Автоматизированное определение классов чувствительности веб-сервисов

Д. Н. Душкин, младший научный сотрудник, аспирант
(Федеральное государственное бюджетное учреждение науки
Институт проблем управления
им. В. А. Трапезникова Российской академии наук)

27 сентября 2013 г.

Аннотация

В работе представлена методика вычисления чувствительности — критерия сравнения веб-сервисов по производительности. Вводятся показатели, характеризующие чувствительность, описаны классы чувствительности, способ их получения и метод определения класса для произвольного веб-сервиса. В работе используются такие алгоритмы машинного обучения, как кластеризация с помощью алгоритма k-средних и классификация методом опорных векторов. Приводятся результаты вычислительного эксперимента, иллюстрирующего представленную методику.

Ключевые слова: *веб-сервисы, сервисно-ориентированная архитектура, машинное обучение, анализ чувствительности*.

УДК 004.032 (Характеристики систем)

ББК 3.32 (Радиоэлектроника. Вычислительная техника. Информатика)

Содержание

Введение

В последние несколько лет развития области информационных технологий (ИТ) наметился переход от всеобъемлющих информационных систем (ИС), включающих все необходимые для работы данные и функции, к распределенным системам, использующим внешние ресурсы [1]. Такой переход оправдан тем, что одна система зачастую уже не способна хранить весь объём необходимых данных и обеспечивать приемлемый уровень производительности вычислений. Одним из современных решений, позволяющих построить распределенную ИС, является использование сервисно-ориентированной архитектуры (СОА).

В основе сервисно-ориентированной архитектуры лежит принцип использования *веб-сервисов* — автономных вычислительных ресурсов, предоставляющих свои функции через сеть Интернет (или Интранет) посредством открытых протоколов обмена данными, не зависящих от платформ как самих ресурсов, так и связываемых с ними программных систем. Системы, использующие в своей работе веб-сервисы, называют системами с СОА.

В предыдущей авторской работе [2] проведен обзор и анализ существующих подходов выбора критериев сравнения веб-сервисов, подробно рассмотрен вопрос оценки чувствительности, дается формальное описание понятия чувствительности веб-сервисов, введены классы чувствительности и описана методика их выделения.

Основной мотивацией ввода нового критерия является отсутствие описания в современных научных публикациях таких критериев, которые описывали бы веб-сервис в долгосрочной перспективе, что важно в ситуациях, когда необходимо быть уверенным в обеспечиваемом уровне производительности при увеличении нагрузки. Определение такого критерия может быть полезно как при решении задачи рационального выбора архитектуры систем с СОА, так и при составлении документов регламентирующих соглашение об уровне предоставления услуги.

Ввод разделения веб-сервисов на классы мотивирован большим числом критериев, характеризующих чувствительность, и вариациями их возможных значений. В таком формирование экспертом наборов продукционных правил, помогающих выделять отдельные классы, является трудоёмкой задачей. Поэтому целесообразно использовать методы машинного обучения, которые хорошо подходят для решения задач в случае неприменимости точных алгоритмов.

В настоящей работе описывается усовершенствованный метод выделения классов чувствительности и метод автоматизированного определения класса чувствительности произвольного веб-сервиса с помощью алгоритма машинного обучения «с учителем» — метода опорных векторов (англ. *Support vector machine, SVM*).

Работа состоит из трех основных частей. В первой части описан усовершенствованный метод выделения классов чувствительности. Во второй части рассмотрен вопрос автоматизированного определения класса чувствительности произвольного веб-сервиса и приведены результаты вычислительного эксперимента. В заключительной части сделаны выводы по проведенной работе и обозначены перспективные направления исследований.

Исследования проведены при финансовой поддержке РФФИ в рамках научного проекта № 12-07-31214 мол_а.

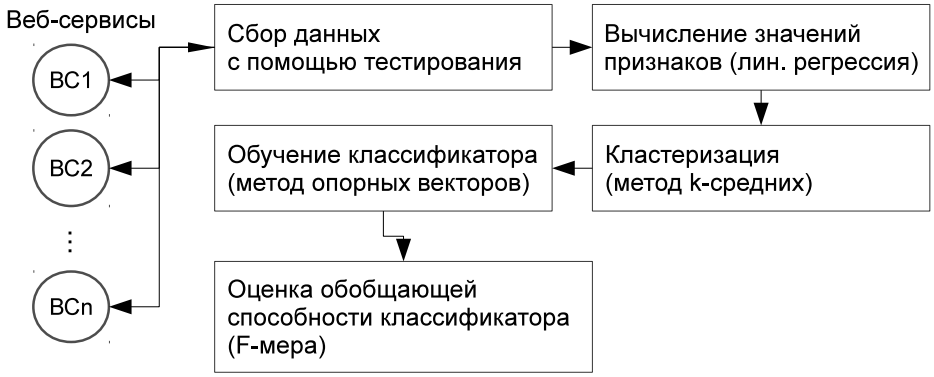


Рис. 1: Схематическая диаграмма предлагаемого комбинированного метода определения чувствительности

