AdaBoost, градиентный бустинг)
- Бэггинг и метод случайных подпространств
- Голосование (простое, взвешенное)
- Стэкинг (решения в качестве новых переменных)

Смеси алгоритмов – если α_t зависит от x, идея областей компетентности.

Идея бустинга



Увеличиваем веса объектов, на которых допущена ошибка.

Алгоритм AdaBoost

В методе AdaBoost решение строится в виде композиции

$$\lambda(x) = \text{sign}(\beta(x)), \quad \beta(x) = \sum_{t=1}^{T} \alpha_t \lambda_t(x)),$$

где базовые классификаторы $\lambda_t(x)$ и их веса α_t находятся следующим образом.
Первый базовый классификатор строится базовым методом на основе исходной выборки, объектам которой приписаны начальные веса $w^1 = (w_1^1, ..., w_N^1)$.
Заметим, что мы будем задавать начальные веса объектом в соответствии с выбранным распределением, но в стандартном варианте метода начальные веса выбираются одинаковыми, т.е. $w_i^1 = \frac{1}{N}$.

Пересчёт весов

Вес построенного базового классификатора в композиции определяется по формуле

$$\alpha_t = \frac{1}{2} \ln \frac{\widetilde{M}^+(V, w^t, \lambda_t)}{\widetilde{M}^-(V, w^t, \lambda_t)},$$

где

$$\widetilde{M}^+(V, w, \lambda) = \sum_{i=1}^{N} w_i \cdot I(y^i = \lambda(x^i)),$$

$$\widetilde{M}^-(V, w, \lambda) = \sum_{i=1}^{N} w_i \cdot I(y^i = -\lambda(x^i)).$$

Итерационный процесс

Следующие базовые классификаторы строятся тем же базовым методом по выборке, веса объектов в которой вычисляются по формулам

$$w_i^{t+1} = \frac{\bar{w}_i^{t+1}}{\sum_{i=1}^{N} \bar{w}_i^{t+1}}, \qquad \bar{w}_i^{t+1} = w_i^t \cdot e^{-\alpha_t y^i \lambda_t(x^i)}.$$

Веса правильно классифицированных объектов умножаются на $e^{-\alpha_t}$, а веса неправильно классифицированных объектов умножаются на e^{α_t}.

Сходимость процесса бустинга

Если бустинг не останавливать, то он будет стремиться оценить функцию условной вероятности.

- Если в точке пространства находится один объект выборки, то эмпирическая условная вероятность соответствующего класса в этой точке равна 1.
- Бустинг приближает вероятность через логистическую функцию.

Если условные вероятности нигде не равны 0 или 1, то бустинг сходится (веса деревьев стремятся к 0).

Полезно исследовать поведение методов не только на выборке, но и на распределениях.

Оценивание условной вероятности

Условную вероятность $g(x) = \mathsf{P}(y = 1 \,|\, x)$ представим как находящиеся в точке x два объекта: класса 1 с весом $w_0 g(x)$ и класса -1 с весом $w_0(1 - g(x))$.

В результате выполнения бустинга вес первого объекта станет равным

$$w^{+1}(x) = w_0 g(x) \cdot A e^{-\beta(x)},$$

где константа A есть произведение всех нормировочных множителей.

Конечный вес второго объекта есть

$$w^{-1}(x) = w_0(1 - g(x)) \cdot A e^{\beta(x)}.$$

Если приравнять веса объектов, то получим

$$g(x) = \frac{1}{1 + e^{-2\beta(x)}}.$$

Градиентный бустинг

Зададимся целью получить композицию деревьев, которая оценила бы условную вероятность в форме логистической функции от суммы прогнозов, т.е.

