

Санкт-Петербургский государственный университет
Кафедра технологии программирования

Зайцева Ольга Олеговна

Магистерская диссертация

Методы классификации в задачах
моделирования наивной квалификации в
теории уголовно-правовой оценки деяния

Направление 02.04.02

Фундаментальная информатика и информационные технологии

Магистерская программа «Технологии баз данных»

Научный руководитель,
кандидат тех. наук,
доцент
Блеканов И.С.

Санкт-Петербург

2017

Содержание

Введение	3
Актуальность работы	3
Цель работы	5
Задачи	6
Глава 1. Обзор существующих средств и методов	6
Глава 2. Основы уголовного права России	10
Глава 3. Архитектура программного комплекса наивной квалификации в теории уголовно-правовой оценки деяния	16
3.1. Процесс построения семантического шаблона для составов преступлений УК РФ	17
3.2. Теоретический алгоритм работы	22
Глава 4. Задача классификации	25
4.1. Постановка задачи классификации	25
4.2. Обзор существующих методов решения задачи классификации с пересекающимися классами	27
4.3. Метод опорных векторов	29
4.3.1. Классический метод опорных векторов	29
4.3.2. Выбор ядра	35
4.3.3. Метод опорных векторов для многоклассовой классификации с пересекающимися классами	37
4.3.4. Преимущества и недостатки метода опорных векторов	38

4.4. ML–kNN — модификация метода k ближайших соседей для многоклассовой классификации с пересекающимися классами	39
4.4.1. Преимущества и недостатки ML–kNN	44
4.5. Метрики качества многоклассовой классификации с пересекающимися классами	44
Глава 5. Эксперимент и апробация	49
5.1. Постановка эксперимента	49
5.2. Подготовка обучающего множества для задачи классификации	50
5.3. Результаты эксперимента	51
Заключение	53
Список литературы	55

Введение

Актуальность работы

Сегодня практически нет такого вида человеческой деятельности, где бы в той или иной мере не использовались компьютеры. Благодаря внедрению во второй половине XX века в повседневную жизнь компьютерной техники, а также специализированного программного обеспечения возросла эффективность правоприменительной практики. Чем новее и точнее применяемые в юридической деятельности средства и методы, тем быстрее решаются такие задачи уголовного судопроизводства, как раскрытие и расследование преступлений.

В настоящее время в юридической практике используются нейросетевой детектор лжи, нейросеть-антихакер и другие технологии, однако систем, которые помогли бы дознавателям, следователям, прокурорам и судьям оценить преступное деяние, предотвратить планируемые преступления на данный момент в России не существует.

Правовая система Российской Федерации (РФ) относится к романо-германской правовой семье, в которой основным источником права является закон (нормативно-правовой акт), в отличие от англо-американской правовой системы, составной частью которой является прецедентное право, т. е. прецедент — решение суда по определенному делу имеет силу источника права.

Одна из основных отраслей права в гражданском обществе — это уголовное право. Ст. 1 ч. 1 Уголовного кодекса (УК) РФ гласит: «Уголовное законодательство Российской Федерации состоит из настоящего Кодекса. Новые законы, предусматривающие уголовную ответственность, подлежат включению в настоящий Кодекс». Уголовный Кодекс представляет собой систематизированное изложение норм уголовного права. Он состоит из двух частей: Общей и Особенной, объединяющих 12 разделов, 34 главы и 360 статей.

Одной из существенных частей правоприменительной практики является осуществление квалификации преступлений. В своей деятельности работники правоохранительных органов постоянно сталкиваются с необходимостью квалифицировать совершенное тем или иным лицом общественно опасное деяние [1]. Квалифицировать преступление — значит дать ему юридическую оценку, установить соответствие между определенным деянием и признаками того или иного состава преступления, определить статью уголовного закона, предусматривающую наказание за совершенное преступление [2].

На данный момент существует большое количество попыток создать систему для анализа преступных деяний. Ключевым компонентом данных комплексов является применение методов интеллектуального анализа данных (с английского Data mining). В каждой из существующих систем используются различные методы, технологии, подходы анализа данных, а также различные правила и эвристики определения важной информации, связанной с преступлениями.

Семантический анализ и анализ текста используются для извлечения сущностей из материалов дела. Основанные на правилах системы созданы благодаря знанию предметной области, однако данные системы имеют ограничения в связи с динамическим характером преступлений. Кластеризация, классификация и визуализация с помощью графов помогают выявить схожие преступления и представить пользователю полученную информацию в удобном и понятном виде.

В виду того, что в России право не прецедентное, т. е. использовать методы латентно-семантического анализа текста для поиска по дубликатам и квалификации преступных деяний не имеет смысла, следовательно, для решения данной задачи согласно структуре права РФ, сначала необходимо произвести ручную семантическую разметку нормативно-правового акта, выделить основные характеристики всех составов преступлений. В данной работе основное внимание уделяется уголовному праву и Уголовному кодексу РФ.

Цель работы

Разработать модель процесса анализа и обработки данных в рамках процессуальных действий уполномоченных лиц, конечным результатом которого является выделение конкретных составов преступлений, предусмотренных действующим законодательством (УК РФ).

Задачи

Для успешного достижения поставленной цели необходимо:

- исследовать существующие методики анализа и оценки преступных деяний;
- изучить структуру права РФ, в особенности Уголовный кодекс РФ;
- разработать архитектуру информационного комплекса;
- произвести ручную семантическую разметку главы 16 УК РФ;
- изучить методы интеллектуального анализа данных, решающие задачу классификации;
- подобрать и апробировать методы классификации в рамках наивной квалификации преступлений.

Глава 1. Обзор существующих средств и методов

Информация играет важнейшую роль в таких ключевых областях, как правоохранительная деятельность. Большой объем данных, исследуемых правоохранителями, создает ряд проблем в разных областях, например, хранение, анализ этих данных. Множество технологических усилий в настоящий момент направлено на решение данных задач.

В существующих программных средствах используются различные подходы и технологии решения поставленной задачи, но, несмотря на это, можно заметить, что все они основаны на методах интеллектуального анализа данных. По характеру решаемых задач данные комплексы можно разделить на [3]:

- анализ преступлений, основанный на геополитической обстановке, и визуализация полученных результатов;
- кластеризация и прогнозирование преступлений в зависимости от географических характеристик [4, 5];
- выявление серийных преступников [6];
- анализ преступности в социальных сетях;
- выявление преступных аномалий (например, мошенничества [7]);
- анализ временных закономерностей преступлений.

Рассмотрим некоторые примеры существующих программных средств.

Авторы статьи [8] используют совокупность методов классификации для прогнозирования места, времени и вероятности планируемых преступлений (на примере квартирных краж). Набор рассматриваемых авторами данных содержит информацию о пространственных и временных характеристиках преступлений в США. Используя эти данные и различные методы классификации (метод ближайшего соседа, деревья решений,

метод опорных векторов, 2-слойные нейронные сети и наивный байесовский классификатор), исследователи анализируют, какой метод потенциально является лучшим для решения поставленной задачи. Наряду с прогнозированием преступлений, авторы оценивают уровень преступности в конкретных районах. Экспериментальные результаты показывают, что комбинация методов классификации и различных характеристик преступлений является наилучшим подходом к прогнозированию преступных деяний. Кроме того, данные исследования представляют собой ценную информацию о том, как организуются и совершаются преступления.

Задача выявления моделей поведения серийных преступников и преступной деятельности с помощью нечетких ассоциативных правил рассматривается в [9]. Изучив набор входных данных, авторы выделили 40 атрибутов и, используя полученные значения, создали правила для выявления таких последствий, как грабеж, убийство и разбойное нападение. С помощью вычисления относительной поддержки правил (с английского *relative support*) были извлечены редкие, наиболее интересные правила. Такой подход избавляет сотрудников правоохранительных органов от необходимости исследовать и отсеивать неинтересные и очевидные правила, чтобы найти интересные и значимые закономерности.

Дональд Браун (Donald E. Brown) в своей статье [10] описывает созданный им программный продукт *The Regional Crime Analysis Program (ReCAP)*, который помогает экспертам-криминалистам производить анализ и предотвращать преступ-

ления. Данная утилита использует данные системы PISTOL Records Management System 2000, которая собирает, хранит и предоставляет удобный доступ ко всей информации, полученной правоохранительными органами. Научные исследования и разработки автора были в первую очередь сосредоточены на следующих компонентах системы:

- база данных;
- географическая информационная система (ГИС);
- инструменты интеллектуального анализа данных.

Вышеперечисленные компоненты выполняют следующие функции:

- база данных хранит всю необходимую информацию о преступных деяниях;
- запросы к базе данных помогают получить всю интересующую информацию в удобном для пользователя виде — информацию о наиболее опасных криминальных активностях в регионе, отчеты о преступлениях за определенный промежуток времени или в конкретном районе;
- инструменты ГИС анализируют и визуализируют пространственные данные преступлений — цифровая карта местности с нанесенной информацией о преступных активностях;
- инструменты, основанные на методах интеллектуального анализа данных, помогают анализировать информацию об

уровне преступности в определенных районах с помощью контрольных карт Шухарта, кластерного анализа и прогнозирования.

В целом, можно отметить, что существующие средства направлены на предотвращение преступлений и борьбу с преступностью. Программные продукты избавляют работников правоохранительных органов от рутинной работы.

Глава 2. Основы уголовного права России

Перед непосредственным семантическим анализом Уголовного кодекса РФ и апробацией методов наивной квалификации преступных деяний необходимо изучить основы уголовного права и структуру Уголовного кодекса РФ.

Уголовное право — это одна из самостоятельных отраслей права в гражданском обществе. Оно включает в себя [11]:

- понятие и виды преступлений;
- цели и систему наказаний за совершенные преступления, общие начала и правила их назначения;
- условия освобождения от уголовной ответственности и наказаний.

