Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Отчёт по заданию практикума №3

«Композиции алгоритмов».

Выполнил: Апишев М.А.

23 апреля 2014

Содержание

1 Постановка задачи **3**

2 Bagging **3**
2.1 Реализация 3
2.2 Классификация 3

3 Gradient Boosting **3**
3.1 Реализация 4
3.2 Классификация 5
3.3 Классификация с шумом 6
3.4 Регрессия 7

1 Постановка задачи

В рамках данного задания требовалось произвести исследование двух методов построения композий алгоритмов машинного обучения — Bagging и Gradient Boosting. Предлагалось решить задачи классификации и регрессии на наборах данных, предложенных в задании, сделать выводы из полученных результатов.

2 Bagging

2.1 Реализация

В Bagging композиция представляет собой набор алгоритмов, обучаемых независимо на подвыборках исходной обучающей выборки. Во время классификации производится голосование каждого элементарного классфикатора за какой-то класс, после чего ответом композиции объявляется класс, набравший максимальное число голосов.

В качестве базовых классификаторов в задании предлагалось использовать SVM (реализации из библиотеки libsvm) и Classification Tree (реализация из Matlab). В ходе каждой итерации обучения композиции из исходной выборки выбирается подвыборка того же размера, что и исходная.

2.2 Классификация

Были произведены следующие эксперименты:

1. Классификация с помощью SVM, композиция длиной 12 влгоритмов, два уровня сложности базового алгоритма[1].

2. Классификация с помощью Classification Tree, композиция длиной 12 влгоритмов, два уровня сложности базового алгоритма.

Графики результатов классификации на обучении и тесте приведены ниже[2]. Строились они по 5-fold кроссвалидации, ошибка — усреднение всех случаев.

Из графиков на рис.1 и рис.2 видно, что более простой классификатор в качестве базового подошёл лучше, нежели классификатор со сложной разделяющей поверхностью.

Из графиков на рис.1 и рис.2 видно, что для Bagging сложность дерева не играет большой роли. Также можно заключить, что для этого метода построения композиций SVM по качеству классификации на тесте обходит Classification Tree.

3 Gradient Boosting

В Gradient Boosting композиция строится таким образом, чтобы каждый последующий классификатор исправлял ошибку всей построенной ранее композиции, при этом сама она имеет следующий вид:

$$a(x) = b_0 + \sum_{i=1}^{n} \gamma_i b_i(x)$$

где n — число алгоритмов в композиции, γ_i — вес i-го классификатора.

[1] На самом деле, в коде присутствует три уровня сложности базовых алгоритмов и протестированны были все, но для краткости здесь описаны только два. На наглядности экспериментов это не сказалось.

[2] Точные значения параметров, соответствующих определённой сложности можно посмотреть в коде, здесь и далее в отчёте будут даваться качественные оценки

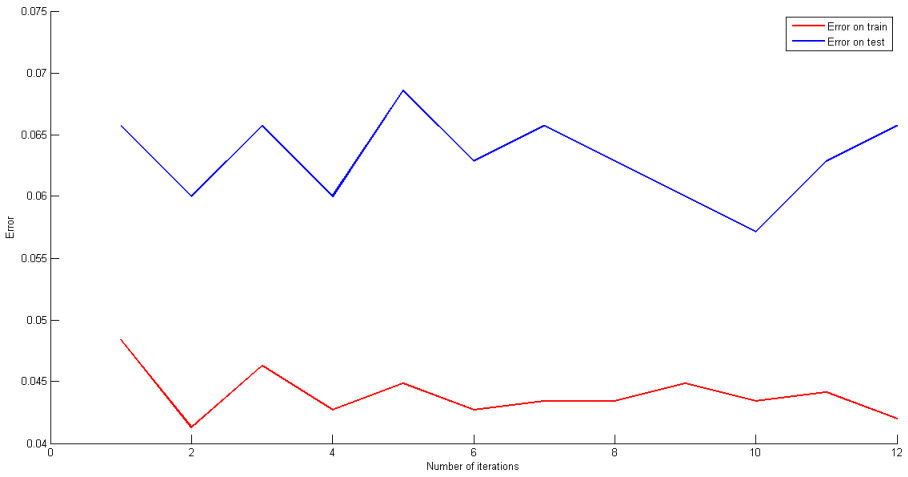

Рис. 1: SVM, простая разделяющая поверхность.