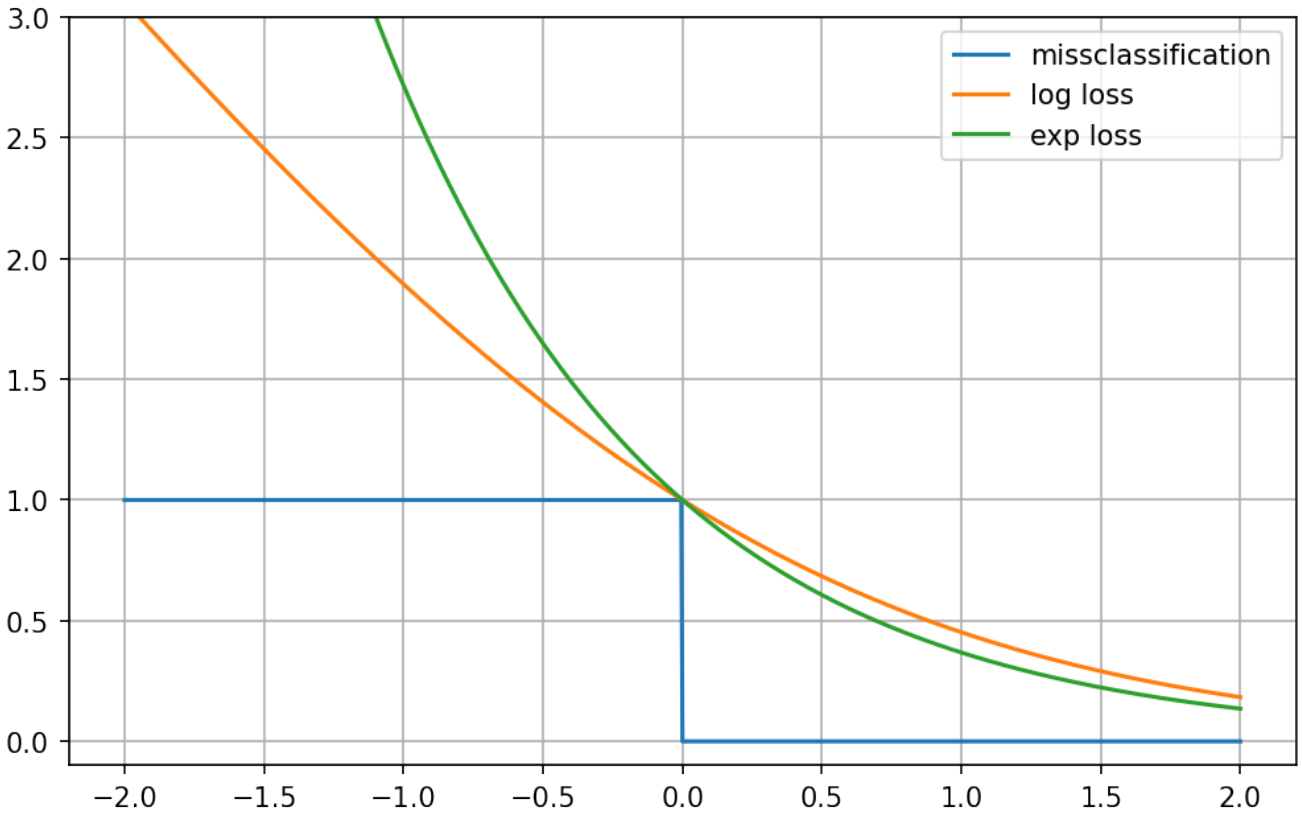
$$g(x) = \frac{1}{1 + e^{-\beta(x)}}.$$

Выразим теперь логарифмическую функцию потерь

$$L(y, g(x)) = \begin{cases} -\ln g(x), & y = 1 \\ -\ln(1 - g(x)), & y = -1 \end{cases} = \ln\left(1 + e^{-y\beta(x)}\right).$$

Градиентный бустинг строит композицию $\beta(x)$, минимизируя функцию потерь на выборке.

Функции потерь



Градиентный бустинг с экспоненциальной функцией потерь примерно эквивалентен AdaBoost.

Обобщённый наивный байесовский классификатор

Ранее мы для наивного байесовского классификатора получили выражение в виде логистической регрессии

$$g(x) = \sigma\left(u_0 + \sum_{j=1}^{n} u_j \sigma^{-1}(g_j(x_j))\right),$$

при $u_0 = (n-1)(\ln p - \ln(1-p))$, $u_j = 1$.

Обобщим это выражение, считая веса свободными параметрами и допуская произвольные оценочные функции. Получим

$$g(x) = \sigma\left(u_0 + \sum_{j=1}^{n} u_j s(x_j)\right).$$

Бустинг на пороговых классификаторах

Бустинг на пороговых классификаторах («пнях») является разновидностью обобщённого наивного байесовского классификатора.
Действительно, каждая $\lambda_t(x)$ в композиции

$$\beta(x) = \sum_{t=1}^{T} \alpha_t \lambda_t(x))$$

зависит только от одной переменной X_{i_t}, поэтому после группировки слагаемых выражение можно привести к виду

$$2\beta(x) = \sum_{i=1}^{n} u_i s(x)).$$

Подставив в выражение для $g(x)$, получим искомый вид.

