

Классификация объектов городского ландшафта по аэрофотоснимкам высокого разрешения

И. И. Юсипов

✉ igor.yusipov@gmail.com

А. И. Калякулина

✉ kalyakulina.alena@gmail.com

С. Ю. Филичева

✉ flicheva.s@gmail.com

19 декабря 2014

1. Введение

Информация об объектах городского ландшафта, полученная с помощью аэрофотосъемки, может использоваться для различных целей, таких как составление карт древесных насаждений [1], анализ зеленых зон [2], составление карт неиспользуемых площадей [3], обновление карт геоинформационных систем [4]. Данные о городском ландшафте также могут использоваться в градостроительстве [5]. Данные, получаемые из аэрофотоснимков, обычно классифицировались с использованием пиксельно-ориентированных подходов. Однако, для снимков высокого разрешения наблюдается сильное различие между объектами внутри одного класса и сходство между объектами разных классов, что приводит к низкой точности классификации при использовании только пиксельно-ориентированных подходов [6], [7], [8]. Объектно-ориентированные подходы, суть которых состоит в сегментации изображений на относительно однородные регионы, более предпочтительны для классификации и показывают более высокую точность благодаря объединению спектральной (например, среднее, стандартное отклонение) и неспектральной (например, текстура, размер, форма) информации о различных сегментах изображения [6], [7], [8], [9].

Сложность объектно-ориентированных методов состоит в выборе параметра сегментации, который определяет средний размер сегмента. Выбор параметра, который обеспечивает размер сегментов меньше размера объекта ландшафта, приводит к «пересегментации». Это нежелательно, потому что неспектральная информация (такая, как размер и форма), вычисляемая для сегментов становится бесполезной для классификации. В случае, когда параметр сегментации задает слишком большой размер сегментов, наблюдается «недосегментация» - сегменты могут содержать пиксели из более чем одного класса. Были проведены соответствующие исследования для выбора оптимального параметра сегментации [10], [11], [12], [13], [14], [15]. Основная сложность этих подходов заключается в том, что различные типы городского ландшафта могут быть хорошо классифицированы при различных значениях параметра сегментации, поэтому, если используется только один параметр сегментации для всего изображения, некоторые объекты могут быть «недосегментированы» или «пересегментированы». Для решения этой проблемы в некоторых исследованиях был использован подход многоуровневой сегментации, который основан на построении иерархии параметров сегментации и классификации различных типов городского ландшафта на основе экспертного знания [8], [12], [16], [17].

В данной иерархии сегменты более мелкого масштаба оказываются вложенными в сегменты более крупного масштаба (Рис. 1). Более крупные сегменты называются «супер-объекты», а более мелкие сегменты называются «суб-объекты» крупных сегментов [18]. Спектральная и несектральная информация (например, размер и форма) этих «супер-объектов» может быть полезна для классификации изображений. Например, спектральная информация самых мелких сегментов может быть полезна при определении отдельных деревьев, в то время как размер и форма крупных супер-объектов может быть полезна для разделения зданий и других поверхностей, похожих на крыши зданий. Основная проблема подхода с использованием одного сегментационного параметра состоит в том, что не все типы городского ландшафта хорошо сегментированы на одном и том же уровне. Преимуществами многоуровневой сегментации являются увеличение вероятности того, что хотя бы одно значение параметра будет соответствовать каждому типу ландшафта, и предоставление информации о том, как каждый объект ведет себя при различных значениях сегментационного параметра.

Основными задачи данной работы является сравнение различных методов классификации объектов городского ландшафта, в частности, деревьев решений и построения случайного леса.

2. Описание данных

Рассматриваемыми данными этой работы являются аэрофотоснимки местности в Deerfield Beach, FL, сделанные 31 декабря 2008 года. Изображения содержат 8-битные данные в инфракрасной (near infrared - NIR), красной и зеленой частях спектра размером примерно 1.4 км × 1.5 км (4630×4967 пикселей). Рассматриваемая площадь содержит такие типы городского ландшафта как строения, автомобили, деревья, газоны, грунт, бетон, асфальт, водоемы, тени. На Рис. 2 изображена рассматриваемая область.