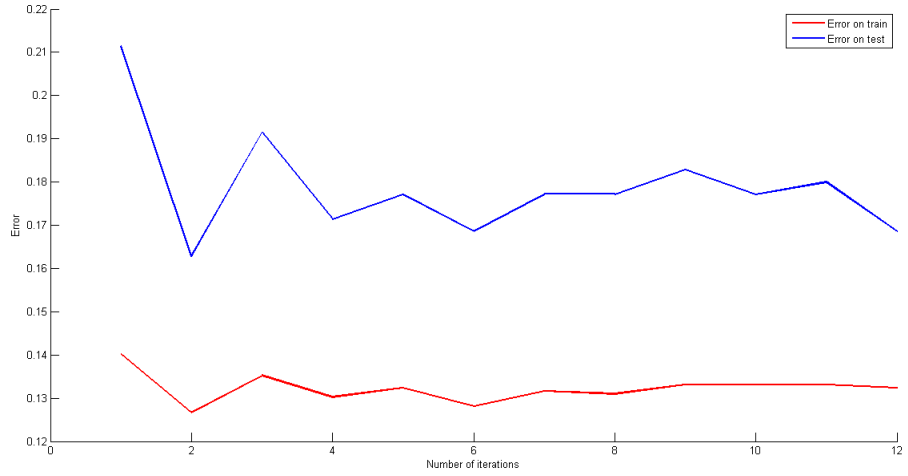

Рис. 2: SVM, сложная разделяющая поверхность.

Композиция, построенная с помощью boosting, подходит для решения как задач регрессии, так и классификации, «режим» зависит от выбора функции потерь.

3.1 Реализация

В данном задании для задач регресии предлагается использовать функцию $|f(x) - y|$ (absolute deviation), для задач классификации — $\log(1 + e^{-yf(x)})$ (logistic loss). Для реализации алгоритма Gradient Boosting требовались градиенты указанных функций потерь, они имеют следующий вид:

$$\frac{dF}{dx} = \frac{-ye^{-yx}}{1 + e^{-yx}}; \quad \frac{dF}{dx} = -sign(y - x);$$

для логистической функции и абсолютного отклонения соответственно.

В качестве базовых алгоритмов в использовались Epsilon-SVR (реализации из библио-

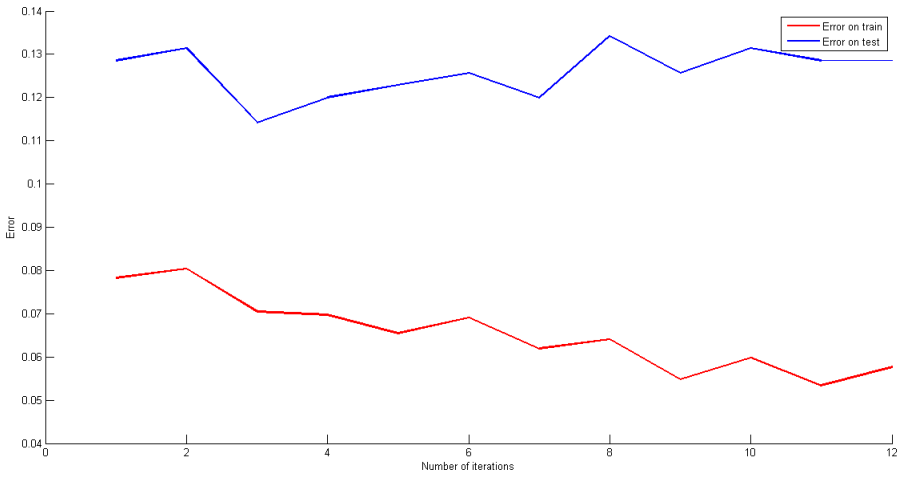

Рис. 3: Classification Tree, небольшая глубина.