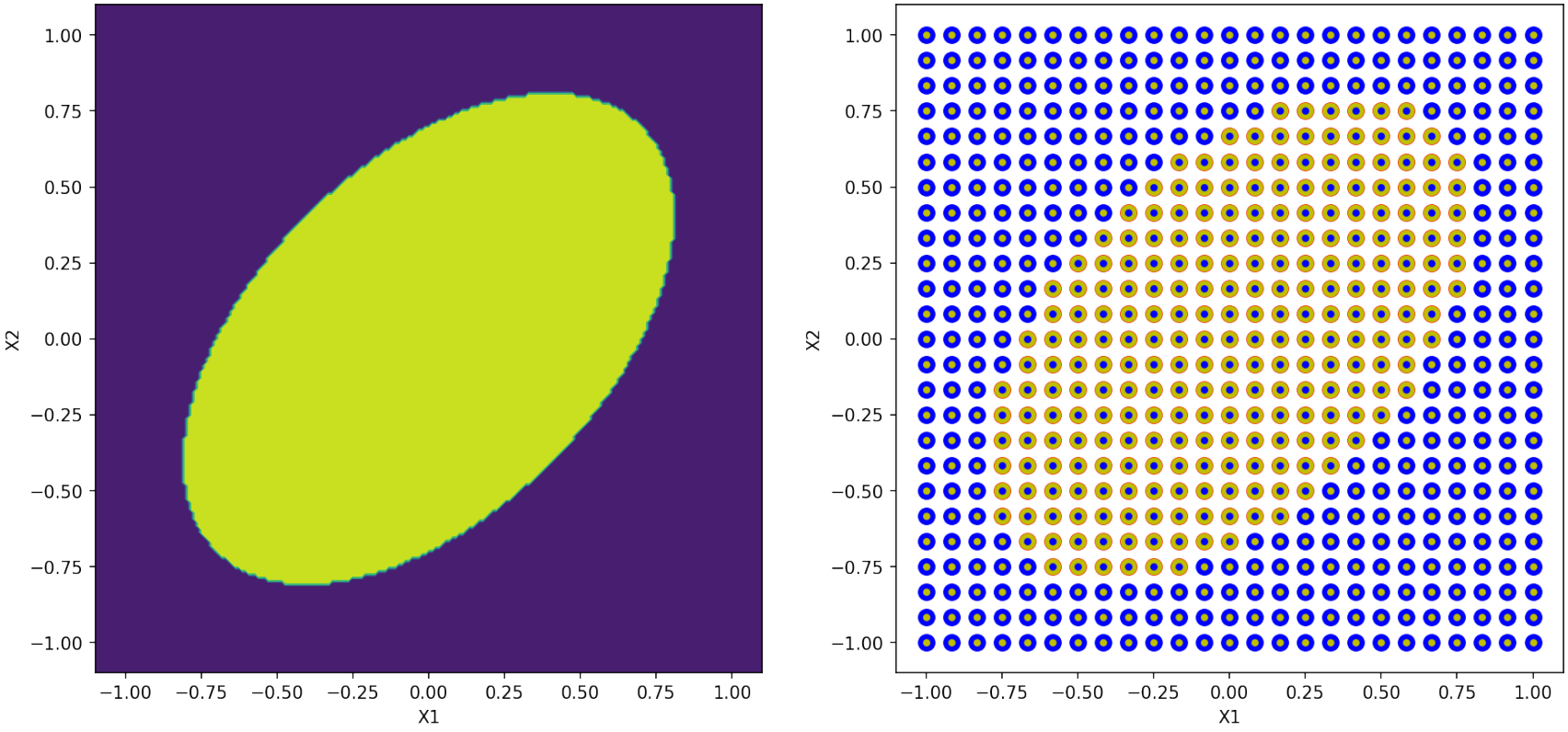
Бустинг на деревьях и ряд Бахадура

Модель можно естественным образом обобщить по аналогии с рядом Бахадура, включив возможность учитывать зависимости между переменными, последовательно добавляя парные зависимости, зависимости в тройках и т.д.

$$g(x) = \sigma \left(u_0 + \sum_{j=1}^{n} u_j s_j(x_j) + \sum_{j,k} u_{jk} s_{jk}(x_j, x_k) + \right.$$

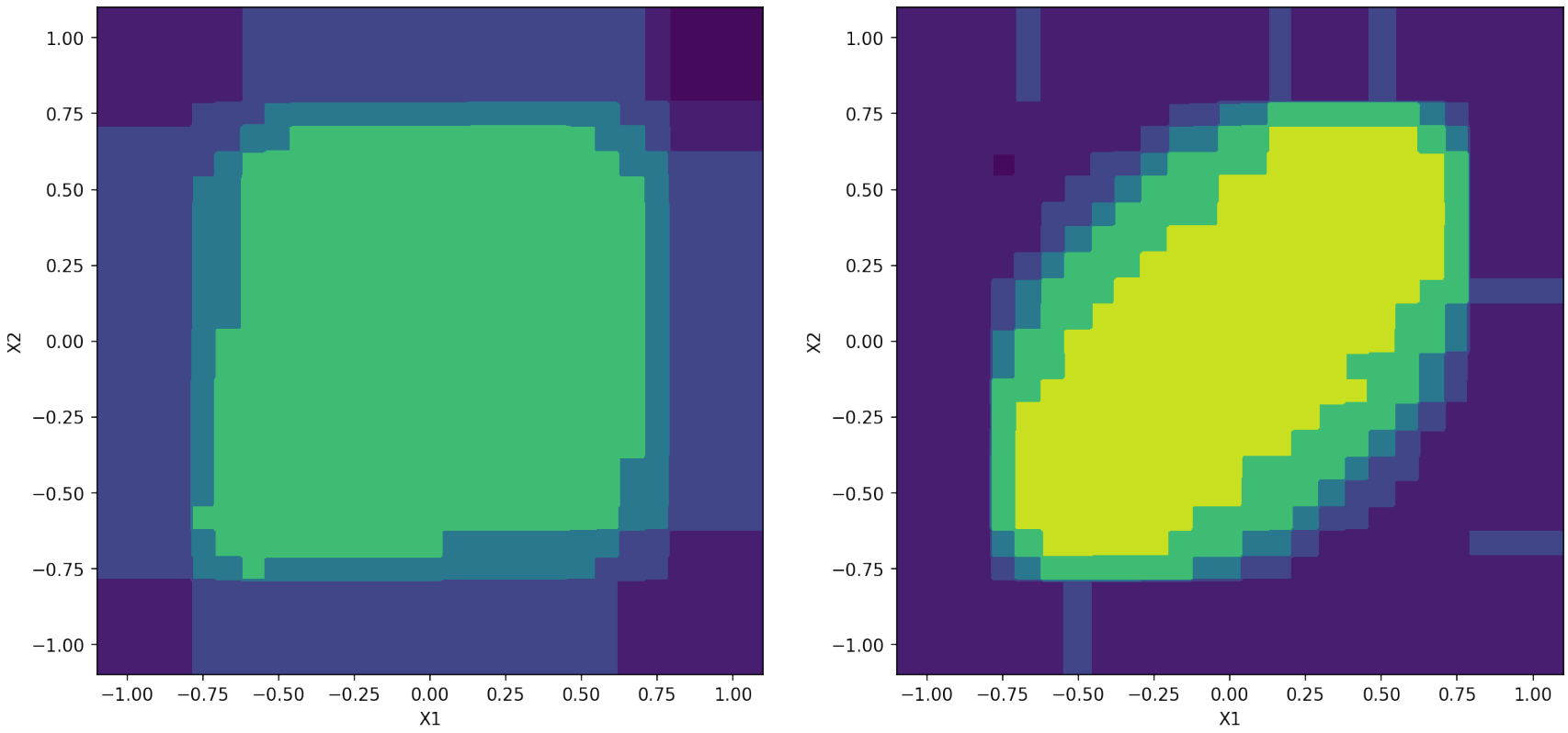
$$\left. + \sum_{j,k,l} u_{jkl} s_{jkl}(x_j, x_k, x_l) + \ldots \right).$$