Изображение рассматриваемой области было сегментировано с помощью алгоритма многоуровневой сегментации [18], который начинает с однопиксельного сегмента и объединяет соседние сегменты до тех пор, пока не будет достигнут некоторый «порог неоднородности» [20]. Этот порог определяется заданным пользователем параметром масштаба, который не имеет реального физического смысла (например, средняя площадь сегмента изображения), но в целом, увеличение значения параметра масштаба влечет за собой увеличение размеров сегмента (более грубая сегментация).

Для адекватного анализа изображения в различных масштабах была проведена серия сегментаций с различными значениями параметра масштаба (20 – 140 с интервалом 20). Указанный набор данных находится в открытом доступе [21]. Предварительно было установлено, что при значении параметра масштаба меньшего, чем 20, наблюдается пересегментация для всех классов объектов. При значении параметра масштаба большего, чем 140, наблюдается недосегментация. Интервал 20 был выбран исходя из соображений адекватности количества уровней сегментации и покрытия ими рассматриваемой поверхности.



Рис. 1. Различные уровни сегментации (серый - малый масштаб, черный - крупный масштаб) [19]

Всем трем спектральным полосам (инфракрасная, красная и зеленая) были присвоены одинаковые веса, так как все они могут содержать полезную информацию. Веса для параметров цвета и формы были установлены 0.9 и 0.1 соответственно, так, чтобы спектральная информация была более значимой для сегментации. Иными словами, цвет предполагаемого объекта имеет первостепенное значение. Эти веса характерны для анализа зашумленных и высокотекстурированных изображений (например, изображений, полученных с радиолокаторов) [22]. Веса для гладкости и компактности были установлены одинаковыми (0.5), чтобы не выделять компактные/некомпактные сегменты.

Для каждого сегмента изображения были рассчитаны спектральная (средние значения и отклонения для каждой рассматриваемой полосы спектра и нормализованный относительный индекс растительности — NDVI), текстурная (полутонная матрица смежности — GLCM, корреляция и энтропия для инфракрасного диапазона по все направлениям) информация, размер (площадь) и форма (округлость, индекс формы, индекс границы, длина, ширина, прямоугольное выравнивание, плотность, протяженность границы, асимметрия). Формулы для всех этих параметров могут быть найдены в работе [18]. Инфракрасная часть спектра была выбрана для расчета матрицы GLCM, потому что это наиболее подходит для классификации как растительных, так и техногенных объектов.

Выделены следующие параметры для каждого уровня сегментации:

- *Class* — Истинный класс, к которому принадлежит объект;
- *BrdIndx* — Индекс границы;
- *Area* — Площадь (m^2);
- *Round* — Округлость;
- *Bright* — Яркость;
- *Compact* — Компактность;
- *ShpIndx* — Индекс формы;
- *Mean_G* — Спектральная переменная для зеленой составляющей;
- *Mean_R* — Спектральная переменная для красной составляющей;
- *Mean_NIR* — Спектральная переменная для инфракрасной составляющей;
- *SD_G* — Отклонение зеленого;
- *SD_R* — Отклонение красного;