Согласно Уголовному кодексу РФ (ст. 2 УК РФ) задачами уголовного права являются «охрана прав и

свобод человека и гражданина, собственности, общественного порядка и общественной безопасности, окружающей среды, конституционного строя Российской Федерации от преступных посягательств, обеспечение мира и безопасности человечества, а также предупреждение преступлений». Для выполнения данных задач Уголовный кодекс «устанавливает основание и принципы уголовной ответственности, определяет, какие опасные для личности, общества или государства деяния признаются преступлениями, и устанавливает виды наказаний и иные меры уголовно-правового характера за совершение преступлений».

Данный нормативный акт является единственным источником уголовного права в Российской Федерации. Он включает в себя две части — Общую и Особенную:

- Общая часть состоит из 6 разделов и включает 16 глав, статьи 1 — 104. Ее нормы отражают основные положения уголовного права, определяют понятие преступления, вины, наказания, соучастия в преступлении, лиц, подлежащих уголовной ответственности.
- Особенная часть включает 6 разделов, 19 глав и статьи 105 — 360. В данной части описаны конкретные составы преступлений и наказания за их совершение. Изучив структуру Особенной части, становится понятен приоритет уголовно-правовой охраны России. На первое место ставятся преступления против личности, а затем уже, соответственно, преступления в сфере экономики,

против общественной безопасности и общественного порядка, против государственной власти, против военной службы и против мира и безопасности человека.

Многие статьи Уголовного кодекса делятся на части, которые имеют цифровое обозначение. Части статьи, в свою очередь, могут делиться на пункты, имеющие буквенное обозначение. Например, статья 105 УК РФ имеет 2 части, в части 2 имеются пункты а) — н).

В соответствие со ст. 8 УК РФ «основанием уголовной ответственности является совершение лицом деяния, содержащего все признаки состава преступления, предусмотренного Уголовным кодексом». Признаки конкретного состава преступления содержатся в Общей и в Особенной частях УК РФ. Состав преступления образуют четыре основных элемента:

- объективные элементы:
 - объект — то, на что направлено преступное деяние и чему в результате причиняется или может быть причинен вред,
 - объективная сторона — внешняя сущность посягательства;
- субъективные элементы:
 - субъект — лицо, совершившее преступное деяние,
 - субъективная сторона — внутреннее отношение субъекта к содеянному.

Признаки состава преступления — это конкретные составляющие элементов состава преступления. Они делятся на обязательные, которые всегда присутствуют в составе преступления, и дополнительные — обязательные для некоторых конкретных составов (см. табл. 1.).

Элемент состава преступления	Обязательные признаки	Дополнительные признаки
Объект	Общественные отношения, подвергшиеся посягательству	Предмет преступления
		Потерпевший
		Факультативный объект
Объективная сторона	Общественно опасное деяние (действие или бездействие)	Место
		Время
	Общественно опасное последствие	Орудие
		Средство совершения
Причинная связь	Способ	
Субъект	Физическое лицо	Специальный субъект
	Вменяемость	
	Возраст	
Субъективная сторона	Вина	Мотив
	Умысел или неосторожность	Цель
		Эмоции

Таблица 1: Признаки состава преступления

Отсутствие хотя бы одного вышеназванного элемента равносильно отсутствию всего состава преступления, а, следовательно, и преступного деяния.

В уголовном праве совместно с объектом преступления рассматривают предмет преступления. Предмет преступления — это вещь, элемент материального мира, воздействуя на который, субъект наносит вред объекту. Зачастую понятия объект и предмет преступления не различают, но они имеют разное значение. Объект преступления — это общественные отношения, блага, интересы и ценности, которым причиняется вред.

Предмет — это то, по поводу чего это отношение возникло [11].
Предмет преступления всегда материален.

Как было написано выше, охрана личности находится на первом месте в приоритете уголовно-правовой охраны РФ. Преступления против личности — это «общественно опасные деяния, предусмотренные уголовным законом и непосредственно посягающие на безопасность жизни, здоровья, свободу, честь и достоинство, половую неприкосновенность, конституционные права и свободы человека и гражданина, интересы семьи и несовершеннолетних» [12]. Согласно гл. 2 Конституции РФ (ст. 20 — 23) каждый человек имеет право на жизнь, охрану достоинства личности, свободу и личную неприкосновенность, а также неприкосновенность частной жизни, личную и семейную тайну, защиту своей чести и доброго имени. Раздел VII УК РФ посвящен преступлениям против личности. Он включает в себя 5 глав, 53 статьи (105 — 157) и описывает составы преступлений, объектом которых являются те или иные блага и интересы человека.

Особое внимание я хочу уделить главе 16 УК РФ — преступления против жизни и здоровья (ст. 105 — 125). Преступления против жизни — это общественно-опасные деяния, объектом которых является жизнь человека. Ясно, что сам человек выступает в качестве предмета преступления в данном случае. Результат таких преступлений — смерть человека. Все преступные деяния, посягающие на жизнь, можно разделить на 3 категории:

- убийства (ст. 105 — 108 УК РФ);
- причинение смерти по неосторожности (ст. 109 УК РФ);
- доведение до самоубийства (ст. 110 УК РФ).

Рассмотрим элементы состава преступления против жизни.

Объектом преступления является жизнь человека. Началом жизни принято считать начало процесса рождения (прорезывания головки младенца, выходящего из организма матери), окончанием жизни — момент наступления биологической смерти (состояния, в котором наступают необратимые изменения функций центральной нервной системы) [13].

Объективная сторона преступления может выражаться в действии или бездействии, направленном на лишение жизни объекта.

Субъективная сторона характеризуется умыслом, кроме ст. 109 — причинение смерти по неосторожности.

Субъектом в данных составах преступления является лицо, достигшее 16-летнего возраста, за исключением ст. 105 УК РФ, в которой субъектом является лицо, достигшее 14 лет.

Преступления против здоровья — это общественно-опасные деяния, посягающие на здоровье человека и приносящие вред различной степени тяжести. Понятие «вред здоровью» трактуется как нарушение целостности тканей, органов или их функционального состояния, а также вызванные механическим, химическим, психическим или биологическим воздействием внешней среды заболевания или иные патологические состояния [13].

Тяжесть нанесенного вреда здоровью определяется в результате судебно-медицинской экспертизы. Уголовный кодекс РФ выделяет три степени тяжести: легкую, среднюю, тяжкую.

Объектом в данных преступлениях является здоровье человека. Под здоровьем следует понимать «физическое и психическое состояние организма человека, обеспечивающее его нормальное функционирование» [13].

Объективная сторона, также как и в преступлениях против жизни, выражается либо в действии, либо в бездействии, а субъективная сторона характеризуется как умышленной, так и неосторожной формой вины.

Субъектом является вменяемое лицо, достигшее 16 лет. В случае нанесения тяжкого вреда здоровью (ст. 111 УК РФ) и нанесения вреда здоровью средней тяжести (ст. 112 УК РФ) ответственность наступает с 14 лет.

Глава 3. Архитектура программного комплекса наивной квалификации в теории уголовно-правовой оценки деяния

Процесс установления соответствия между признаками содеянного общественно-опасного деяния и признаками конкретных составов преступлений, предусмотренными Особенной и Общей частями УК РФ, называется квалификацией преступлений. Результатом квалификации является вывод о применении

той или иной статьи Уголовного кодекса [13].

Ясно, что для установления данного соответствия необходимо выделить конкретные признаки каждого состава преступления, предусмотренного Уголовным кодексом РФ, которые отделяют один состав преступления от другого (например, убийство матерью новорожденного ребенка от убийства человека старше 14 лет).

3.1. Процесс построения семантического шаблона для составов преступлений УК РФ

Рассмотрим процесс выделения атрибутов и их значений на примере главы 16 УК РФ. Из вышенаписанного следует, что в главе 16 УК РФ имеется два основных объекта преступления — жизнь человека и здоровье человека. Иными словами, можно выделить два корневых элемента семантического шаблона — живой человек (*AliveHuman*) и мертвый человек (*DeadHuman*). Элемент *DeadHuman* содержит атрибуты, описывающие личность убитого, его деятельность, семейное положение, а также следы на теле. Атрибуты элемента *AliveHuman* описывают личность потерпевшего, его деятельность, семейное положение, состояние здоровья. Семантические шаблоны преступлений против жизни и преступлений против здоровья начинаются соответственно с элементов *DeadHuman* и *AliveHuman*.

Помимо элементов, описывающих состояние личности, необ-

ходимо выделить элемент, характеризующий объективную сторону преступления или обстоятельства совершения преступления — *Circumstances*. Данный элемент содержит атрибуты, описывающие метод совершения преступлений, орудия, а также чем занимался объект во время совершения преступления.

На рисунках 1 — 3 представлены все возможные атрибуты выделенных элементов.