1 Выделение классов чувствительности

1.1 Основные определения

Чувствительность веб-сервиса — критерий, на основе которого может быть оценена возможность обеспечения определенного уровня производительности веб-сервиса при возрастающей нагрузке. Таким образом, для определения чувствительности необходимо произвести тестирование веб-сервиса по *определенному плану* и вычислить значения ряда признаков, характеризующих чувствительность, используя результаты теста. Поскольку значения признаков различных веб-сервисов имеют большой разброс, целесообразно ввести *классы чувствительности*, характеризуемые диапазонами значений признаков.

1.2 Формальная постановка задачи

Дано X — множество описаний веб-сервисов, Y — конечное множество классов чувствительности. Существует неизвестная целевая зависимость — отображение $y*\colon X \to Y$, значения которой известны только на объектах конечной обучающей выборки X^m и соответствующим им классам y^m. Требуется построить алгоритм $a\colon X \to Y$, способный классифицировать произвольный объект $x \in X$. Говоря менее формально, необходимо, используя предварительно обученный алгоритм классификации, присвоить класс чувствительности новому веб-сервису.

На рис. 1 представлено схематическое описание предлагаемого метода решения поставленной задачи.

Конечная обучающая выборка X^m и соответствующие классы y^m, вычисляются следующим образом:

1. Составляется случайная выборка веб-сервисов объемом m (раздел 1.3).
2. Вводится описание ряда признаков, характеризующих чувствительность веб-сервисов, и способов их вычисления (раздел 1.5).
3. Проводится тестирование выборки по определенному плану (раздел 1.4).
4. На основе результатов тестов вычисляется матрица «объект-признак» X^m.

5. При помощи визуализации матрицы X^m экспертом проводится эвристическое выделение схожих по значению показателей подмножеств веб-сервисов (раздел 1.6).

6. Формируются классы чувствительности на основе упорядоченных по предпочтению подмножеств веб-сервисов, при этом во вводимой градации классов число ошибок является более значимым критерием (раздел 1.6).

7. Составляется множество меток y^m принадлежности веб-сервисов X^m к определенным классам чувствительности.

Пункт 7 при достаточно большом объеме выборки является крайне трудозатратным, поэтому целесообразно автоматизировать этот процесс, проведя кластеризацию данных с числом кластеров, равным числу классов чувствительности. Также результаты кластеризации могут служить индикатором правильности эвристического разделения на классы: если полученные кластеры достаточно обособлены друг от друга, то классы выбраны верно. Кластеризация проводится методом k-средних [3].

Получив матрицу «объект-признак» X^m и множество соответствующих классов y^m, возможно обучить и оценить алгоритм классификации «с учителем». В качества такого алгоритма используется метод опорных векторов [4].

Далее поэтапно описывается весь процесс выделения классов чувствительности и организации автоматизированной классификации.

1.3 Формирование выборки

Для вычислительного эксперимента выборка была сформирована на основе данных из каталога API Directory [5], содержащего информацию о более чем 5000 различных веб-сервисах. Выбираются разнородные сервисы, реализующие функции картографии и геокодинга, предоставляющие информацию о различных показателях торговых бирж, о погоде, новостях и т.д. Все сервисы реализуют архитектуру REST [6], предоставляя свои функции по протоколу HTTP.

В работе используется выборка объемом $n = 100$ веб-сервисов.

1.4 План теста

Для вычисления признаков необходимо провести сбор необходимых данных посредством тестирования выборки веб-сервисов. В процессе тестирования осуществляется последовательное выполнение итераций. Одна итерация длится 1 секунду. Количество запросов, отправляемых в течение одной итерации, зависит от её номера (см. далее). Запросы равномерно распределены в рамках секунды. В среднем один тест длится 15-25 минут в зависимости от скорости ответа на запросы веб-сервисом. Каждый веб-сервис тестируется 3 раза с разницей между тестами в 9 часов, результаты проведенных тестов усредняются. Такой подход к тестированию позволяет снизить разброс значений, возникающий из-за различной загруженности веб-сервисов в течение дня.