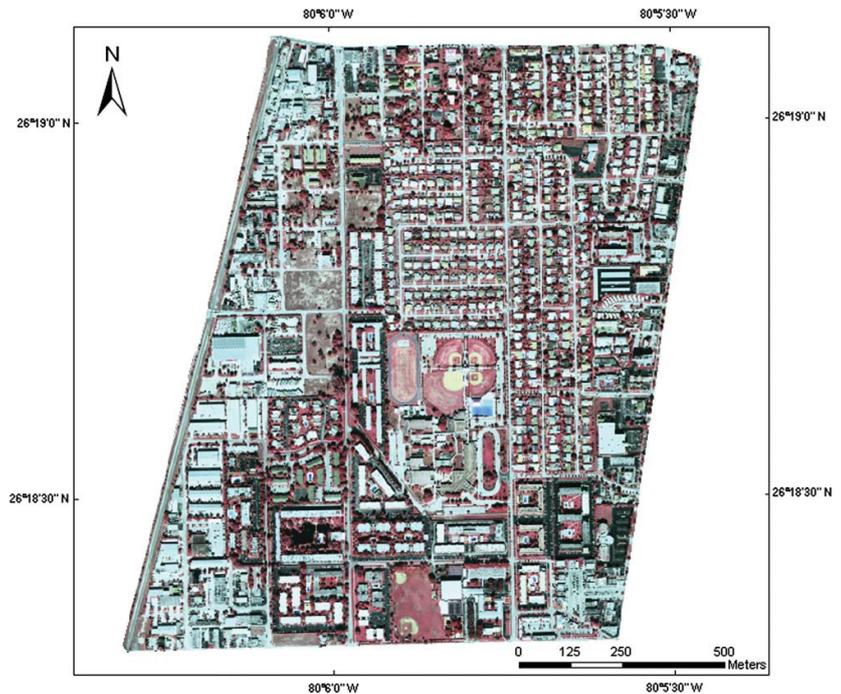


Рис. 2. Рассматриваемая область [19]

- *SD_NIR* — Отклонение инфракрасного;
- *LW* — Длина/ширина;
- *GLCM1, GLCM2, GLCM3* — Атрибуты полутоновой матрицы смежности;
- *Rect* — «Прямоугольность»;
- *Dens* — Плотность;
- *Assym* — Асимметрия;
- *NDVI* — Нормализованный относительный индекс растительности;
- *BordLength* — Длина границы

Тренировочная выборка содержит по 20 – 50 экземпляров из каждого класса: дерево, газон, строение, бетон, асфальт, машина, водоем, грунт и тень. Каждый элемент выборки описывается семью наборами параметров, полученных при различных значениях параметра масштаба (уровня сегментации). На Рис. 3 изображен пример экземпляра тренировочной выборки с различными уровнями сегментации.

Тренировочная выборка в общей сложности содержит 168 экземпляров, а общее количество параметров — 148. Тестовая выборка устроена аналогичным образом и содержит 507 экземпляров и такое же число параметров.

В работе использовались готовые выборки признаков с репозитория машинного обучения (UCI) [21].

3. Деревья решений

3.1. Общие принципы

Деревья решений – один из методов автоматического анализа данных. Основополагающей работой по данной тематике является [23].

Сначала необходимо уточнить основные понятия из теории деревьев решений, которые в дальнейшем будут употребляться (Таб. 1).

Деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение.

Основные классы задач, решаемые с помощью деревьев решений:

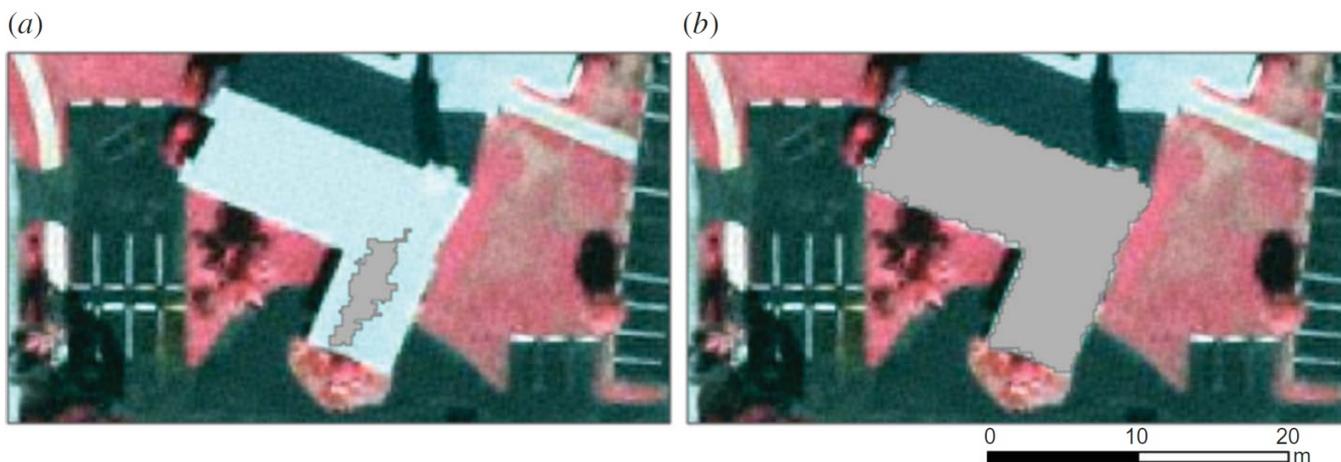


Рис. 3. Экземпляр тренировочной выборки для параметра масштаба 20 (a) и 80 (b) [19]