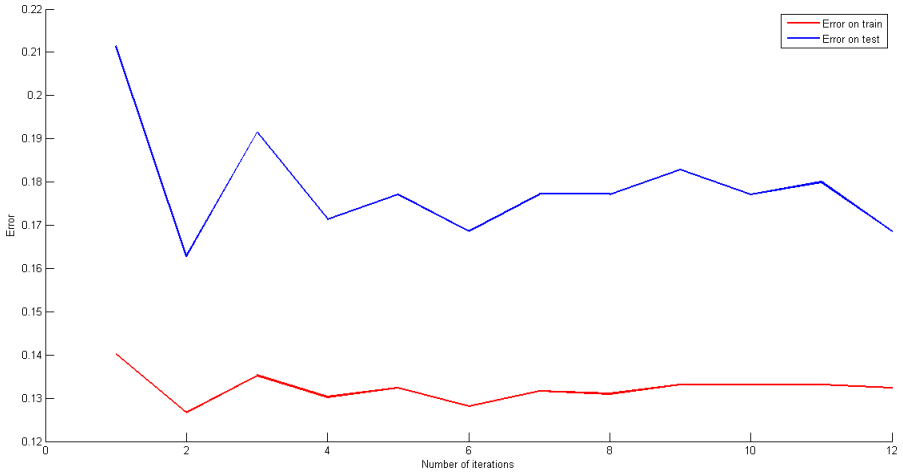

Рис. 4: Classification Tree, большая глубина.

теки libsvm) и Regression Tree (реализация из Matlab).

3.2 Классификация

Были произведены следующие эксперименты:

1. Классификация с помощью Epsilon-SVR, композиция длиной 12 влгоритмов, два уровня сложности базового алгоритма.

2. Классификация с помощью Regression Tree, композиция длиной 12 влгоритмов, два уровня сложности базового алгоритма.

Результаты экспериментов приведены на графиках ниже.

Как видно из рис.5 и рис.6, классификация композиции из Epsilon-SVR лучше при использовании простых разделяющих поверхностей базовых классификаторов.

Графики на рис.7 и рис.8 демонстрируют, что Regression Tree небольшой глубины опять-таки предпочтительней, нежели более сложные деревья. А использование самих

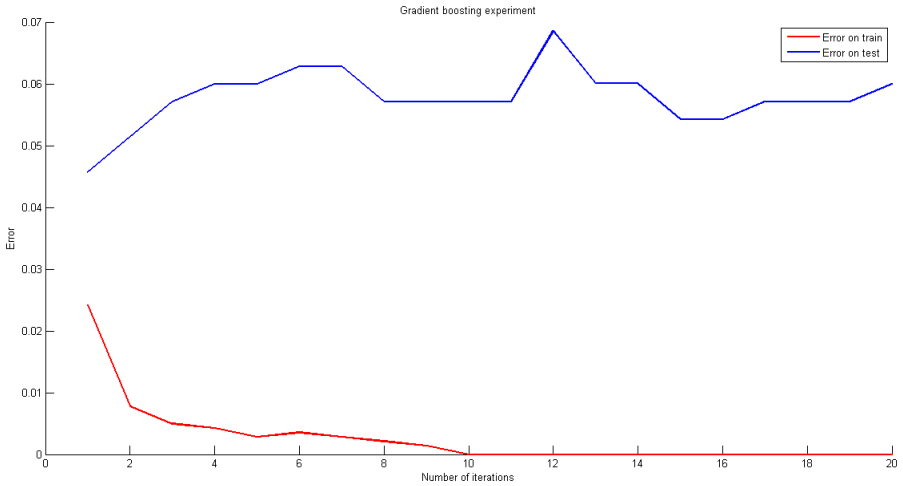

Рис. 5: Классификация Epsilon-SVR, простая разделяющая поверхность.