Пусть λ_{max} — число запросов в секунду, отправляемое в последней итерации, S — шаг теста (число, на которое увеличивается количество запросов в секунду в последующей итерации). Тогда можно вычислить общее число итераций в тесте N_{iter}:

$$N_{iter} = \left\lceil \frac{\lambda_{max}}{S} \right\rceil . \tag{1}$$

Пусть $\bar{r}_{j,k}^{(i)}$ — усреднённое по трем тестам значение времени обработки запроса, где $i = 1, ..., n$ — номер веб-сервиса, n — объем выборки, $j = 1, ..., N_{iter}$ — номер итерации, $k = 1, ..., j \cdot S$ — номер запроса в рамках j-ой итерации.

В работе используется следующий план теста: максимальное число запросов в секунду $\lambda_{max} = 300$, шаг теста $S = 10$, объем выборки $n = 100$. Такие значения параметров выбраны экспериментально, т.к. было установлено, что начиная с 280-290 запросов в секунду большинство веб-сервисов демонстрируют устойчивое поведение. Выборка состоит из 100 веб-сервисов: Google Maps, Яндекс Карты, Bing Maps, Nokia Maps, Twitter, Factolex, Quora и др.

1.5 Вычисление матрицы «объект-признак»

Матрица «объект-признак» $\widetilde{X^m} \in \mathbb{R}^{m \times n}$, где m — число объектов, n — количество признаков, формируется на основе результатов тестов веб-сервисов. Ряды матрицы соответствуют объектам (веб-сервисам), столбцы — признакам. Обозначим $\mathbf{x}^{(i)}$ i-ый ряд матрицы X. Опишем каждый веб-сервис следующим образом:

$$\mathbf{x}^{(i)} = \left[\mathbf{t}^{(i)}, \mathbf{d}^{(i)}, \mathbf{e}^{(i)}\right], \tag{2}$$

где $\mathbf{t}^{(i)}$ — вектор, описывающий изменение среднего времени обработки запросов i-ым веб-сервисом при изменении нагрузки, $\mathbf{d}^{(i)}$ — вектор, описывающий изменение стандартного отклонения, $\mathbf{e}^{(i)}$ — вектор, описывающий изменение числа ошибок.

По каждому веб-сервису сначала вычислим среднее время обработки запросов, стандартное отклонение и число необработанных запросов по итерациям. Далее с целью устранения ограничения на однозначный план теста (шаг теста и максимальное число запросов в секунду) и более экономного описания характеристики чувствительности веб-сервиса произведем аппроксимацию значений признаков по итерациям полиномом первой степени с помощью алгоритма линейной регрессии [7]. Значения коэффициентов полученной регрессионной модели, а также суммарную квадратичную ошибку будем использовать в качестве признаков, характеризующих чувствительность веб-сервиса.

В дальнейших вычислениях среднего времени обработки и стандартного отклонения исключаются ошибочные запросы, т.е. такие запросы, которые не были обработаны в течение 30 секунд (время timeout) или веб-сервис вернул в ответе на запрос HTTP-код, отличный от «200 — OK».

Примем за $t_j^{(i)}$ среднее время обработки запросов i-ым веб-сервисом в рамках j-ой итерации:

$$t_j^{(i)} = \frac{1}{j \cdot S} \sum_{k=0}^{j \cdot S} \bar{r}_{j,k}^{(i)}. \tag{3}$$

Воспользуемся регрессионной моделью вида:

$$h_t(j \cdot S)^{(i)} = \theta_{t,0}^{(i)} + \theta_{t,1}^{(i)} \cdot (j \cdot S).$$

Параметр $\theta_{t,0}^{(i)}$ является показателем ординаты точки пересечения прямой с осью ординат, $\theta_{t,1}^{(i)}$ — тангенс угла наклона прямой. Необходимо подобрать такие параметры $\theta_t^{(i)}$, чтобы регрессионная модель наиболее точно описывала исходную зависимость среднего времени обслуживания запросов от числа запросов в секунду. Подбор параметров производится с помощью метода наименьших квадратов. Суть метода в минимизации значения стоимостной функции, равной