Таблица 1. Терминология деревьев решений [24]

Название	Описание
Объект	Пример, шаблон, наблюдение
Атрибут	Признак, независимая переменная, свойство
Метка класса	Зависимая переменная, целевая переменная
Узел	Внутренний узел дерева, узел проверки
Лист	Конечный узел дерева, узел решения
Проверка (test)	Условие в узле

- Описание данных: Деревья решений позволяют хранить информацию о данных в компактной форме. Вместо данных может храниться дерево решений, которое содержит точное описание объектов.
- Классификация: отнесение объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения.
- Регрессия: Если целевая переменная имеет непрерывные значения, деревья решений позволяют установить зависимость целевой переменной от входных переменных.

3.1.1. Схематический алгоритм построения дерева

Пусть задано некоторое обучающее множество T , содержащее объекты (примеры), каждый из которых характеризуется m атрибутами, причем один из них указывает на принадлежность объекта к определенному классу.

Пусть через $\{C_1, C_2, \dots, C_k\}$ обозначены классы (значения метки класса), тогда существуют 3 ситуации:

- 1) Множество T содержит один или более примеров, относящихся к одному классу C_k . Тогда дерево решений для T — это лист, определяющий класс C_k ;
- 2) Множество T не содержит ни одного примера, т.е. пустое множество. Тогда это снова лист, и класс, ассоциированный с листом, выбирается из другого множества отличного от T , например, из множества, ассоциированного с родителем;
- 3) Множество T содержит примеры, относящиеся к разным классам. В этом случае следует разбить множество T на некоторые подмножества. Для этого выбирается один из признаков, имеющий два и более отличных друг от друга значений O_1, O_2, \dots, O_n . T разбивается на подмножества T_1, T_2, \dots, T_n , где каждое подмножество T_i содержит все примеры, имеющие значение O_i для выбранного признака. Это процедура будет рекурсивно продолжаться до тех пор, пока конечное множество не будет состоять из примеров, относящихся к одному и тому же классу.

Вышеописанная процедура построения деревьев лежит в основе многих современных алгоритмов построения деревьев решений.

Для построения дерева на каждом внутреннем узле необходимо найти такое условие (проверку), которое бы разбивало множество, ассоциированное с этим узлом, на подмножества. В качестве такой проверки должен быть выбран один из атрибутов. Общее правило для выбора атрибута можно сформулировать следующим образом: выбранный атрибут должен разбить

множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому, т.е. количество объектов из других классов («примесей») в каждом из этих множеств было как можно меньше.

В дополнение к основному методу построения деревьев решений предлагаются следующие правила:

- Использование статистических методов для оценки целесообразности дальнейшего разбиения, так называемая «ранняя остановка» (prepruning). В конечном счете «ранняя остановка» процесса построения привлекательна в плане экономии времени обучения, но здесь уместно сделать одно важное предостережение: этот подход строит менее точные классификационные модели и поэтому ранняя остановка крайне нежелательна.
- Ограничить глубину дерева. Остановить дальнейшее построение, если разбиение ведет к дереву с глубиной, превышающей заданное значение.
- Разбиение должно быть нетривиальным, т.е. получившиеся в результате узлы должны содержать не менее заданного количества примеров.

3.1.2. Отсечение ветвей

Алгоритмы построения деревьев решений могут давать сложные деревья, которые имеют много узлов и ветвей. Такие «ветвистые» деревья разбивают обучающее множество на все большее количество подмножеств, состоящих из все меньшего количества объектов, и поэтому сложны для восприятия. Ценность правила, справедливого, например, для 2-3 объектов, крайне низка, и в целях анализа данных такое правило практически непригодно. Предпочтительнее иметь дерево, состоящее из малого количества узлов, которым бы соответствовало большое количество объектов из обучающей выборки. Возникает вопрос о построении всех возможных вариантов деревьев, соответствующих обучающему множеству, и выборе среди них дерева с наименьшей глубиной. Данная задача является NP-полной [26] и не имеет эффективных методов решения.

Для решения вышеописанной проблемы часто применяется отсечение ветвей (pruning).