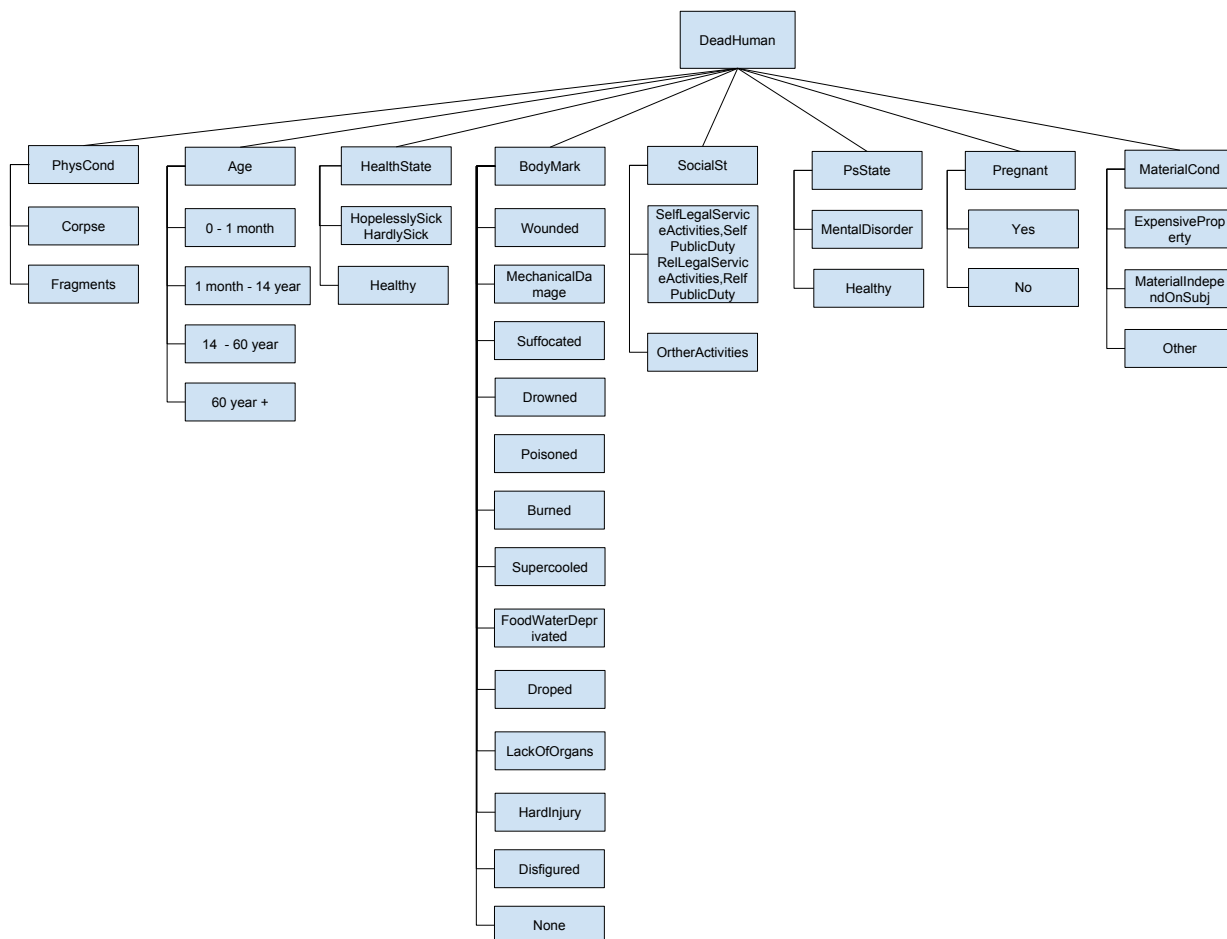


Рис. 1: Дерево атрибутов элемента Dead Human

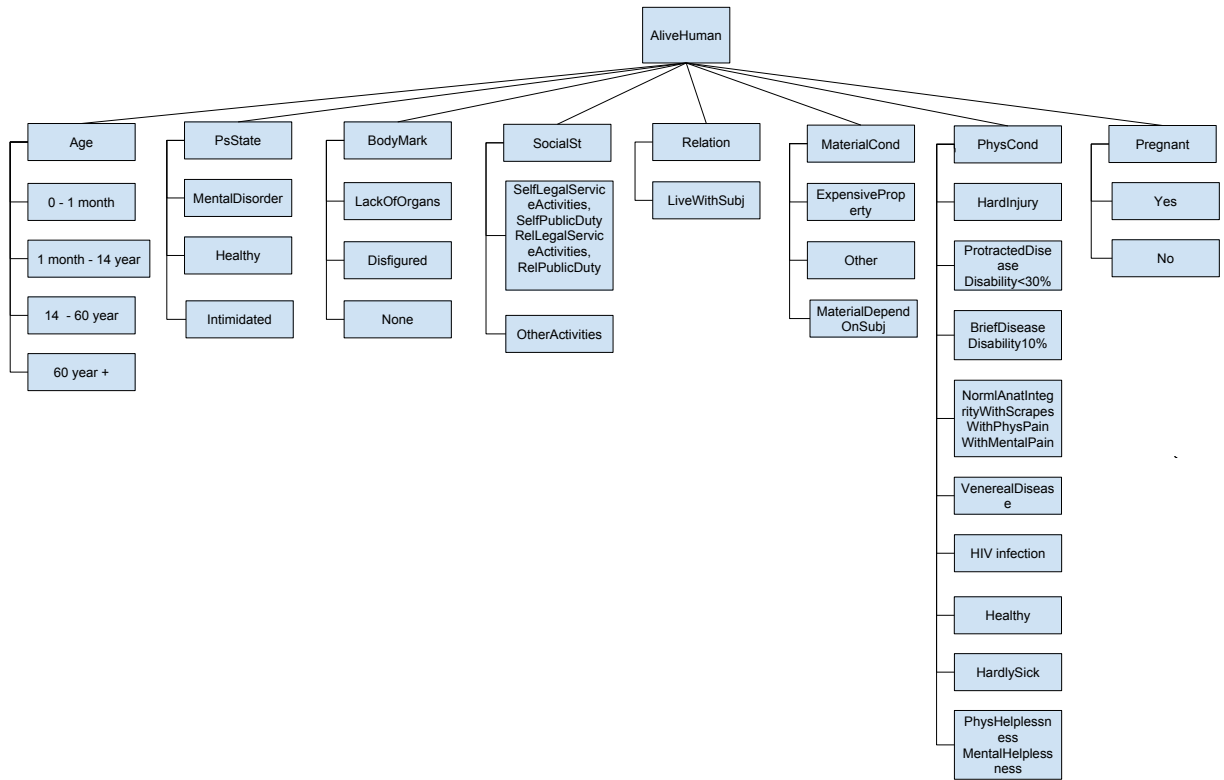


Рис. 2: Дерево атрибутов элемента Alive Human

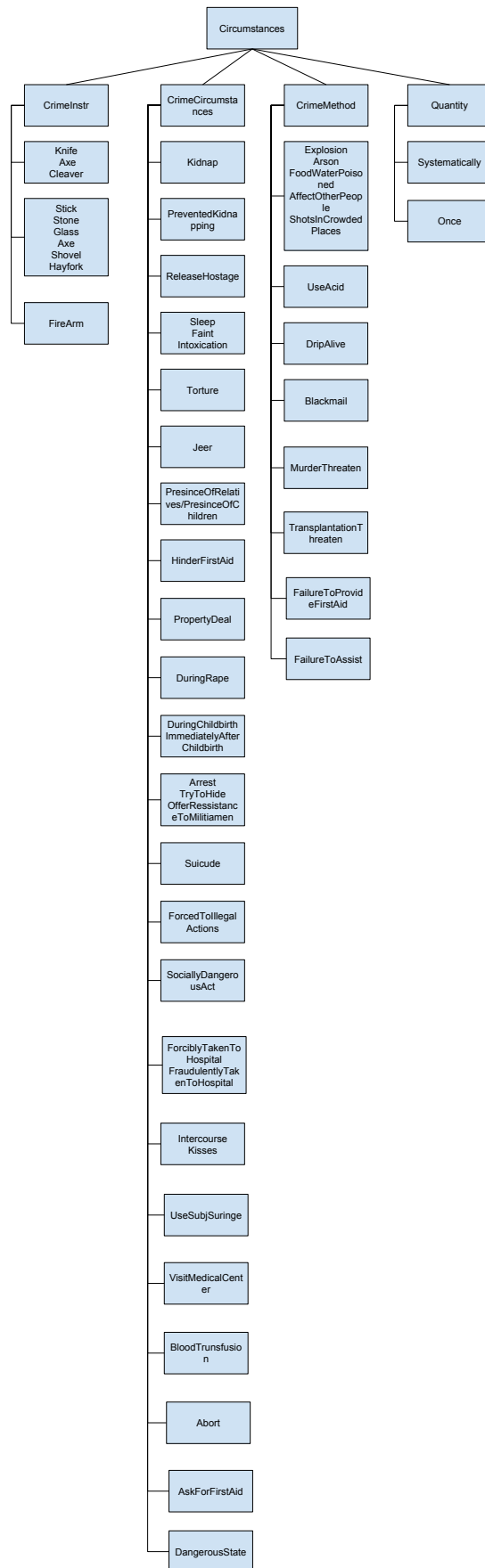


Рис. 3: Дерево атрибутов элемента Circumstances

На примере статьи 105 ч. 2 п. г. УК РФ — убийство женщины, заведомо для виновного находящейся в состоянии беременности, — рассмотрим процесс построения семантического шаблона для данного состава преступления. В данном случае объектом преступления является право человека на жизнь, а предметом преступления — сам человек. Естественно предположить, что на месте преступления было найдено тело или фрагменты тела мертвого человека, следовательно, корневым элементом шаблона является *DeadHuman*. Также известно, что найдено тело или фрагменты тела взрослой беременной женщины:

- *DeadHuman:PhysCond:CorpseOrFragments:1*,
- *DeadHuman:Age:14-60:1*,
- *DeadHuman:Gender:Female:1*,
- *DeadHuman:Pregnant:1*.

Повышенная опасность данного состава преступления состоит в том, что, убивая беременную женщину, субъект убивает и ее плод — зародыш будущей жизни. Однако совершенное опасное деяние следует квалифицировать по данному пункту статьи только в том случае, когда субъект заведомо знал о беременности потерпевшей.

Аналогично выделяются значимые атрибуты и в остальных статьях главы 16 УК РФ.

3.2. Теоретический алгоритм работы

Схематично разрабатываемый комплекс можно представить следующим образом (рис. 4).