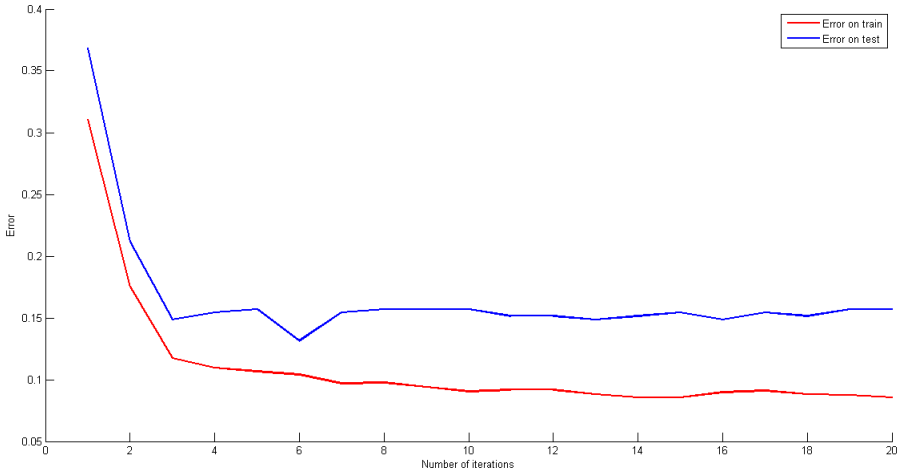

Рис. 6: Классификация Epsilon-SVR, сложная разделяющая поверхность.

деревьев в boosting даёт лучшие результаты, чем использование Epsilon-SVR, что так же можно увидеть на графиках.

3.3 Классификация с шумом

В рамках задания требовалось произвести эксперимент по классификации шумнах данных. У часть данных (около 20%) были инвертированы ответы, после чего на них был запущен boosting над решающими деревьями минимальной сложности. Композиция при этом начала переобучаться, что явно видно из рис.9. Для борьбы с этим были исползованы уменьшение параметра learning rate и досрочное завершение построения композиции[3] в алгоритме Gradient Boosting.

Результаты регуляризации можно наблюдать на рис.10 и рис.11. Оба метода дали неко-

[3]Останов можно было производить по отслеживанию роста ошибки, однако в работе он был сделан максимально простым способом — урезанием в три раза числа базовых алгоритмов композиции.

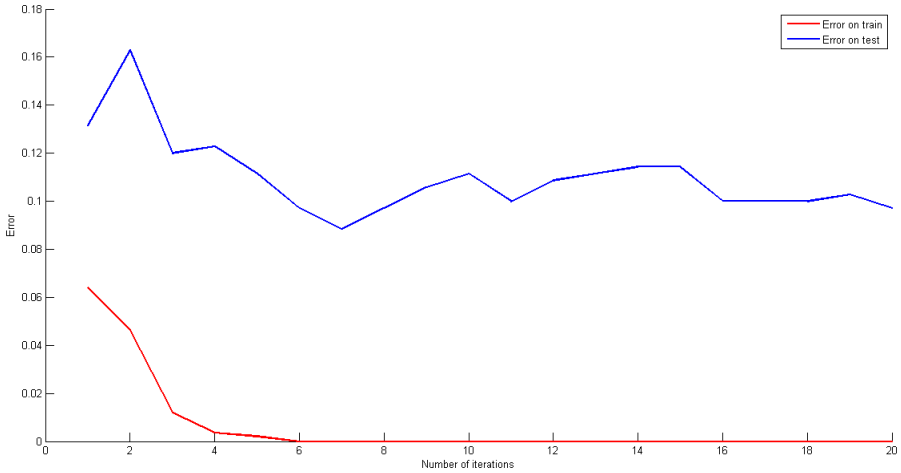

Рис. 7: Классификация Regression Tree, небольшая глубина.