Пусть под точностью (распознавания) дерева решений понимается отношение правильно классифицированных объектов при обучении к общему количеству объектов из обучающего множества, а под ошибкой – количество неправильно классифицированных. Предположим, что известен способ оценки ошибки дерева, ветвей и листьев. Тогда возможно использовать следующее правило:

- Построить дерево;
- Отсечь или заменить поддеревом те ветви, которые не приведут к возрастанию ошибки.

В отличие от процесса построения, отсечение ветвей происходит снизу вверх, начиная с листьев дерева, отмечая узлы как листья, либо заменяя их поддеревом.

3.1.3. Преимущества деревьев решений:

- Быстрый процесс обучения;
- Генерация правил в трудноформализуемых областях знаний;
- Извлечение правил на естественном языке;
- Интуитивно понятная классификационная модель;
- Высокая точность прогноза.

3.2. Алгоритм CART

Существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, ITrule, CHAID, CN2 и т.д.

В данной работе рассматривается CART [25] (Classification and Regression Tree) – это алгоритм построения бинарного дерева решений – дихотомической классификационной модели. Каждый узел дерева при разбиении имеет только двух потомков. Существует несколько модифицированных версий – алгоритмы IndCART и DB-CART. Алгоритм IndCART отличается от CART использованием иного способа обработки пропущенных значений. Он не осуществляет регрессионную часть алгоритма CART и имеет иные параметры отсечения. Алгоритм DB-CART базируется на следующей идее: вместо того чтобы использовать обучающий набор данных для определения разбиений, используем его для оценки распределения входных и выходных значений и затем используем эту оценку, чтобы определить разбиения. DB, соответственно, означает – «distribution based». Утверждается, что эта идея дает значительное уменьшение ошибки классификации по сравнению со стандартными методами построения дерева.

3.2.1. Бинарное представление дерева решений

В алгоритме CART каждый узел дерева решений имеет двух потомков. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров (обучающую выборку) на две части – часть, в которой выполняется правило (потомок – right) и часть, в которой правило не выполняется (потомок – left). Для выбора оптимального правила используется функция оценки качества разбиения.

3.2.2. Функция оценки качества разбиения

Обучение дерева решений относится к классу обучения с учителем, то есть обучающая и тестовая выборки содержат классифицированный набор примеров. Оценочная функция, используемая алгоритмом CART, базируется на интуитивной идее уменьшения нечистоты (неопределённости) в узле. В алгоритме CART идея «нечистоты» формализована в индексе *Gini*. Если набор данных T содержит данные n классов, тогда индекс *Gini* определяется как:

$$Gini(T) = 1 - \sum_{i=1}^n p_i^2, \quad (1)$$

где T – текущий узел, а p_i – вероятность класса i в узле T .

Если набор T разбивается на две части T_1 и T_2 с числом примеров в каждом N_1 и N_2 соответственно, тогда показатель качества разбиения будет равен:

$$Gini_{split}(T) = \frac{N_1}{N} Gini(T_1) + \frac{N_2}{N} Gini(T_2) \quad (2)$$

Наилучшим считается то разбиение, для которого $Gini_{split}(T)$ минимально.

Обозначим N – число примеров в узле – предке, L , R – число примеров соответственно в левом и правом потомке, l_i и r_i – число экземпляров i -го класса в левом/правом потомке. Тогда качество разбиения оценивается по следующей формуле:

$$Gini_{split}(T) = \frac{L}{N} \left(1 - \sum_{i=1}^n \left(\frac{l_i}{L} \right)^2 \right) + \frac{R}{N} \left(1 - \sum_{i=1}^n \left(\frac{r_i}{R} \right)^2 \right) \rightarrow \min \quad (3)$$

Данную задачу минимизации можно в итоге свести к задаче максимизации следующего выражения:

$$\tilde{G}_{split}(T) = \frac{1}{L} \sum_{i=1}^n l_i^2 + \frac{1}{R} \sum_{i=1}^n r_i^2 \rightarrow \max \quad (4)$$