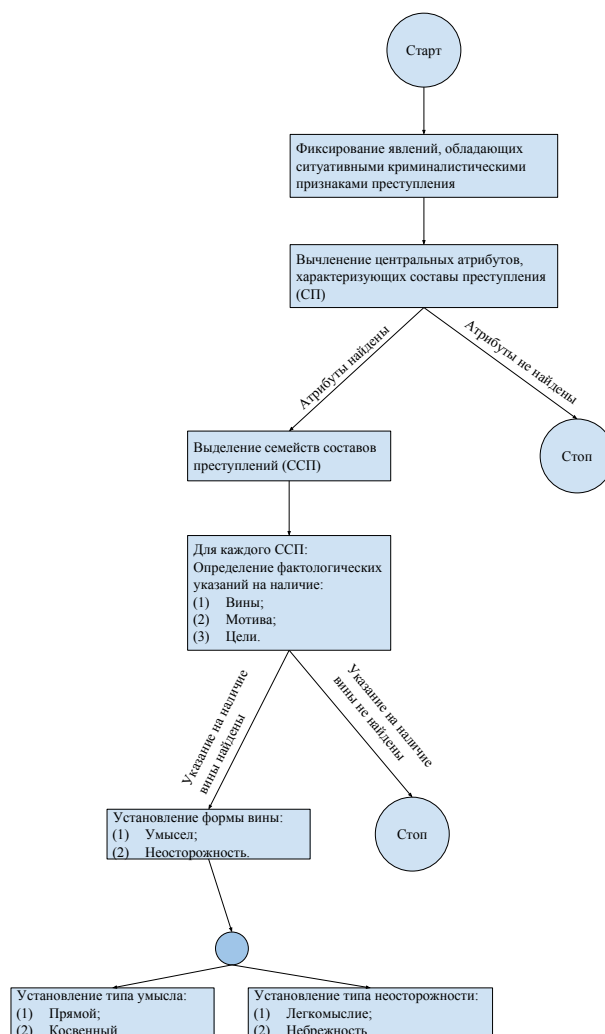


Рис. 4: Схема разрабатываемого информационного комплекса

На вход системе поступают материалы дела. Материалы дела представляют собой набор «информационных потоков». Каждый информационный поток — это версия фигуранта (обвиняемого, свидетеля, подозреваемого, потерпевшего, дознавателя и т. д.). Вследствие того, что основанием для признания деяния преступным и привлечения совершившего его лица к

уголовной ответственности является наличие в нем всех признаков некоего состава преступления, как было описано выше, на этапе предварительной обработки с помощью семантического анализа материалов дела «вычлняются» основные атрибуты, относящиеся ко всем выделенным признакам объекта и объективной стороны всех составов преступлений (0 — отсутствие данного атрибута в рассматриваемом информационном потоке, 1 — наличие данного атрибута в рассматриваемом информационном потоке), после чего осуществляется сопоставление данных наборов значений атрибутов и семейств составов преступлений, иными словами установление определенного класса или классов. На следующих шагах устанавливаются формы вины, мотив и цели, то есть признаки, относящиеся к субъекту и субъективной стороне.

В данной работе я хочу остановиться на определении семейств составов преступлений, основываясь на атрибутах, описывающих объект и объективную сторону. Например, рассмотрим состав преступления, предусмотренный ст. 106 УК РФ, убийство матерью новорожденного ребенка (во время или сразу же после родов). Пусть на вход системе подаются показания, в которых сказано, что на месте преступления, в родильном доме, было найдено мертвое тело новорожденного ребенка возрастом до четырех недель, на котором имелись следы удушения. Также предположим известно, что опасное деяние совершила мать ребенка сразу же после родов, однако роды не вызвали никаких заметных психических расстройств у роженицы. Таким образом, следующие атрибуты должны

принять значение 1:

- *DeadHuman:PhysCond:Corpse:1,*
- *DeadHuman:Age:0-1 month:1,*
- *DeadHuman:BodyMark:Suffocated:1,*
- *Circumstances:CrimeCircumstances:
ImmediatelyAfterChildbirth:1.*

Остальные атрибуты должны принять значение 0. В результате работы на выходе мы должны получить класс, относящийся к ст. 106 УК РФ.

Становится ясно, что задача наивной квалификации преступных деяний математически сводится к задаче классификации — одной из классических задач интеллектуально анализа данных. Интеллектуальный анализ данных (с английского Data Mining) — это технология, которая изучает процесс нахождения новых, действительных и потенциально полезных знаний в сырых данных. Данная технология лежит на пересечении нескольких наук: искусственный интеллект, математическая статистика, и нашла применение в различных областях. Право — одна из областей, в которой применение техник интеллектуального анализа данных, в том числе методов классификации, могут принести полезные и важные результаты.

Глава 4. Задача классификации

4.1. Постановка задачи классификации

Классификация — задача, в которой необходимо отнести рассматриваемый объект к одному из заранее известных классов.

В машинном обучении задача классификации относится к задачам обучения с учителем, т. е. известна конечная совокупность прецедентов «объект-класс» (x^i, y_i) , называемая обучающей выборкой, по которой необходимо построить алгоритм, который любому объекту ставит в соответствие определенный класс.

Классическая постановка задачи в контексте задачи наивной квалификации преступных деяний выглядит следующим образом:

пусть X — множество атрибутов всех составов преступлений, \vec{x}^i — вектор, размерность которого совпадает с общим количеством атрибутов, известных системе, $x_j^i = \{0; 1\}$ — бинарный признак, Y — конечное множество классов, в нашем случае составов преступлений. Существует неизвестная целевая зависимость — отображение $y^* : X \rightarrow Y$, значения которой известны только на объектах обучающей выборки $X^m = \{(\vec{x}^1, y_1), \dots, (\vec{x}^m, y_m)\}$. Требуется построить алгоритм $a : X \rightarrow Y$, способный классифицировать произвольный объект $\vec{x} \in X$.

В зависимости от типов классов, задача классификация бывает:

- двухклассовая классификация — $Y \in \{-1; 1\}$;
- многоклассовая классификация — количество классов больше двух;
- непересекающиеся классы — объект относится только к одному классу;
- пересекающиеся классы — объект может относиться сразу к нескольким классам. В русскоязычной литературе для данного определения также можно встретить термин «многозначная классификация» в виду того, что нет четкого перевода английского термина multi-label classification;
- нечеткие классы — определение степени принадлежности каждому классу, значение в диапазоне $[0, 1]$.

В данном случае перед нами ставится задача многоклассовой классификации с пересекающимися классами, так как:

- количество классов: 71;
- возможна ситуация, когда субъект последовательно совершил несколько преступлений, ни за одно из которых не был осужден ранее, тогда, согласно ст. 17 УК РФ, данные деяния необходимо квалифицировать по совокупности преступлений, т. е. описываемое атрибутом деяние может относиться к нескольким классам одновременно. В рамках данной работы мы рассматриваем совокупность преступлений относительно одного объекта.

Таким образом, задачу наивной квалификации преступных деяний математически можно определить следующим образом:

пусть X — множество атрибутов всех составов преступлений, \vec{x}^i — вектор, размерность которого совпадает с общим количеством атрибутов, известных системе, $x_j^i = \{0; 1\}$ — бинарный признак, Y — конечное множество классов, в нашем случае составов преступлений $Y = \{0, \dots, q\}$. Дана обучающая выборка $T = \{(\vec{x}^1, y^1), \dots, (\vec{x}^m, y^m)\}$, $\vec{x}^i \in X, y^i \subseteq Y$, взятая из неизвестного распределения D . Необходимо построить алгоритм $h : X \rightarrow 2^Y$, который оптимизирует заданную функцию потерь [14].

4.2. Обзор существующих методов решения задачи классификации с пересекающимися классами

Нетрудно понять, что задача многоклассовой классификации с пересекающимися классами является более сложной, чем классическая задача бинарной классификации и задача многоклассовой классификации.

На сегодняшний день только несколько статей затрагивают данную задачу в контексте задачи категоризации текстов [15, 16, 17, 18, 19, 20], задач биоинформатики [21, 22] и задачи классификации изображений [23].

Методы решения задачи классификации с пересекающимися классами делятся на две основные группы:

- методы преобразования задачи;
- методы адаптации алгоритмов.

Первая группа методов — преобразование задачи — сводят задачу классификации с пересекающимися классами к классической задаче многоклассовой классификации. К данной группе относятся следующие алгоритмы: Label Powerset, Binary Relevance, Error-Correcting Output Code.

Label Powerset [24] преобразует каждую уникальную совокупность классов в обучающей выборке в новый класс. Однако после такого преобразования большинство классов содержат очень мало объектов, выборка становится крайне несбалансированной. Для решения данной проблемы была предложена модификация метода: вводится порог отсеивания классов — все классы, которые имеют частоту встречаемости ниже данного порога, заменяются на непересекающиеся подмножества данных классов.

Error-Correcting Output Code (ECOC) [25] — в данном алгоритме каждая метка класса кодируется бинарным вектором длины l , далее для каждого бита кодового слова i строится бинарный классификатор f_i .

Вторая группа методов модифицируют существующие алгоритмы классификации для решения поставленной задачи. Для решения задачи классификации с пересекающимися классами используют модифицированные версии таких алгоритмов классификации, как метод k ближайших соседей, деревья решений, метод опорных векторов, нейронные сети, алгоритмы бустинга

(с англ. boosting).

4.3. Метод опорных векторов

Одним из наиболее популярных методов обучения по прецедентам является метод опорных векторов, в англоязычной литературе известный под названием Support Vector Machine (SVM). Сначала рассмотрим классический метод опорных векторов [26].

4.3.1. Классический метод опорных векторов

Пусть дана обучающая выборка $T = \left\{ \left(\vec{x}^1, y_1 \right), \dots, \left(\vec{x}^m, y_m \right) \right\}$, $\vec{x}^i \in R^d$ — вектор признаков описаний объекта, $y_i \in \{-1; +1\}$ — метка класса. Необходимо построить алгоритм, относящий произвольный объект к одному из классов по следующему линейному решающему правилу:

$$\hat{y} = \text{sign}(\vec{w}^T \vec{x}^i + b), w_i \in R \text{—вес, } b \in R \text{—параметр сдвига.} \quad (1)$$

С геометрической точки зрения, (1) соответствует разделяющей гиперплоскости в признаковом пространстве R^d . Если объект лежит с положительной стороны от нее, то он относится к классу с меткой +1, иначе — к классу с меткой -1.