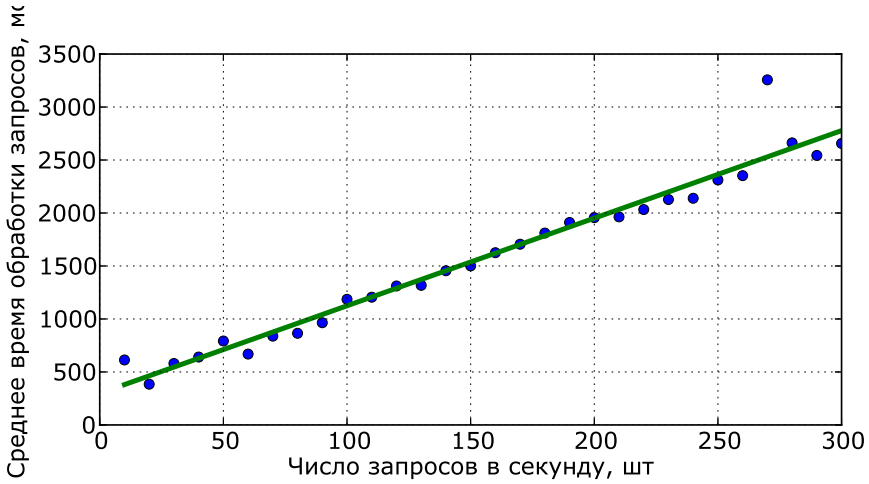


Рис. 2: Аппроксимация среднего времения обслуживания запросов от числа запросов в секунду. Точки — исходные данные, линия — аппроксимация.

разности между фактическим значением $t_j^{(i)}$ и вычисленным регрессионной моделью значением $h_t(x)$:

$$\min_{\theta_0,\theta_1} \quad J_t(\theta_t^{(i)}) = \sum_{j=1}^{N_{iter}} (t_j^{(i)} - h_t(j \cdot S)^{(i)})^2 \,. \tag{4}$$

С целью уменьшения диапазона возможных значений этого признака преобразуем тангенс угла наклона аппроксимирующей прямой, $\theta_{t,1}^{(i)}$, в градусы:

$$\alpha_t^{(i)} = arctan(\theta_{t,1}^{(i)}) \frac{180}{\pi} \,. \tag{5}$$

С целью вычисления относительных величин дополним стоимостную функцию постоянным делителем, равным максимальному числу запросов. В итоге, в качестве признаков, характеризующих изменение среднего времени обслуживания запросов при увеличении нагрузки, примем:

$$\mathbf{t}^{(i)} = \left[\alpha_t^{(i)}, \theta_{t,0}^{(i)}, \frac{J_t(\theta_t^{(i)})^{(i)}}{\lambda_{max}} \right] . \tag{6}$$

На рис. 2 синими кругами показаны усредненные по трем тестам данные веб-сервиса картографии «Яндекс.Карты», зеленой линией — аппроксимирующий полином. Видно, что в данном случае полином хорошо описывает тенденцию роста среднего времени обслуживания запросов при росте числа запросов в секунду, поэтому значение стоимостной функции $J(\theta_t^{(i)})$ будет мало. Значения признаков для примера на рис. 2: $\alpha_t^{(i)} = 83.1, \theta_{t,1} = 298.1, J_t(\theta_t)/\lambda_{max} = 80.9$.

Аналогично проведем вычисления для стандартного отклонения времени обработки запросов:

$$\begin{aligned} d_j^{(i)} &= \sqrt{\frac{1}{j \cdot S - 1} \sum_{k=0}^{j \cdot S} (\bar{r}_{j,k}^{(i)} - t_j^{(i)})^2} \,, \\ \mathbf{d}^{(i)} &= \left[\alpha_d^{(i)}, \theta_{d,0}^{(i)}, \frac{J_d(\theta_d^{(i)})^{(i)}}{\lambda_{max}} \right] . \end{aligned} \tag{7}$$

И для числа ошибок:

$$e_j^{(i)} = \sum_{k=0}^{j \cdot S} s_k, s_k = \begin{cases} 1 & \text{если запрос } \bar{r}_{j,k}^{(i)} \text{ не обработан} \\ 0 & \text{если запрос } \bar{r}_{j,k}^{(i)} \text{ обработан} \end{cases},$$
$$\mathbf{e}^{(i)} = \left[\alpha_e^{(i)}, \theta_{e,0}^{(i)}, \frac{J_e(\theta_e^{(i)})^{(i)}}{\lambda_{max}} \right]. \tag{8}$$

В итоге каждый веб-сервис описывается следующим набором признаков:

$$\mathbf{x}^{(i)} = \left[\alpha_t^{(i)}, \theta_{t,0}^{(i)}, \frac{J_t(\theta_t^{(i)})^{(i)}}{\lambda_{max}}, \alpha_d^{(i)}, \theta_{d,0}^{(i)}, \frac{J_d(\theta_d^{(i)})^{(i)}}{\lambda_{max}}, \alpha_e^{(i)}, \theta_{e,0}^{(i)}, \frac{J_e(\theta_e^{(i)})^{(i)}}{\lambda_{max}} \right].$$

Нормализуем полученные данные. Для этого зададим функцию нормализации z_j:

$$z_j : x_j^{(i)} \mapsto \frac{x_j^{(i)} - E_j[x_j^{(i)}]}{\mu_j(x_j^{(i)})}, \tag{9}$$

где E_j и μ_j— среднее и стандартное отклонение по j-ому признаку соответственно. Примем за $X^m = z(\widetilde{X^m})$ нормализованную по признакам матрицу «объект-значение».