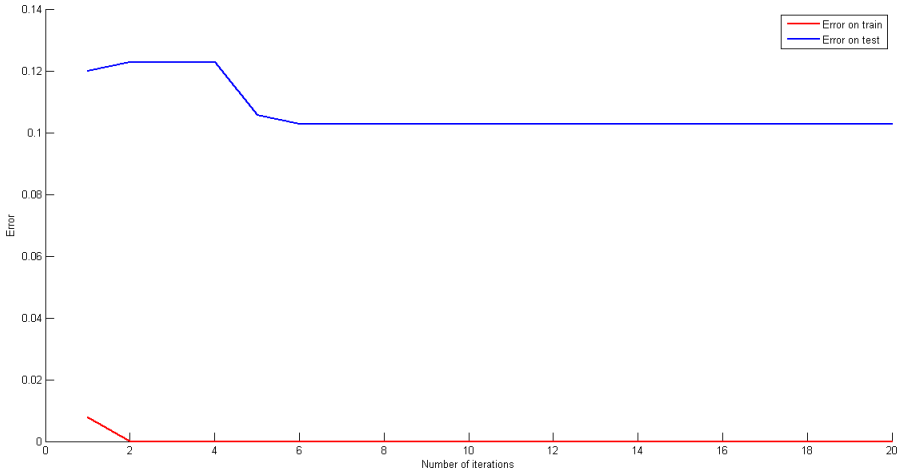

Рис. 8: Классификация Regression Tree, большая глубина.

торое улучшение качества классификации.

3.4 Регрессия

Заключительной частью задания являлось решение задачи регрессии с помощью Gradient Boosting над Regression Tree. При этом использовались деревья различной сложности, результаты приведены на рис.12 и рис.13. Оттуда сразу видно, что более простые деревья в качестве базовых алгоритмов оказались эффективнее деревьев со сложной структурой.

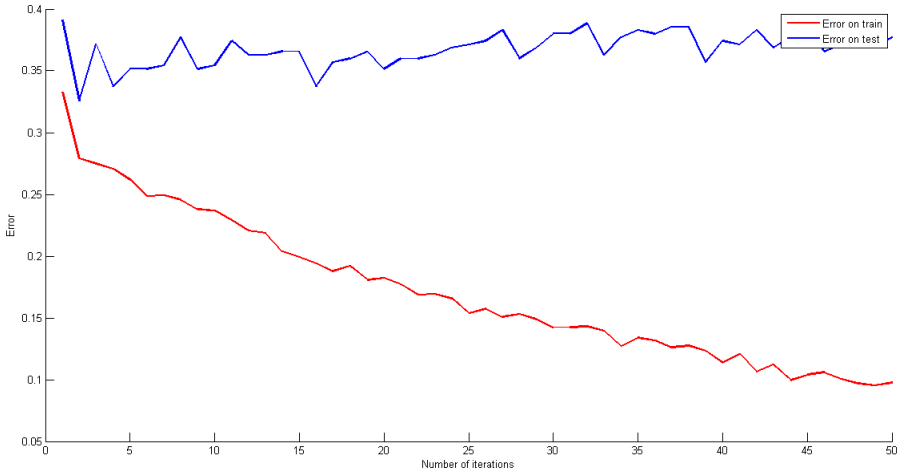

Рис. 9: Классификация Regression Tree, переобучение.

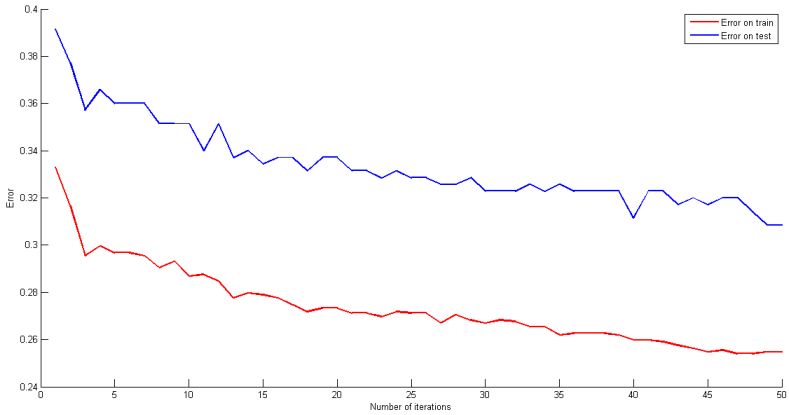

Рис. 10: Классификация Regression Tree, переобучение, регуляризация с помощью learning rate.