Предположим, что обучающая выборка T является линейно разделимой, т. е. можно найти такие параметры w и b ,

что для любого объекта \vec{x}^i верно следующее:

$$\vec{w}^T \vec{x}^i + b > 0, \text{ если } y_i = +1$$

$$\vec{w}^T \vec{x}^i + b < 0, \text{ если } y_i = -1.$$

Среди всех возможных гиперплоскостей (построить их можно разными способами) необходимо выбрать ту, которая разделяет классы наилучшим образом — оптимальная разделяющая гиперплоскость. Такая гиперплоскость должна максимально далеко отстоять от ближайших к ней точек обоих классов. Чем выше зазор, тем надежнее классификация и выше обучающая способность. Изначально данное предположение было сделано из эвристических соображений, однако позже получило теоретическое обоснование.

Вычислим ширину разделяющей полосы. Для того, чтобы каждый класс отстоял от гиперплоскости максимально далеко, необходимо, чтобы ширина полосы была максимальной. Обозначим за x_+ и x_- — элементы классов $+1$ и -1 , лежащие на границе полосы, соответственно. Тогда ширина полосы:

$$\begin{aligned} \left\langle (x_+ - x_-), \frac{\vec{w}}{\|\vec{w}\|} \right\rangle &= \frac{\langle \vec{w}, x_+ \rangle - \langle \vec{w}, x_- \rangle}{\|\vec{w}\|} = \\ &= \frac{(b+1) - (b-1)}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}. \end{aligned}$$

Таким образом, задачу максимизации зазора математически можно записать следующим образом:

$$\left\{ \begin{array}{l} \frac{2}{\|\vec{w}\|} \rightarrow \max_{\vec{w}, b} \\ \vec{w}^T \vec{x}^i + b > 0 \text{ если } y_i = +1 \\ \vec{w}^T \vec{x}^i + b < 0 \text{ если } y_i = -1 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \frac{1}{2} \|\vec{w}\|^2 \rightarrow \min_{\vec{w}, b}, \\ y_i (\vec{w}^T \vec{x}^i + b) \geq 1. \end{array} \right.$$

Ясно, что для произвольных данных условие $y_i (\vec{w}^T \vec{x}^i + b) \geq 1$ может не выполняться, т. е. необходимо обобщить метод опорных векторов на случай линейной неразделимости выборки. Для решения данной проблемы разрешим классификатору ошибаться на обучающих объектах, но за каждую ошибку будем накладывать неотрицательный штраф $\xi_i: y_i (\vec{w}^T \vec{x}^i + b) \geq 1 - \xi_i$.

Возможны три ситуации:

- $\xi_i = 0$ — ошибки нет, объект лежит по одну из сторон разделяющей полосы;
- $0 < \xi_i \leq 1$ — ошибки нет, но объект лежит внутри разделяющей полосы;
- $\xi_i > 1$ — ошибка есть, величина ошибки пропорциональна расстоянию от объекта до разделяющей гиперплоскости.

В таком случае задача максимизации зазора принимает вид:

$$\begin{cases} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^m \xi_i \rightarrow \min_{\vec{w}, b, \xi}, \\ y_i (\vec{w}^T \vec{x}^i + b) \geq 1 - \xi_i, \forall i = 1, \dots, n, \\ \xi_i > 0. \end{cases} \quad (2)$$

Параметр C — коэффициент регуляризации, отвечающий за компромисс между величиной ошибок и близостью весов w_i к 0.

Построим двойственную задачу к задаче (2), записав функцию Лагранжа:

$$\begin{aligned} L(\vec{w}, b, \xi, \lambda, \mu) = & \frac{1}{2} \vec{w}^T \vec{w} + C \sum_{n=1}^m \xi_n - \\ & - \sum_{n=1}^m \lambda_n (y_n (\vec{w}^T \vec{x}^n + b) - 1 + \xi_n) - \sum_{n=1}^m \mu_n \xi_n. \end{aligned}$$

Условия Куна-Таккера:

$$\begin{aligned} \nabla_w L(\vec{w}, b, \xi) = \vec{w} - \sum_{n=1}^m \lambda_n y_n \vec{x}^n = 0 & \Rightarrow \vec{w} = \sum_{n=1}^m \lambda_n y_n \vec{x}^n, \\ \frac{\partial}{\partial b} L(\vec{w}, b, \xi) = - \sum_{n=1}^m \lambda_n y_n = 0 & \Rightarrow \sum_{n=1}^m \lambda_n y_n = 0, \\ \frac{\partial}{\partial \xi_n} L(\vec{w}, b, \xi) = -\lambda_n - \mu_n + C = 0 & \Rightarrow \lambda_n + \mu_n = C. \end{aligned} \quad (3)$$

Подставим полученные условия (3) в функцию Лагранжа и найдем двойственную функцию:

$$\begin{aligned}
g(\lambda, \mu) &= \frac{1}{2} \sum_{n,l=1}^m \lambda_n \lambda_l y_n y_l (\vec{x}^n)^T \vec{x}^l - \sum_{n=1}^m \lambda_n = \\
&= -\frac{1}{2} \lambda^T \text{diag}(y) X^T X \text{diag}(y) \lambda + \lambda^T a_e,
\end{aligned}$$

где a_e — это единичный вектор длины m .

Таким образом, получаем следующую двойственную задачу:

$$\begin{cases}
-\frac{1}{2} \lambda^T \text{diag}(y) X^T X \text{diag}(y) \lambda + \lambda^T a_e \rightarrow \min_{\lambda}, \\
y^T \lambda = 0, \\
0 \leq \lambda_n \leq C, n = 1, \dots, m.
\end{cases} \quad (4)$$

Решение прямой задачи (2) может быть получено из решения двойственной задачи с помощью первого условия из (3) и условия дополняющей нежесткости:

$$\lambda_n^* (y_n (\vec{w}^*)^T \vec{x}^n + b^*) + 1 - \xi_n^* = 0, n = 1, \dots, m.$$

Возможны три ситуации:

- $\lambda_n^* = 0$, объект \vec{x}^n лежит внутри класса с верной стороны относительно гиперплоскости;
- $y_n ((\vec{w}^*)^T \vec{x}^n + b^*) = 1, \xi_n^* = 0$, объект \vec{x}^n лежит на границе разделяющей полосы;
- $y_n ((\vec{w}^*)^T \vec{x}^n + b^*) = 1 - \xi_n^*, \xi_n^* > 0$, на объекте \vec{x}^n классификатор ошибается, либо объект \vec{x}^n лежит внутри

разделяющей полосы.

Получившееся линейное решающее правило имеет вид:

$$\hat{y} = \text{sign}((\vec{w}^*)^T \vec{x} + b^*) = \sum_{n=1}^m \lambda_n^* y_n (\vec{x}^n)^T \vec{x} + b^*. \quad (5)$$

Только объекты, лежащие на границе или внутри разделяющей полосы, а также объекты, на которых классификатор ошибается, влияют на данное решающее правило. Такие объекты называются опорными векторами.

На практике при решении задачи классификации решают именно задачу (4), т. к. нельзя точно гарантировать, что исходная выборка является линейно разделимой. Модификацию метода опорных векторов для линейно неразделимой выборки называют методом опорных векторов с мягким зазором, а для линейно разделимого случая — методом опорных векторов с жестким зазором

Для простоты вычислений далее опустим свободный член b .

Перепишем данную задачу иначе.

Пусть $\vec{y}^i \in \{0, 1\}^2$, $\sum_{j=1}^2 y_j^i = 1$ — бинарный двухмерный вектор, в котором присутствует только одна единица в той позиции, номер которой соответствует номеру соответствующего класса i -го объекта. Введем функцию:

$$S(\vec{y}, \vec{x}, W) = \vec{y}^T W \vec{x},$$

где $W \in R^{2 \times m}$, $W = (w_{ij})_{i=1, j=1}^{2, m}$ — набор коэффициентов

линейной комбинации весов для каждого класса.

Хотим, чтобы выход S на верной метке класса был больше, чем на иной: $S(\vec{y}^i, \vec{x}^i, W) \geq S(\vec{y}^j, \vec{x}^i, W)$. Формализуем данное условие.

Функционал остается без изменений, меняются лишь ограничения:

$$\begin{cases} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^m \xi_i \rightarrow \min_{W, \vec{\xi}}, \\ S(\vec{y}^i, \vec{x}^i, W) \geq S(\vec{y}, \vec{x}^i, W) + \Delta(\vec{y}, \vec{y}_i) - \xi_i, \forall \vec{y}, \forall i, \end{cases}$$

где $\Delta(\vec{y}, \vec{y}_i)$ — манхэттенское расстояние между векторами.

Убедимся, что полученные условия эквивалентны исходным.

Случай 1. $\vec{y} = \vec{y}^i \Rightarrow S(\vec{y}^i, \vec{x}^i, W) = S(\vec{y}, \vec{x}^i, W)$,

$\Delta(\vec{y}, \vec{y}_i) = 0 \Rightarrow \xi_i \geq 0$.

Случай 2. $\vec{y} \neq \vec{y}^i \Rightarrow S(\vec{y}^i, \vec{x}^i, W) = -S(\vec{y}, \vec{x}^i, W)$,

$\Delta(\vec{y}, \vec{y}_i) = 2 \Rightarrow S(\vec{y}, \vec{x}^i, W) \geq 1 - \frac{\xi_i}{2}$, что эквивалентно второму условию из (2).

Данное преобразование позволяет эффективно обобщить классический метод опорных векторов.

4.3.2. Выбор ядра

Обобщить метод опорных векторов на случай линейно неразделимой выборки можно и с помощью «ядрового перехода» — перехода от исходного пространства X к новому

пространству H достаточно высокой размерности, в котором с большей вероятностью выборка является линейно разделимой: $\psi : X \rightarrow H$. Пространство H называется спрямляющим пространством [27].