1.6 Описание классов чувствительности

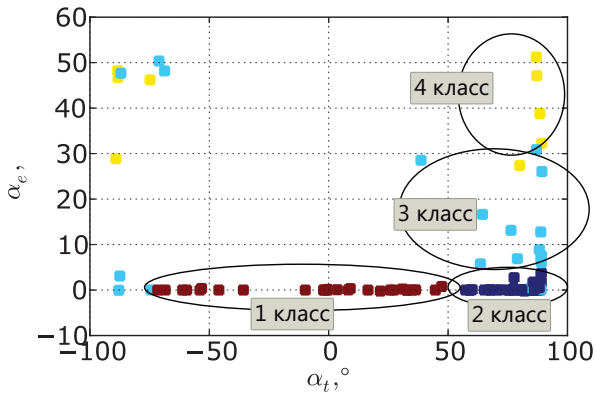
После проведения теста по заданному плану и вычисления матрицы «объект-признак» X^m можно визуализировать полученные данные. На рисунке 3 различными цветами обозначены различные классы чувствительности, полученные в результате этапа кластеризации. Овалами обведены *примерные* границы классов веб-сервисов. Проиллюстрировать точные границы на данном типе графиков невозможно ввиду изображения только 2 признаков на каждом графике, в то время как каждый веб-сервис характеризуется 9 показателями.

Введем эвристическое разделение веб-сервисов по классам чувствительности: от предпочтительной низкой чувствительности к высокой.

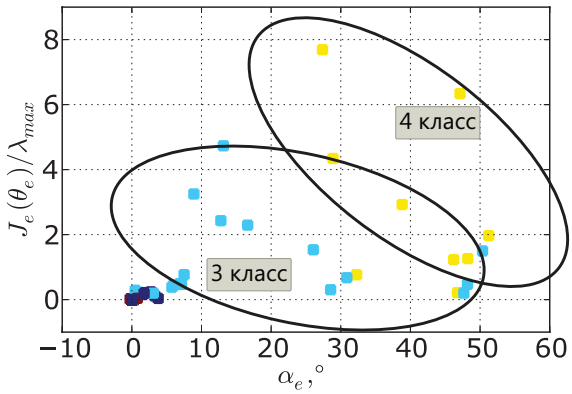
Первый класс — низкая чувствительность — характеризуется *небольшим изменением среднего времени обслуживания запросов и стандартного отклонения*, а также *отсутствием необработанных запросов*. На графике 3(а) можно увидеть, что ряд веб-сервисов имеют *отрицательное* значение признака α_t, угла аппроксимирующей прямой графика среднего времени обработки запроса, это означает, что начальное среднее время обработки запросов при малой нагрузке *больше* конечного. Такое поведение характерно для ряда «облачных» веб-сервисов (при увеличении нагрузки динамически увеличивается мощность обслуживающего узла) или веб-сервисов с адаптивным распределителем нагрузки (англ. *Load balancer*) (при высокой утилизации ресурсов одного обслуживающего узла часть запросов передается на обслуживание узлам с меньшей утилизацией).

Второй класс характеризуется *более быстрым повышением среднего времени обслуживания запросов, стандартного отклонения* и наличием *небольшого числа необработанных запросов*.

Третий класс характеризуется *быстрым повышением среднего времени обслуживания запросов и стандартного отклонения*, а также *большим количеством необработанных запросов*.



(a) Значения показателей α_t и α_e



(b) Значения показателей α_e и $J_e(\theta_e)/\lambda_{max}$

Рис. 3: Значения показателей выборки веб-сервисов, сгруппированных по кластерам

Четвертый класс — высокая чувствительность — характеризуется в первую очередь *наибольшим числом необработанных запросов*. При небольшом повышении нагрузки быстро растет число необработанных запросов, среднее время обработки запросов и стандартное отклонение.

1.7 Формирование множества классов с помощью кластеризации

С целью проверки обоснованности эвристического разделения веб-сервисов, а также для последующей автоматизации процесса определения класса чувствительности используется кластеризация данных с помощью алгоритма k-средних [3]. В качестве входных данных используются нормализованная матрица «объект-признак» X^m, 4 центроида (по количеству эвристически определенных классов), первоначальное положение центроидов выбирается случайно, всего проводится 200 итераций, в качестве меры расстояния используется манхэттенское расстояние:

$$d_{ij} = |x_i - x_j| \tag{10}$$

На рис. 3(a) и рис. 3(b) обозначены эвристически выделенные классы чувствительности. Принимая во внимая эвристическое описание классов чувствительности, можно графически обозначить классы на рисунке.