Аппроксимация распределения выборкой



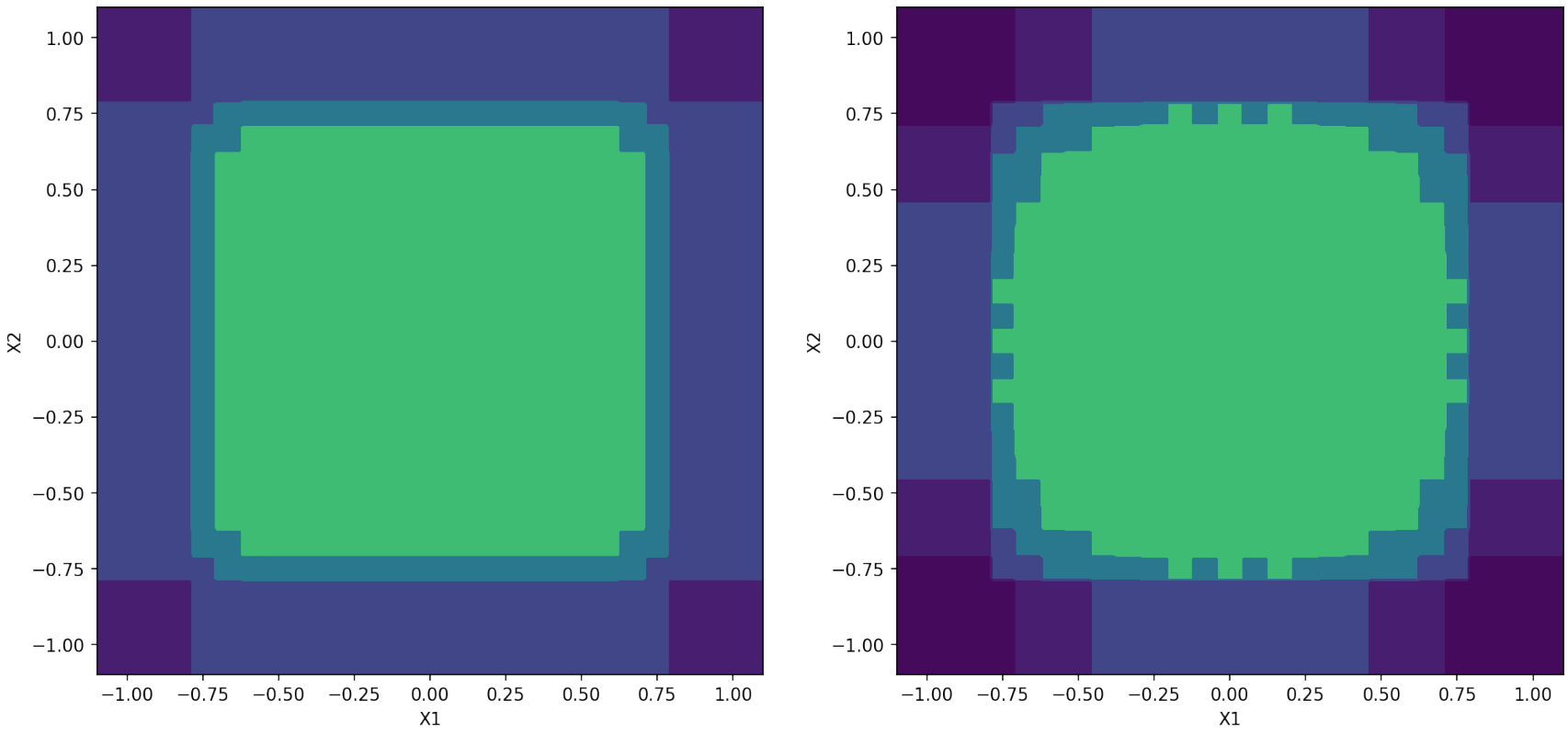
Условная вероятность «чужого» класса в каждой точке равна 0,1.