Построение оптимальной разделяющей гиперплоскости в новом пространстве H проводится аналогично. Двойственная задача (4), решающее правило (5) теперь зависят от скалярных произведений $\langle \psi(x^m), \psi(x^n) \rangle$, следовательно, пространство H должно обладать скалярным произведением. Таким образом, в качестве спрямляющего пространства подойдет любое гильбертово пространство.

Пусть известна функция $K(x^m, x^n) = \langle \psi(x^m), \psi(x^n) \rangle$ — скалярное произведение в пространстве H . Такую функцию будем называть ядровой функцией: $K : X \times X \rightarrow R$. Ясно, что для обучения метода достаточно лишь подобрать ядровую функцию K .

Согласно теореме Мерсера [28] функция K должна обладать следующими свойствами:

- Симметричность: $K(x^m, x^n) = K(x^n, x^m)$.
- Неотрицательная определенность:

$$\int_X \int_X K(x^m, x^n) g(x^m) g(x^n) dx^m dx^n \geq 0$$

для любой $g : X \rightarrow R$.

Зачастую проверка второго условия — неотрицательной определенности — является достаточно трудным процессом, поэтому на практике занимаются перебором заранее известных

ядер с помощью критерия скользящего контроля. К стандартным функциям ядра относятся:

- линейное: $K(x^m, x^n) = (x^m)^T x^n$;
- полиномиальное: $K(x^m, x^n) = ((x^m)^T x^n + r)^d$;
- радиально-базисное: $K(x^m, x^n) = e^{-\gamma \|x^m - x^n\|^2}$;
- сигмоидальное: $K(x^m, x^n) = \tanh(\gamma (x^m)^T x^n + r)$.

4.3.3. Метод опорных векторов для многоклассовой классификации с пересекающимися классами

Пусть теперь объект может относиться сразу к нескольким классам. Дана обучающая выборка $T = \{(\vec{x}^1, \vec{y}^1), \dots, (\vec{x}^m, \vec{y}^m)\}$, $\vec{x}^i \in X$, $\vec{y}^i \in \{0, 1\}^q$ — бинарный вектор размерности q . Если $y_j^i = 1$, — это значит, что объект \vec{x}^i принадлежит классу с номером j , иначе — не принадлежит.

Для случая многоклассовой классификации с пересекающимися классами существует алгоритм, который на английском называется Binary Relevance (BR) [29], сводящий задачу к задаче бинарной классификации.

BR — это метод, основанный на предположении о независимости исходных классов. Данный алгоритм создает q выборок $T_i, i = 1, \dots, q$, где q — это количество классов. Каждая выборка содержит одинаковое число объектов, совпадающее с числом объектов в изначальной обучающей

выборке. В выборке T_i каждый объект имеет либо позитивный класс, если он принадлежит классу i , либо негативный класс в ином случае. На каждом таком наборе данных T_i обучается бинарный классификатор. Результирующими классами объекта являются релевантные классы каждого бинарного классификатора для данного объекта, иными словами, j -ый компонент вектора $\vec{y}^i - y_j^i$ — это результат, полученный с помощью j -го бинарного классификатора.

VR является аналогом стратегии «один против всех» для многоклассового случая. Однако стоит заметить, что эти две стратегии имеют много общего, но не являются идентичными. В алгоритме VR мы обучаем один бинарный классификатор для каждого класса, в отличие от стратегии «один против всех», где обучается один классификатор для каждого возможного значения класса.

В качестве бинарного классификатора можно использовать классический метод опорных векторов, тогда линейное решающее правило для каждого классификатора имеет вид (1).

4.3.4. Преимущества и недостатки метода опорных векторов

Метод опорных векторов, как было написано выше, сводится к решению задачи квадратичного программирования, решение которой всегда единственно. Данный метод находит разделяющую полосу максимальной ширины, что позволяет производить более уверенную классификацию. Также на

практике зачастую данный метод показывает результаты, превосходящие результаты других алгоритмов.

Однако помимо перечисленных выше преимуществ, метод опорных векторов обладает следующими недостатками:

- медленное обучение;
- всего один параметр C , которым мы можем варьировать;
- чувствителен к шумам и стандартизации данных;
- не существует общего подхода к выбору ядра в случае линейной неразделимости классов.

4.4. ML–kNN — модификация метода k ближайших соседей для многоклассовой классификации с пересекающимися классами

Рассмотренный в разделе 4.3.3. метод Binary Relevance, как было написано выше, относится к первой группе методов решения задачи классификации с пересекающимися классами — преобразование задачи. Главный недостаток алгоритмов данной группы заключается в том, что они не учитывают корреляцию между классами, поэтому их выразительная сила слаба. Рассмотрим теперь метод, который относится ко второй группе — методы адаптации алгоритмов, — ML–kNN.

Метод k ближайших соседей относится к метрическим методам классификации объектов. Основная особенность данного метода заключается в том, что он относит объект к классу, к которому принадлежат большинство из его k соседей. Ясно, что перед обучением необходимо выбрать функцию расстояния между двумя объектами.

ML-kNN — это модификация алгоритма k ближайших соседей для многоклассовой классификации с пересекающимися классами [14]. Сначала для каждого объекта находят k ближайших соседей из обучающего множества. Затем с помощью статистических данных, полученных из наборов меток классов каждого объекта, например, количество соседних объектов, которые принадлежат определенному классу, и оценки апостериорного максимума определяется набор меток классов для каждого тестового объекта.

Пусть известна обучающая выборка $T = \{(\vec{x}^1, \vec{y}^1), \dots, (\vec{x}^m, \vec{y}^m)\}$, $\vec{x}^i \in X, \vec{y}^i \in \{0, 1\}^q$ — бинарный вектор размерности q . Если $y_j^i = 1$, это значит, что объект \vec{x}^i принадлежит классу с номером j , иначе — не принадлежит. Также предположим, что мы нашли k соседей для каждого объекта — они составляют множество $N(\vec{x})$.

Основываясь на метках классов соседей объекта, можно ввести вектор $\vec{C}_{\vec{x}}(l)$:

$$\vec{C}_{\vec{x}}(l) = \sum_{a \in N(\vec{x}^i)} \vec{y}_l^a.$$

Вектор $\vec{C}_{\vec{x}}(l)$ — это количество соседей объекта \vec{x} , которые

принадлежат классу с индексом l .

Для каждого тестового объекта \vec{t} алгоритм ML-kNN сначала строит множество соседей $N(\vec{t})$. Пусть величина H_1^l описывает событие, когда объект \vec{t} принадлежит классу с индексом l , H_0^l — объект \vec{t} не принадлежит классу с индексом l . Введем величину E_j^l , $j \in \{0, \dots, k\}$, которая указывает на то, что среди k ближайших соседей объекта \vec{t} существует хотя бы j объектов, которые относятся к классу с индексом l .

Таким образом, основываясь на векторе $\vec{C}_{\vec{t}}$, вектор классов $\vec{y}_l(\vec{t})$ определяется, используя оценку апостериорного максимума:

$$\vec{y}_l(\vec{t}) = \underset{b \in \{0,1\}}{\operatorname{argmax}} P(H_b^l | E_{\vec{C}_{\vec{t}}}^l). \quad (6)$$

Используя формулу Байеса, (6) может быть переписано как:

$$\begin{aligned} \vec{y}_l(\vec{t}) &= \underset{b \in \{0,1\}}{\operatorname{argmax}} \frac{P(H_b^l) P(E_{\vec{C}_{\vec{t}}}^l | H_b^l)}{E_{\vec{C}_{\vec{t}}}^l} = \\ &= \underset{b \in \{0,1\}}{\operatorname{argmax}} P(H_b^l) P(E_{\vec{C}_{\vec{t}}}^l | H_b^l). \end{aligned} \quad (7)$$

Как показано в формуле (7), для того, чтобы определить вектор классов $\vec{y}_l(\vec{t})$, надо знать априорные вероятности $P(H_b^l)$ и апостериорные вероятности $P(E_{\vec{C}_{\vec{t}}}^l | H_b^l)$. Априорные и апостериорные вероятности могут быть вычислены с помощью обучающей выборки, основываясь на вычислении частоты.

Ниже представлен псевдокод, дающий полное описание

алгоритма. Параметр s — это параметр сглаживания, ранее он рассматривался равным 1 (сглаживание Лапласа). $\vec{r}_l(\vec{t})$ — это вещественный вектор для ранжирования результатов. Для вычисления апостериорных вероятностей вводятся параметры $c[j]$ — количество объектов обучающей выборки, имеющих метку класса l , среди k соседей которых хотя бы j относятся к классу с индексом l , $c'[j]$ — количество объектов обучающей выборки, не имеющих метку класса l , среди k соседей которых хотя бы j относятся к классу с индексом l .

Algorithm 1 ML-kNN

//вычисление априорных вероятностей $P(H_b^l)$

for $l = 1, \dots, q$ do

$$P(H_1^l) = \frac{s + \sum_{i=1}^m y_i^l}{s \times 2 + m}; P(H_0^l) = 1 - P(H_1^l);$$

//вычисление апостериорных вероятностей

Найти $N(\vec{x}^i), i = 1, \dots, m;$

for $l = 1, \dots, q$ do

for $j \in \{0, \dots, k\}$ do

$$c[j] = 0; c'[j] = 0;$$

for $i \in \{0, \dots, m\}$ do

$$\delta = \vec{C}_{\vec{x}^i}(l) = \sum_{a \in N(\vec{x}^i)} \vec{y}_l^a;$$

$$\text{if } (y_i^l == 1) \text{ then } c[\delta] = c[\delta] + 1;$$

$$\text{else } c'[\delta] = c'[\delta] + 1;$$

for $j \in \{0, \dots, k\}$ do

$$P(E_j^l | H_1^l) = \frac{s + c[j]}{s \times (k+1)} + \sum_{p=0}^k c[p];$$

$$P(E_j^l | H_0^l) = \frac{s + c'[j]}{s \times (k+1)} + \sum_{p=0}^k c'[p];$$

//вычисление $\vec{y}(t), \vec{r}(t)$

Найти $N(\vec{t});$

for $l = 1, \dots, q$ do

$$\delta = \vec{C}_{\vec{t}}(l) = \sum_{a \in N(\vec{t})} \vec{y}_l^a;$$

$$\vec{y}_l(t) = \operatorname{argmax}_{b \in \{0,1\}} P(H_b^l) P(E_{\vec{C}_{\vec{t}}^l}^l | H_b^l);$$

$$\vec{r}_l(t) = \frac{P(H_1^l | E_{\vec{C}_{\vec{t}}^l}^l)}{P(E_{\vec{C}_{\vec{t}}^l}^l)} = \frac{P(H_1^l) P(E_{\vec{C}_{\vec{t}}^l}^l | H_1^l)}{\sum_{b \in \{0,1\}} P(H_b^l) P(E_{\vec{C}_{\vec{t}}^l}^l | H_b^l)}.$$

При $k = 1$ данный метод неустойчив к шуму —

классификатор ошибается не только на шумовых выбросах, но и на ближайших к ним объектах других классов. При $k = m$ (размер выборки), наоборот, алгоритм вырождается в константу. На практике оптимальное значение параметра k выбирается с помощью критерия скользящего контроля.