2 Определение класса чувствительности

2.1 Краткое описание метода опорных векторов

Решение задачи бинарной классификации при помощи метода опорных векторов заключается в поиске параметров функции, описывающей гиперплоскость, которая правильно разделяет набор данных на два класса. Гиперплоскость выбирается таким образом, чтобы расстояние от объектов разных классов до гиперплоскости было максимальным. Формально эту задачу оптимизации

с ограничениями можно записать следующим образом:

$$min_{\gamma,\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$
$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi_i, i = 1, ..., m$$
$$\xi_i \geq 0, i = 1, ..., m\ ,$$

где $\mathbf{x^{(i)}}$ — вектор признаков i-го объекта, $y^{(i)}$ — метка принадлежности i-го объекта к определенному классу, $\mathbf{w}$ — направляющий гиперплоскость вектор, b (от англ. «bias» — смещение) — кратчайшее расстояние между разделяющей гиперплоскостью и началом координат. ξ — дополнительные переменные, характеризующие величину ошибки. Если $\xi_i > 0$, то предполагается, что объект x_i может лежать внутри разделяющего зазора или ошибочно отнесен не к тому классу. Коэффициент C — коэффициент регуляризации, который позволяет изменять отношение между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки классификации.

Описанный метод подходит для случаев, когда возможно линейно разделить классы. Однако, на практике такие случаи встречаются нечасто. Для обеспечения возможности нелинейного разделения классов метод опорных векторов был дополнен ядрами (kernel trick). Суть дополнения заключается в переводе исходных объектов в расширенное пространство, где возможно линейное разделение на классы. В настоящей работе используется радиальная базисная функция Гаусса (далее РБФ):

$$K(x, x') = exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)\,.$$

Если необходимо разделить данные на большее число классов, применяют процедуру обучения нескольких классификаторов по принципу «один против всех». В этом случае каждый классификатор определяет гиперплоскость, разделяющую данные одного класса от всех остальных.

2.2 Выбор и оценка алгоритма классификации

В качестве алгоритма классификации используется метод опорных векторов. Необходимо:

1. Выбрать подходящую оценку обобщающей способности алгоритма (раздел 2.3).
2. Выбрать наилучшую комбинацию ядра алгоритма классификации и его параметров по показателям полноты и точности с помощью процедур скользящего контроля (кросс-валидации) и сеточного поиска (англ. *grid search*) (раздел 2.4).

2.3 Оценка обобщающей способности алгоритма

На практике многие классы чувствительности являются смещенными (англ. *scewed*), т.е. включают себя малое число объектов по сравнению с другими классами. Например, в выборке, используемой в настоящей работе, объемом 100 веб-сервисов к 3-ому классу относится только 4 объекта, в то время как к 1-ому — 40. Поэтому целесообразно в качестве оценки обобщающей способности алгоритма использовать не просто отношение верно классифицированных веб-сервисов к их общему числу, но оценки обобщающей способности по каждому классу. Такой оценкой может служить F-мера [8].

Введем основные определения. Пусть C_i — множество объектов, принадлежащих i-ому классу, u_i — множество объектов, которые алгоритм отнес к i-ому классу. Тогда полнотой (англ. *recall*) классификации объектов по i-ому классу является отношение количества объектов, правильно отнесенных к i-ому классу, к общему количеству объектов, относящихся к этому классу:

$$r_i(u_i) = \frac{|u_i \cap C_i|}{|C_i|}\,. \tag{11}$$

Точностью (англ. *precision*) классификации объектов по i-ому классу является отношение количества объектов, правильно отнесенных к i-ому классу, к общему количеству объектов, отнесенных к i-ому классу:

$$p_i(u_i) = \frac{|u_i \cap C_i|}{|u_i|}\,. \tag{12}$$

Идеальный алгоритм обеспечивает 100% полноту и точность.

Для удобства полноту и точность сводят к одной оценке, называемой, F-мерой (англ. *F-score, F_1-score*):

$$F_i(u_i) = 2 \times \frac{p_i(u_i) \times r_i(u_i)}{p(u_i) + r(u_i)}\,. \tag{13}$$

Для идеального классификатора F-мера равна 1, для худшего — 0.

F-меру по всем классам можно записать как:

$$F = \sigma \pm \mu\ , \tag{14}$$

где σ — среднее арифметическое, μ — среднеквадратическое отклонение по всем F_i.