Решение градиентным бустингом



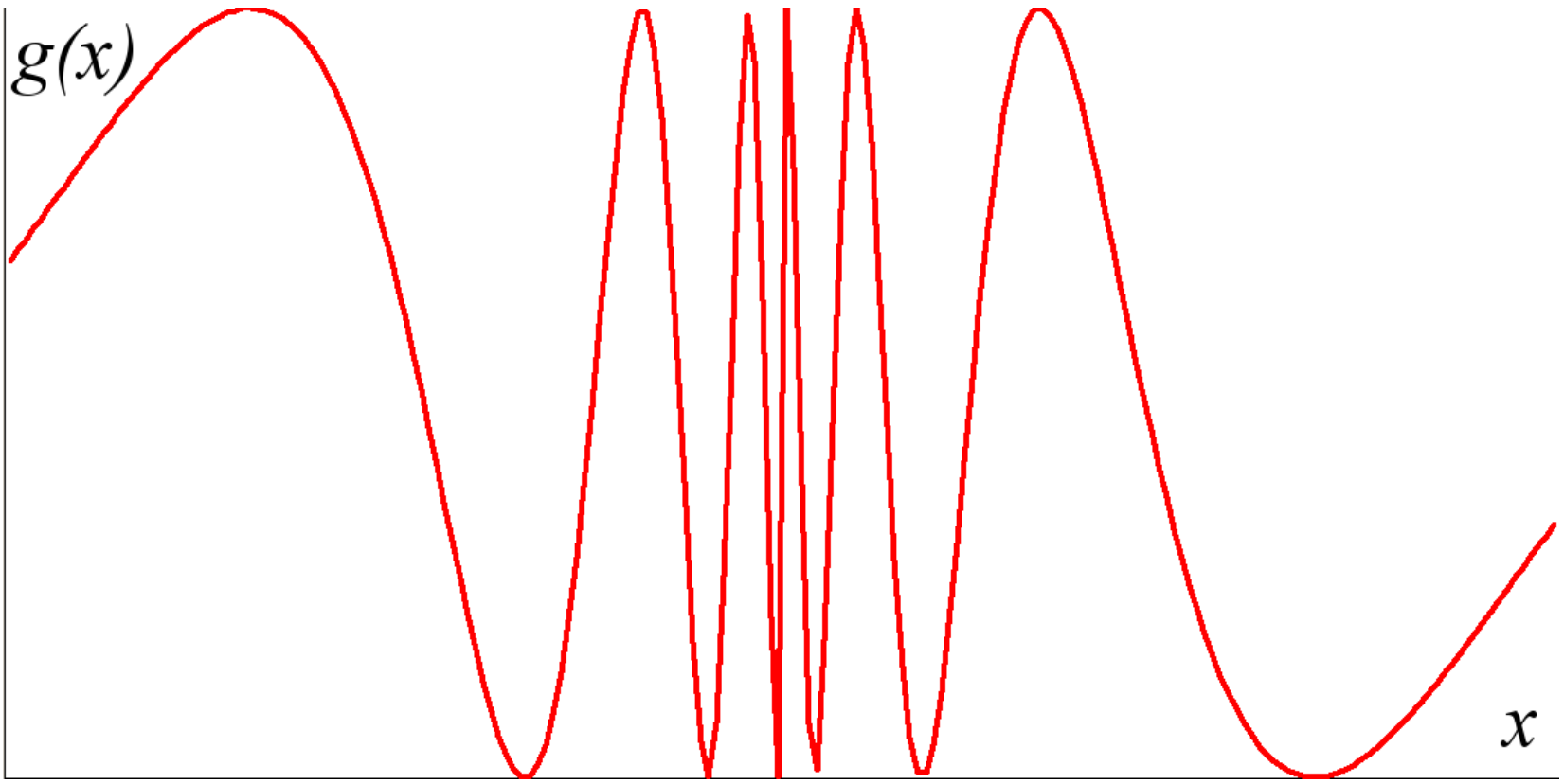
Глубина дерева 2, решения на основе 15 и 1000 деревьев.