Скользящий контроль или кросс-валидация — это процедура эмпирического оценивания обобщающей способности алгоритмов. Вся выборка разделяется на две подвыборки — обучающую и контрольную. Для каждого такого разбиения по обучающей выборке выполняется настройка алгоритма, а затем оценивается средняя ошибка на контрольной подвыборке. Скользящий контроль бывает различных видов:

- полный скользящий контроль;
- контроль по отдельным объектам;
- случайные разбиения;
- контроль на отложенных данных;
- контроль по k блокам;
- контроль по $r \times k$ блокам.

Если объем исходной выборки достаточно большой, то рекомендуется использовать скользящий контроль с помощью случайных разбиений или контроль по k блокам в виду того, что другие виды могут оказаться очень «дорогостоящими».

4.4.1. Преимущества и недостатки ML–kNN

Нетрудно заметить, что данный алгоритм довольно прост в реализации. Он обладает преимуществами, характерными для методов «ленивого обучения», а также для Байесовских методов:

- граница принятия решения может быть скорректирована, варьируя различным числом соседей для объектов;
- проблема несбалансированности классов (когда количество объектов, имеющих «позитивный» класс намного меньше, чем число объектов с «отрицательным» классом) может быть решена благодаря вычислению априорных вероятностей для каждого класса.

Главным недостатком ML–kNN является неэффективный расход памяти и высокая вычислительная трудоемкость. Данный метод чем-то схож с методом Binary Relevance, т. к. для каждого класса обучаются независимые классификаторы. Другими словами, он тоже не учитывает корреляцию между классами, за что и критикуется.

4.5. Метрики качества многоклассовой классификации с пересекающимися классами

Для классической задачи классификации существует большое количество метрик оценки качества алгоритма:

полнота, точность, F -мера, ROC-кривая. Данные метрики применимы при оценке качества алгоритмов, решающих задачу многоклассовой классификации. Однако, для задачи многоклассовой классификации с пересекающимися классами, где количество классов для каждого объекта может быть больше одного, мы можем говорить о различной степени точности (здесь под точностью стоит понимать полное совпадение множеств): полностью совпадает, не полностью совпадает или не совпадает.

Пусть $S : (x^i, Y_i)$ — тестовая выборка, $1 \leq i \leq p$, $x^i \in X$, $Y_i = \{0, 1\}^q$, h — классификатор, $Z_i = h(x^i) = \{0, 1\}^q$.

В виду того, что при многоклассовой классификации с пересекающимися классами полученный вектор классов для каждого объекта может лишь частично совпадать с реальным вектором, вычисление метрик качества достаточно затруднено. Именно поэтому, первый подход [30] предлагает забыть о частичном совпадении и считать, что в случае частичного совпадения векторы не равны, тогда в качестве метрики качества работы алгоритма можно использовать точность, как ее понимают в классической задаче классификации:

$$ExactMatchRatio, MR = \frac{1}{p} \sum_{i=1}^p I(Y_i = Z_i).$$

Ясно, что главный недостаток данной метрики заключается в том, что мы не разделяем понятия «полного несоответствия» и «частичного несоответствия» полученного классификатором вектора и тестового вектора меток классов.

Для того, чтобы можно было говорить о частичном совпадении, вводятся следующие метрики качества [30], которые можно разделить на две группы:

- основанные на объектах метрики — вычисляется значение метрики качества для каждого тестового объекта и возвращается среднее значение по всем объектам;
- основанные на классах метрики — вычисляется значение метрики качества для каждого класса отдельно и возвращается среднее значение по всем классам.

К первой группе относятся следующие метрики качества:

- метрика правильности (с английского accuracy) для каждого объекта — процент верно предсказанных меток класса. Метрика правильности — это среднее значение метрик правильности для всех объектов:

$$\text{Accuracy, } A = \frac{1}{p} \sum_{i=1}^p \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|},$$

- точность (с английского precision) — характеризует, сколько предсказанных положительных ответов являются правильными:

$$\text{Precision, } P = \frac{1}{p} \sum_{i=1}^p \frac{|Y_i \cap Z_i|}{|Z_i|},$$

- полнота (с английского recall) для каждого объекта — отношение числа верно предсказанных ответов к общему

числу предсказанных ответов для каждого объекта. Аналогично, полнота — это среднее значение метрик полноты для каждого объекта:

$$\text{Recall}, R = \frac{1}{p} \sum_{i=1}^p \frac{|Y_i \cap Z_i|}{|Y_i|},$$

- F_1 -мера. Точность и полнота характеризуют классификатор с разных сторон, но зачастую для оценки качества работы алгоритма удобно использовать одну величину, эта величина и есть F_1 -мера:

$$F_1 = \frac{1}{p} \sum_{i=1}^p \frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|},$$

- 1/0 функция потерь — штрафует модель на 1 за ошибку на каждом объекте:

$$L_{0-1}(Z^i, Y^i) = I(Z^i \neq Y^i),$$

где $I(z^i \neq y^i) = \begin{cases} 1, & \text{если } z^i \neq y^i \\ 0, & \text{иначе.} \end{cases}$

- функция потерь Хэмминга — показывает среднюю частоту ошибки классификатора:

$$HL = \frac{1}{qp} \sum_{i=1}^p \sum_{l=1}^q [I(l \in Z_i \wedge l \notin Y_i) + I(l \notin Z_i \wedge l \in Y_i)],$$

где $I(l \in M \wedge l \notin K) = \begin{cases} 1, & \text{если } l \in M \wedge l \notin K \\ 0, & \text{иначе.} \end{cases}$

В идеальном случае, когда классификатор не ошибается, предполагается, что $HL = 0$. Чем меньше значение функции потерь Хэмминга, тем лучше работает классификатор.

Для того, чтобы определить, какие метрики качества относятся ко второй группе, определим следующие величины:

$$TP_j = |\{x^i | y_j \in Y_i \wedge y_j \in Z_i, 1 \leq i \leq p\}|;$$

$$FP_j = |\{x^i | y_j \notin Y_i \wedge y_j \in Z_i, 1 \leq i \leq p\}|;$$

$$TN_j = |\{x^i | y_j \notin Y_i \wedge y_j \notin Z_i, 1 \leq i \leq p\}|;$$

$$FN_j = |\{x^i | y_j \notin Y_i \wedge y_j \notin Z_i, 1 \leq i \leq p\}|.$$

Другими словами, TP_j, FP_j, TN_j, FN_j — это число истинно-положительных, ложно-положительных, истинно-отрицательных и ложно-отрицательных тестовых объектов для каждой метки класса y_j . Ясно, что $TP_j + FP_j + TN_j + FN_j = p$.

Пусть $B(TP_j, FP_j, TN_j, FN_j)$ — это вычисленная определенная метрика качества бинарного классификатора, $B \in \{Accuracy, Precision, Recall, F_1\}$. Основанные на классах метрики могут быть вычислены следующим образом:

- Макро-усреднение:

$$B_{macro}(h) = \frac{1}{q} \sum_{j=1}^q B(TP_j, FP_j, TN_j, FN_j);$$

- Микро-усреднение:

$$B_{micro}(h) = B\left(\sum_{j=1}^q TP_j, \sum_{j=1}^q FP_j, \sum_{j=1}^q TN_j, \sum_{j=1}^q FN_j\right).$$

Глава 5. Эксперимент и апробация

5.1. Постановка эксперимента

В контексте создания информационного комплекса, предназначенного для сбора и анализа данных в рамках процессуальных действий уполномоченных лиц, необходимо проверить методику семантической разметки Уголовного Кодекса РФ и апробировать алгоритмы многоклассовой классификации с пересекающимися классами для задачи наивной квалификации преступных деяний.

В качестве модели машинного обучения были выбраны следующие методы:

- Binary Relevance с использованием метода опорных векторов (Support Machine Vector) и их реализации для языка *Python* — пакеты *sklearn.svm* и *sklearn.multiclass* из библиотеки *scikit-learn*. С помощью критерия скользящего контроля для метода опорных векторов было выбрано линейное ядро, так как оно показало наилучшие результаты;
- модифицированный метод k ближайших соседей — ML-kNN, реализованный в библиотеке *scikit-multilearn*

для языка *Python*. Оптимальное значение параметра k подбиралось с помощью критерия скользящего контроля.

5.2. Подготовка обучающего множества для задачи классификации

Для решения задачи квалификации преступных деяний предварительно был выделен 81 признак, описывающий объект и объективную сторону составов преступлений, предусмотренных главой 16 УК РФ. Количество классов, выделенных в данной главе, составляет 71.