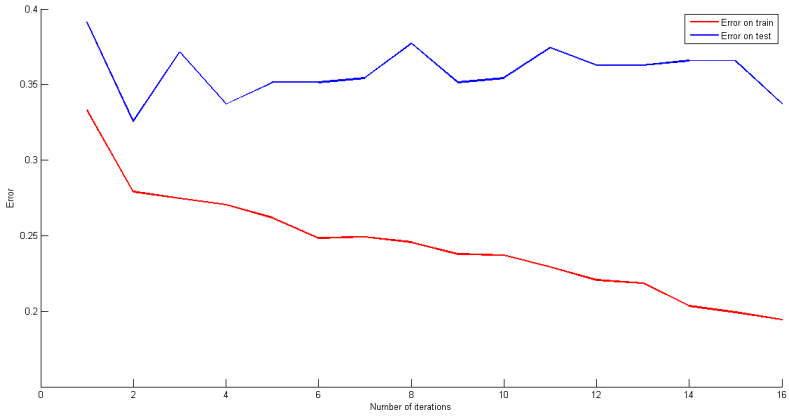

Рис. 11: Классификация Regression Tree, переобучение, регуляризация с помощью досрочного останова.

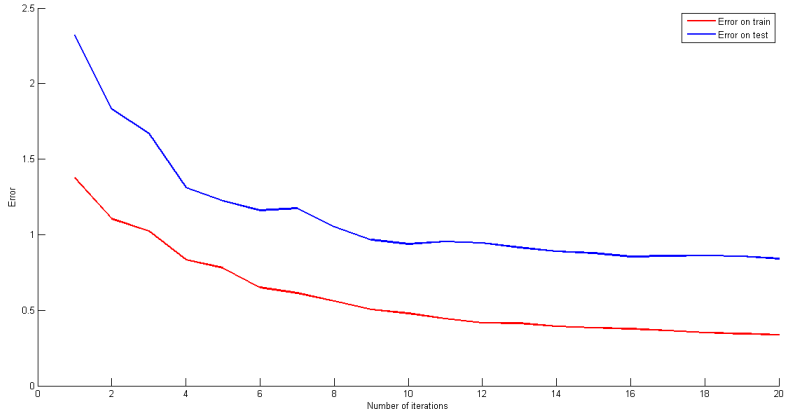

Рис. 12: Регрессия с помощью Regression Tree, простые деревья.

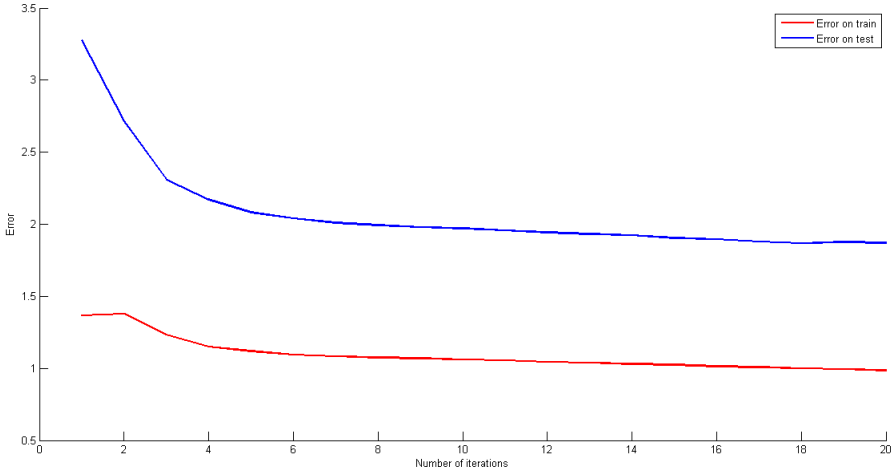

Рис. 13: Регрессия с помощью Regression Tree, сложные деревья.