Бустинг на деревьях минимальной глубины



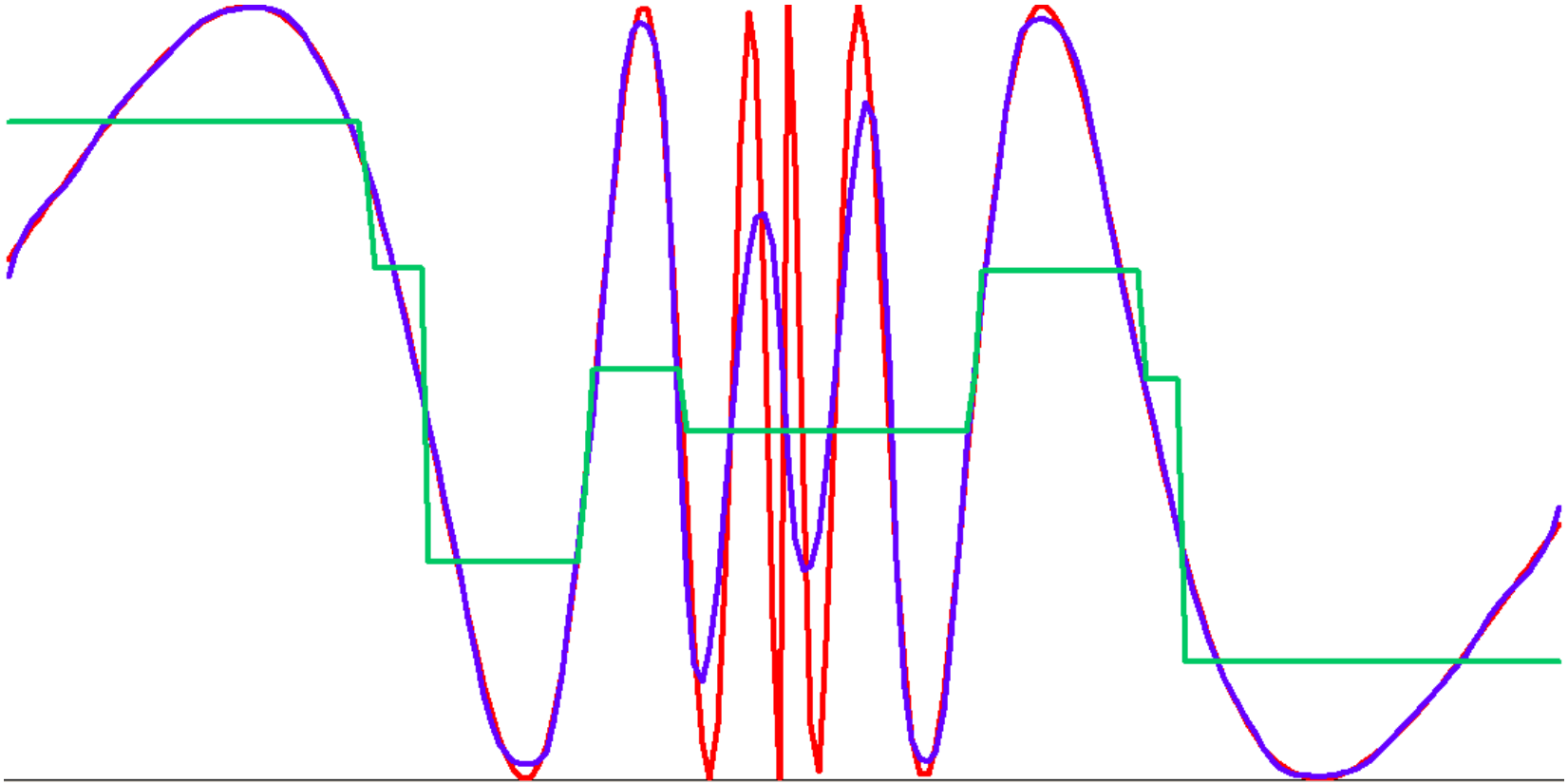
Глубина дерева 1, решения на основе 15 и 1000 деревьев.

Модельный пример



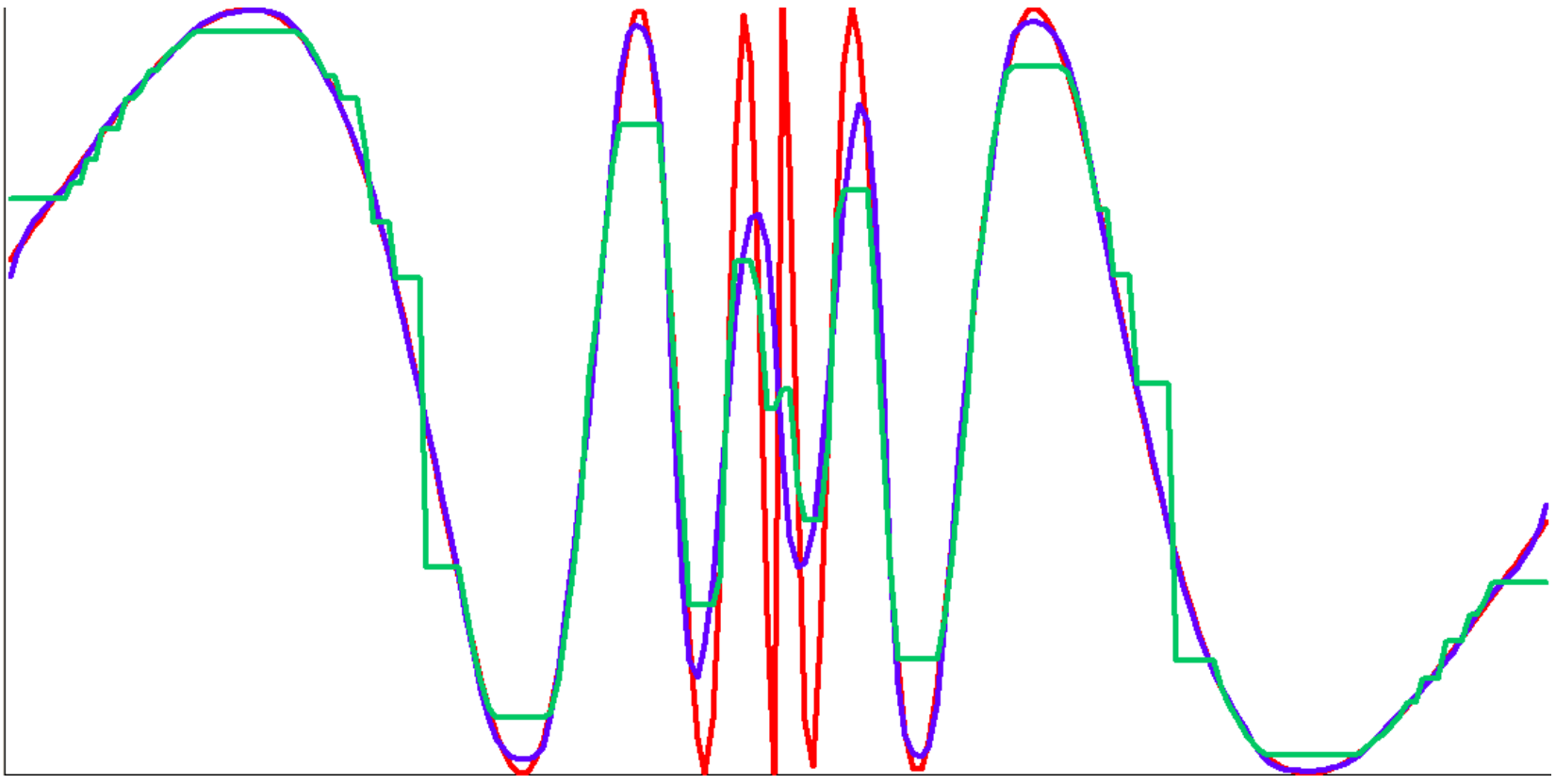
Функция условной вероятности.