Из выделенного набора атрибутов, искусственно ограниченного на примере главы 16 УК РФ, вручную была составлена выборка следующего вида (см. табл. 2.):

DeadHuman: PhysCond: Corpse	DeadHuman: PhysCond: Fragments	DeadHuman: Age: 0 — 1 m.	...	Circumstances: Quantity: Systematically	Class
1	0	0	...	0	1
0	1	0	...	0	3,7
0	0	0	...	1	69,71

Таблица 2: Вид обучающей выборки

В полях, описывающих атрибуты, 1 — наличие данного атрибута, 0 — отсутствие, например, если известно, что было найдено мертвое тело человека в возрасте от 20 до 30 лет, то тогда атрибут *DeadHuman:Age:18-60* примет значение 1, а остальные атрибуты, касающиеся возраста объекта примут значение 0. Естественно предположить также следующую зависимость: если на месте преступления было найдено мертвое тело человека, то данное деяние не относится к

семействам преступлений против здоровья человека, в нем можно предусмотреть только преступления против жизни человека.

Поле Class — это номер/номера соответствующих выделенных составов преступлений.

При составлении выборки рассматривались и продумывались базовые, самые распространенные составы преступлений. Выбирались основные атрибуты для конкретного состава, выставлялись их значения, а затем комбинировались с другими возможными, но не обязательными атрибутами. Оценка объектов производилась экспертами в области юриспруденции. Размер полученной выборки составляет 1000 прецедентов, из которых 80% — это обучающая выборка, а 20% — тестовая.

5.3. Результаты эксперимента

Для обучения модели на вход системе подается бинарная последовательность всех признаков описаний из обучающей выборки, которой сопоставлены соответствующие метки классов, имеющие вид бинарного вектора после преобразования.

Оценка качества работы алгоритма производилась с помощью критерия скользящего контроля вида случайные разбиения, когда разбиения выбираются случайно, независимо и равновероятно из всего множества возможных разбиений.

В ходе работы алгоритмов были получены результаты, представленные в табл. 3.

	Binary Relevance + SVM (Linear)	ML-kNN
Exact Match Ratio	0,69	0,43
Precision	0,85	0,68
Precision micro	0,89	0,87
Precision macro	0,74	0,56
Recall	0,83	0,61
Recall micro	0,84	0,64
Recal macro	0,71	0,46
F_1 -мера	0,83	0,62
F_1 -мера micro	0,84	0,74
F_1 -мера macro	0,72	0,48
1/0 Loss	0,3	0,63
Hamming Loss	0,005	0,01

Таблица 3: Полученные результаты

Помимо качественных результатов необходимо сравнить вычислительную сложность используемых алгоритмов. Сложность алгоритмов можно разделить на:

- вычислительную сложность для этапа обучения;
- вычислительную сложность для этапа тестирования.

Согласно статье [31] алгоритм Binary Relevance имеет временную вычислительную сложность $O(k \cdot F_B(m, d))$ для тренировки и $O(k \cdot F'_B(d))$ для тестирования, где q — это число классов, m — размер обучающей выборки, d — размерность вектора \vec{x} , а F_B и F'_B — это вычислительные сложности бинарного классификатора для этапов тренировки и тестирования соответственно.

В качестве бинарного классификатора в Binary Relevance используется линейный метод опорных векторов, реализован-

ный в пакете *sklearn.svm.LinearSVC* на основе *LIBLINEAR*, обладающий линейной временной сложностью $O(m)$.

Алгоритм ML-kNN имеет вычислительную сложность: $O(m^2d + qmk)$ для этапа тренировки, $O(md + qk)$ для этапа тестирования, где k — это число соседей [31].

Таким образом, видно, что метод, сводящий задачу к задаче бинарной классификации, — Binary Relevance, в котором используется классификатор на основе линейного метода опорных векторов, показал более хорошие качественные результаты, чем модифицированный алгоритм k ближайших соседей — ML-kNN, а также он обладает меньшей вычислительной сложностью, следовательно, подходит для решения задачи наивной квалификации преступных деяний.

Заключение

В данной работе была разработана архитектура информационного комплекса, предназначенного для сбора и анализа данных в рамках задачи квалификации преступных деяний, произведена ручная семантическая разметка главы 16 Уголовного кодекса РФ и апробированы методы классификации для решения задачи наивной квалификации преступлений. В отличие от существующих систем и технологий, разработанный подход предназначен для стран с непрецедентным правом.

В дальнейшем планируется выделить атрибуты для других статей Уголовного кодекса РФ, рассмотреть атрибуты, относящиеся к субъекту и субъективной стороне, а также реали-

зовать модуль для автоматического определения значений атрибутов в конкретном преступном деянии.

Список литературы

1. Прохоров Л. А., Прохорова М. Л. Уголовное право. М.: Юристъ, 1999. 480 с.
2. Корнеева А. В. Теоретические основы квалификации преступлений / под ред. А. И. Рарога. М.: ТК Велби, Проспект, 2008. 176 с.
3. Saeed U., Sarim M., Usmani A., Mukhtar A., Shaikh A. B., Raffat S. K. Application of machine learning algorithms in crime classification and classification Rule Mining // Research Journal of Recent Sciences. 2015. Vol. 4(3). P. 106 — 114.
4. Shojaee S., Mustapha A., Sidi F., Jabar M. A study on classification learning algorithms to predict crime status // International Journal of Digital Content Technology and its Applications. 2013. Vol. 7. P. 361 — 369.
5. Kianiv R., Mahdavi S., Keshavarzi A. Analysis and prediction of crimes by clustering and classification // International Journal of Advanced Research in Artificial Intelligence. 2015. Vol. 4. No. 8. P. 11 — 17.
6. Dahbur K., Muscarello T. Classification system for serial criminal patterns // Artificial Intelligence and Law. 2003. Vol. 11. P. 251 — 269.
7. Bhowmik R. Data Mining techniques in fraud detection // Journal of Digital Forensics, Security and Law. Vol. 3(2). P. 35 — 53.

8. Yu C., Ward M., Morabito M., Ding W. Crime forecasting using Data Mining techniques // Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops. 2011. P. 779 — 786.
9. Buczak A. L., Gifford C. M. Fuzzy association rule mining for community crime pattern discovery // ACM SIGKDD Workshop on Intelligence and Security Informatics. 2012. P. 1 — 10.
10. Brown D. The Regional Crime Analysis Program (RECAP): A framework for mining data to catch criminals // Proceedings of the International Conference on Systems, Man, and Cybernetics. 1998. P. 2848 — 2853.
11. Комиссаров В. С. Российское уголовное право. Общая часть. СПб.: Питер, 2005. 560 с.
12. Преступления против личности. Энциклопедия юриста. [Электронный ресурс]: URL:http://dic.academic.ru/dic.nsf/enc_law/1799 (дата обращения 31.01.2017)
13. Коробеев А. И. Полный курс уголовного права. Том II. Преступления против личности. СПб.: Издательство Р. Асланова «Юридический центр Пресс». 2008. 682 с.
14. Zhang M. L., Zhou Z. H. ML-KNN: A lazy learning approach to multi-label learning // Pattern Recognition. 2007. Vol. 40. P. 2038 — 2048.
15. Gao S., Wu W., Lee C.-H., Chua T.-S. A MFoM learning

- approach to robust multiclass multi-label text categorization // Proceedings of the 21st International Conference on Machine Learning. 2004. P. 329 — 336.
16. Kazawa H., Izumitani T., Taira H., Maeda E. Maximal margin labeling for multi-topic text categorization // Neural Information Processing Systems 17. 2005. P. 649 — 656.
 17. McCallum A. Multi-label text classification with a mixture model trained by EM // Working Notes of the AAAI'99 Workshop on Text Learning. 1999. 7 p.
 18. Schapire R. E., Singer Y. Boostexter: a boosting-based system for text Categorization // Machine Learning 39 (2/3). 2000. P. 135 —168.
 19. Comite F. D., Gilleron R., Tommasi M. Learning multi-label alternating decision tree from texts and data // Lecture Notes in Computer Science 2734. 2003. P. 35 — 49.
 20. Ueda N., Saito K. Parametric mixture models for multi-label text // Neural Information Processing Systems 15. 2003. P. 721 — 728.
 21. Elisseef A., Weston J. A kernel method for multi-labelled classification // Neural Information Processing Systems 14. 2002. P. 681 — 687.
 22. Clare A., King R. D. Knowledge discovery in multi-label phenotype data // Lecture Notes in Computer Science 2168. 2001. P. 42 — 53.

23. Boutell M. R., Luo J., Shen X., Brown C. M. Learning multi-label scene classification // Pattern Recognition 37 (9). 2004. P. 1757 — 1771.
24. Read J. A pruned problem transformation method for multi-label classification // New Zealand Computer Science Research Student Conference Proceedings, NZCSRS'08. 2008. P. 143 — 150.
25. Dietterich T. G., Bakiri G. Solving multiclass learning problems via error-correcting output codes // Artificial Intell. 1995. Vol. 2. P. 263 — 286.
26. Li X., Wang L., Sung E. Multilabel SVM active learning for image classification // International Conference on Image Processing (ICIP). 2004. Vol. 4. P. 2207 — 2010
27. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. [Электронный ресурс]: URL:http://www.machinelearning.ru/wiki/images/2/25/SMAIS11_SVM.pdf (дата обращения 10.04.2017)
28. Mercer J. Functions of positive and negative type and their connection with the theory of integral equations // Philos. Trans. Roy. Soc. London. 1909. Vol. A. No. 209. P. 415 — 446.
29. Godbole S., Sarawagi S. Discriminative methods for multi-labeled classification // PAKDD'04: 8th Pacific-Asia

Conference on Knowledge Discovery and Data Mining.
2004. P. 22 — 30.

30. Sorower Mohammad S. A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis. 2010. 25 p.
31. Zhang M.-L., Zhou Z.-H. A review on multi-label learning algorithms // IEEE Transactions on Knowledge and Data Engineering. 2014. Vol. 26. P. 1819 — 1837.
32. Halder C., Obaidullah S.M., Roy K. Offline writer identification from isolated characters using textural features // Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA). 2015. P. 221 — 231.