2.4 Скользящий контроль и сеточный поиск

Скользящий контроль — процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам («с учителем») [9]. В общем случае исходная выборка разбивается на множество сочетаний двух подвыборок: обучающей и контрольной. Для каждого разбиения выполняется настройка алгоритма на обучающей подвыборке, затем оценивается ошибка на контрольной подвыборке. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

В настоящей работе используется k-кратный скользящий контроль (англ. *k-fold cross-validation*), который заключается в следующем: выборка случайным образом разбивается на k непересекающихся подмножеств одинаковой (или почти одинаковой) длины. Процедура кросс-валидации начинается с выбором первого подмножества как контрольного, оставшиеся $k-1$ подмножеств используются как обучающая выборка. Процедура повторяется k раз, при этом каждое подмножество по одному разу используется как контрольное. k результатов оценки точности и полноты алгоритма усредняются.

Дополним k-кратный скользящий контроль техникой сеточного поиска (англ. *grid search*) [10]. Сеточный поиск — достаточно простой, но эффективный метод оптимизации модели алгоритма и значений параметров. Задаются множества возможных моделей и значений параметров и с помощью полного перебора возможных сочетаний выбирается такое сочетание, для которого оценка обобщающей способности (в нашем случае это F-мера) по итогам скользящего контроля будет наивысшей.

2.5 Вычислительный эксперимент

Проведем процедуры скользящего контроля и сеточного поиска с целями выбора наилучшей комбинации параметров и оценки обобщающей способности алгоритма.

Разделим случайным образом исходную выборку на два подмножества: 80% — подмножество, используемое в процедурах сеточного поиска и скользящего контроля, 20% — контрольное подмножество.

Зададим $k = 3$ для процедуры k-кратного скользящего контроля.

Зададим множество параметров, среди которых необходимо найти наилучшую комбинацию с помощью метода сеточного поиска и кросс-валидации.

1. Для линейного ядра: $C = [1, 10, 100, 1000]$.

2. Для радиальной базисной функции: $\gamma = [0.001, 0.0001], C = [1, 10, 100, 1000]$.

Используя библиотеку с открытым исходным кодом, содержащую множество алгоритмов машинного обучения, Scikit Learn [11], осуществим вычислительный эксперимент с заданными параметрами. Результаты сеточного поиска и кросс-валидации (в качестве оценки используется средняя F-мера по всем классам) показаны в таблице 1.

Таблица 1: Результаты сеточного поиска и кросс-валидации

F-мера	Ядро	Параметры ядра
0.303 ± 0.010	РБФ	$C = 1, \gamma = 0.001$
0.303 ± 0.010	РБФ	$C = 1, \gamma = 0.0001$
0.470 ± 0.010	РБФ	$C = 10, \gamma = 0.001$
0.303 ± 0.010	РБФ	$C = 10, \gamma = 0.0001$
0.799 ± 0.019	РБФ	$C = 100, \gamma = 0.001$
0.470 ± 0.010	РБФ	$C = 100, \gamma = 0.0001$
0.862 ± 0.013	РБФ	$C = 1000, \gamma = 0.001$
0.799 ± 0.019	РБФ	$C = 1000, \gamma = 0.0001$
0.862 ± 0.022	Линейное	$C = 1$
0.848 ± 0.013	Линейное	$C = 10$
0.787 ± 0.013	Линейное	$C = 100$
0.787 ± 0.013	Линейное	$C = 1000$

Лучший набор параметров наблюдается для ядра РБФ с параметрами $C = 1000, \gamma = 0.001$. Проведем оценку классификатора с заданными параметрами на контрольном подмножестве. Стоит отметить, что результаты сеточного поиска и скользящего контроля, а также оценка F-меры на контрольном подмножестве зависят от первоначального случайного разбиения всей выборки на подмножества, поэтому конечное значение общей F-меры может варьироваться. После нескольких повторений всей процедуры обучения классификатора оценка на контрольном подмножестве F-мера $= 0.955 \pm 0.025$.

Полученные результаты можно представить визуально. Для этого уменьшим размерность пространства признаков с 9 до 2 с помощью метода главных компонент [12], настроим параметры классификатора, используя полученное пространство признаков и построим границы принятия решений по классам (рис. 4).