Аппроксимация сплайном



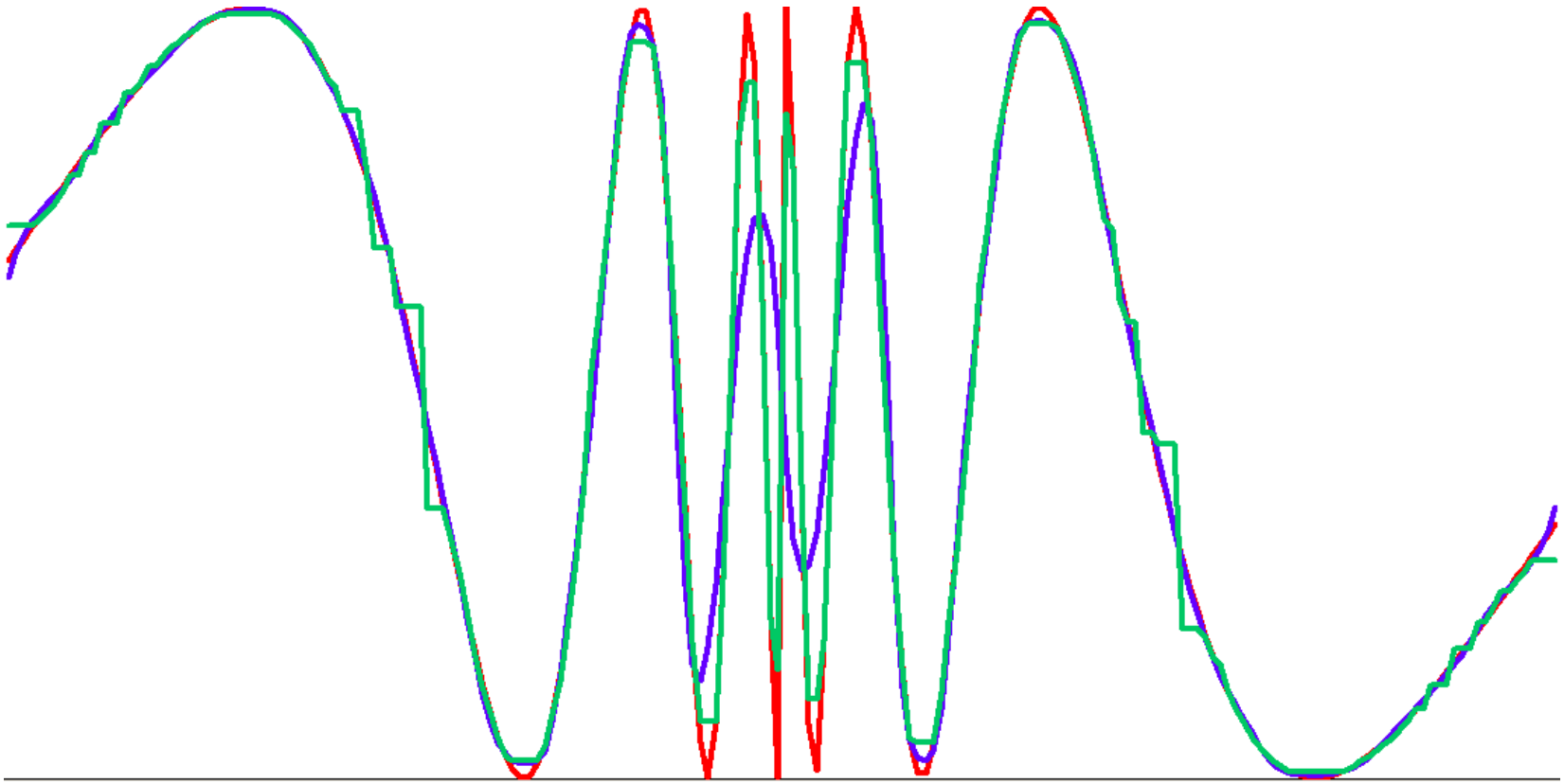
Кубический сплайн на 20 интервалов.
AdaBoost 10 итераций.

Boosting



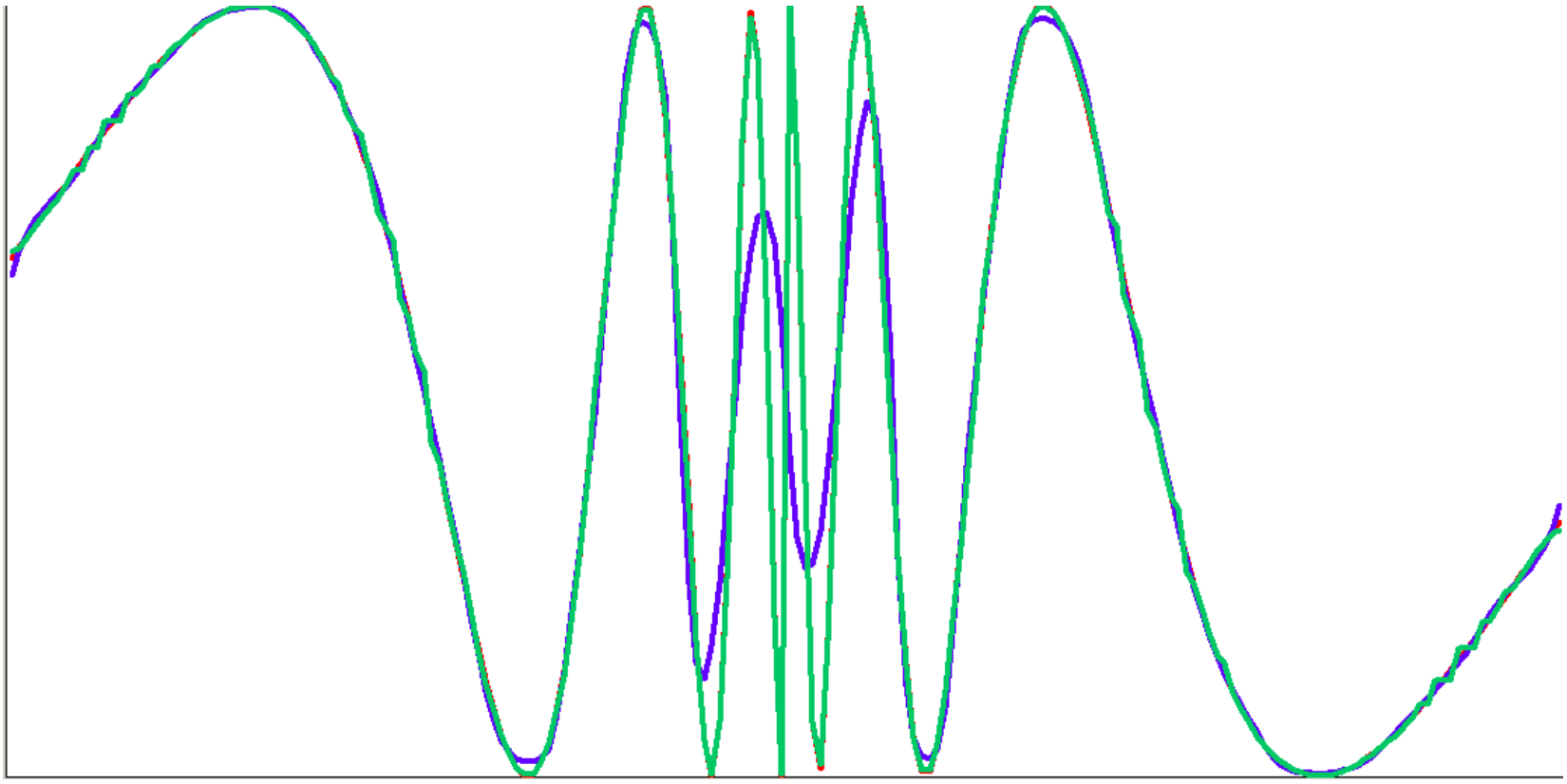
AdaBoost 100 итераций.

Boosting



AdaBoost 1000 итераций.

Boosting



AdaBoost 10000 итераций.

Понятие отступа

Иногда для обоснования бустинга вводится понятие отступа.

Отступ есть

$$\theta = \frac{y\beta(x)}{\varkappa}, \quad \varkappa = \sum_{t=1}^{T} \alpha_t.$$

Из-за нормировки в виде $\varkappa$ сложность композиции влияет на оценку риска.

Проклятие размерности

Для случая независимых переменных «проклятие» размерности превращается в преимущество:

- чем больше зависимых переменных — тем больше требуемый объём выборки,
- чем больше независимых переменных — тем меньше требуемая выборка.

С увеличением числа независимых переменных качество решения только растёт.

Бустинг и случайный лес

Методы существенно различаются в настройке.

- Для бустинга параметры сложности — глубина дерева и количество деревьев.
- Для random forest сложность не увеличивается с ростом числа деревьев.

Как правило, бустинг использует деревья меньшей глубины.

Как правило, бустинг достигает лучшего качества.

Выводы

- Важнейшей причиной эффективности бустинга является использование эффекта независимости (переменных, подпространств, моделей).
- Бустинг на пороговых классификаторах является разновидностью непараметрической логистической регрессии, также его можно считать разновидностью (существенно обобщённого) наивного байесовского классификатора.
- Бустинг реализует «удачный» вариант непараметрической аппроксимации условной вероятности.