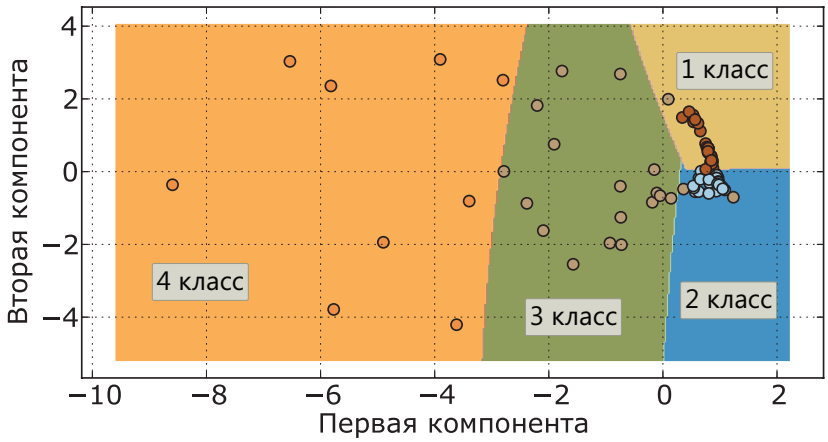


Рис. 4: Границы принятия решения классификатора по классам

Методика определения класса чувствительности веб-сервиса демонстрирует определенную устойчивость. В результате вычислительного эксперимента каждый веб-сервис проходил тестирования 3 раза с разницей в 9 часов между тестами. В дальнейшем для проверки гипотезы устойчивости метода каждый результат теста использовался как входные данные для определения чувствительности. 83% всех веб-сервисов показывают одинаковое значение чувствительности за все три измерения, остальные 17% веб-сервисов только единожды изменяют класс, т.е. присутствовали последовательности классов вида $1\ 2\ 1,3\ 2\ 2$ и др.

3 Выводы

В работе проведен анализ и предложено решение проблемы получения численной характеристики чувствительности веб-сервиса. Показана практическая возможность эффективного выделения классов чувствительности, приведено эвристическое описание классов. Сформирована теоретическая и алгоритмическая основа программного комплекса для определения класса чувствительности произвольного веб-сервиса.

Преимуществами предложенного подхода являются независимость показателей от выбранного шага тестирования и невысокое начальное значением максимального числа запросов в секунду. Таким образом методика может быть эффективна в отсутствии возможности полноценного тестирования веб-сервиса при высокой нагрузке.

Видится перспективным продолжение работ в данном направлении. На данный момент отсутствуют методики и программное обеспечение, которого могло бы провести полноценную оценку веб-сервиса и дать рекомендации по набору используемых веб-сервисов в системе с сервисно-ориентированной архитектурой при наличии функциональных и нефункциональных требований и предпочтений. Поэтому, как правило, при решении таких вопросов руководствуются экспертными рекомендациями веб-сервисов в виду отсутствия каких-либо строгих методик оценки.

Список литературы

[1] Wirsing Martin, Hölzl Matthias, Koch Nora, Mayer Philip. SENSORIA — Software Engineering for Service-Oriented Overlay Computers. — 2011.

[2] Душкин Д.Н. Анализ чувствительности веб-сервисов в задаче выбора оптимальной конфигурации систем с сервисно-ориентированной архитектурой // Управление большими системами. — 2012. — Т. 40. — С. 164–182. — URL: http://ubs.mtas.ru/archive/search_results_new.php?publication_id=18929.

[3] MacQueen J B. Some Methods for Classification and Analysis of MultiVariate Observations // Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability / Ed. by L M Le Cam, J Neyman. — Vol. 1. — University of California Press, 1967. — P. 281–297.

[4] Vapnik Vladimir N. The nature of statistical learning theory. — New York, NY, USA : Springer-Verlag New York, Inc., 1995. — ISBN: 0-387-94559-8.

[5] Programmable Web. API Directory. http://www.programmableweb.com/apis/directory. — 2012.

[6] Fielding Roy Thomas. Architectural styles and the design of network-based software architectures : Ph. D. thesis / Roy Thomas Fielding. — University of California, Irvine, 2000.

[7] В.В. Стрижов. Методы выбора регрессионных моделей. — Москва: ВЦ РАН, 2010. — С. 60. — URL: http://www.machinelearning.ru/wiki/images/5/52/Strijov-Krymova10Model-Selection.pdf.

[8] Rijsbergen C J Van. Information Retrieval. — 2nd edition. — Newton, MA, USA : Butterworth-Heinemann, 1979. — ISBN: 0408709294.

[9] Воронцов К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики / Под ред. О.Б. Лупанов. — Москва : Физматлит, 2004. — Т. 13. — С. 5–36. — URL: www.ccas.ru/frc/papers/voron04mpc.pdf.

[10] Bergstra James, Bengio Yoshua. Random Search for Hyper-Parameter Optimization // J. Mach. Learn. Res. — 2012. — Vol. 13. — P. 281–305. — URL: http://dl.acm.org/citation.cfm?id=2188385.2188395.

[11] Scikit-learn: Machine Learning in Python / F Pedregosa, G Varoquaux, A Gramfort et al. // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.

[12] Pearson K. On lines and planes of closest fit to systems of points in space // Philosophical Magazine. — 1901. — Vol. 2, no. 6. — P. 559–572.