3.2.3. Правила разбиения

Вектор предикторных переменных, подаваемый на вход дерева, может содержать как числовые (порядковые), так и категориальные переменные. В любом случае, в каждом узле разбиение идет только по одной переменной. Если переменная числового типа, то в узле формируется правило вида $x_i \leq c$, где c — некоторый порог, который чаще всего выбирается как среднее арифметическое двух соседних упорядоченных значений переменной x_i обучающей выборки. Если переменная категориального типа, то в узле формируется правило $x_i \in V(x_i)$, где $V(x_i)$ — некоторое непустое подмножество множества значений переменной x_i в обучающей выборке. Следовательно, для n значений числового атрибута алгоритм сравнивает $n - 1$ разбиений, а для категориального — $(2^{n-1} - 1)$. На каждом шаге построения дерева алгоритм последовательно сравнивает все возможные разбиения для всех атрибутов и выбирает наилучший атрибут и наилучшее разбиение для него.

3.2.4. Механизм отсечения дерева

Механизм отсечения дерева (Minimal cost-complexity tree pruning) — наиболее серьезное отличие алгоритма CART от других алгоритмов построения дерева. CART рассматривает отсечение как получение компромисса между двумя проблемами: получение дерева оптимального размера и получение точной оценки вероятности ошибочной классификации.

Основная проблема отсечения - большое количество всех возможных отсеченных поддеревьев для одного дерева. Более точно: если бинарное дерево имеет $|T|$ листов, тогда существует $\approx [1.5028369 |T|]$ отсечённых поддеревьев. И если дерево имеет хотя бы 1000 листов, то число отсечённых поддеревьев становится просто огромным.

Основная идея метода – не рассматривать все возможные поддеревья, ограничившись только «лучшими представителями».

3.2.5. Выбор финального дерева

Самым очевидным простым способом является выбор финального дерева через тестирование на тестовой выборке. Дерево, давшее минимальную ошибку классификации, и будет лучшим. Однако, качество тестирования во многом зависит от объема тестовой выборки и «равномерности данных», которые попали в обучающую и тестовую выборки. В случае, когда набор данных для обучения мал или каждая запись в нем по-своему «уникальна» так, что невозможно выделить выборку для обучения и выборку для тестирования, используется метод перекрестной проверки (V-fold cross-validation).

3.3. Применение деревьев решений к рассматриваемой задаче

Задача состоит в классификации объектов городского ландшафта по аэрофотоснимкам высокого разрешения, которые были перекодированы в различные типы параметров с разным уровнем сегментации объектов. Для классификации были использованы деревья решений, основанные на алгоритме CART. Основным инструментом являлся R [27] - язык программирования для статистической обработки данных и библиотека деревьев решений - «rpart» [28]. Полученные результаты сравнивались с данными, полученными с помощью MATLAB [29] - языка программирования для решения задач технических вычислений и класса ClassificationTree [30].

Для каждого уровня сегментации объектов было построено дерево-решение. В таблице 2 приведена достигнутая точность классификации и параметры, используемые при построении деревьев решений для каждого из уровней сегментации.

Таблица 2. Результаты построения деревьев решений для разного уровня сегментации

Параметр масштаба (уровень сегментации)	Значимые параметры, R	Достигнутая точность классификации на тестовой выборке, R (%)	Достигнутая точность классификации на тестовой выборке, Matlab (%)
20	Bright, Compact, NDVI, SD_G	68.27	72.19
40	BrdIndx, Bright, Mean_NIR, Mean_R, NDVI, SD_G	65.48	68.44
60	Area, Mean_NIR, Mean_R, NDVI, ShpIndx	65.88	70.61
80	BrdIndx, Bright, Mean_R, NDVI, SD_G	68.44	69.03
100	BordLngth, Bright, Compact, NDVI, SD_G	70.22	65.68
120	Bright, Dens, Mean_NIR, NDVI, SD_G, ShpIndx	61.93	58.78
140	Bright, Dens, Mean_NIR, NDVI, SD_G, ShpIndx	60.95	57.79
All scales	BrdIndx_80, Bright_80, Mean_R_40, NDVI, NDVI_40, Rect, SD_G	68.05	74.16

Полученные деревья решений изображены на рисунках 4 — 11. На узлах показаны условия ветвления дерева по соответствующему параметру. На листьях расположен вектор классов в лексикографическом порядке: asphalt/building/car/concrete/grass/pool/shadow/soil/tree. Значением каждого элемента этого вектора является число экземпляров конкретного класса тренировочной выборки, попавших в данный лист.

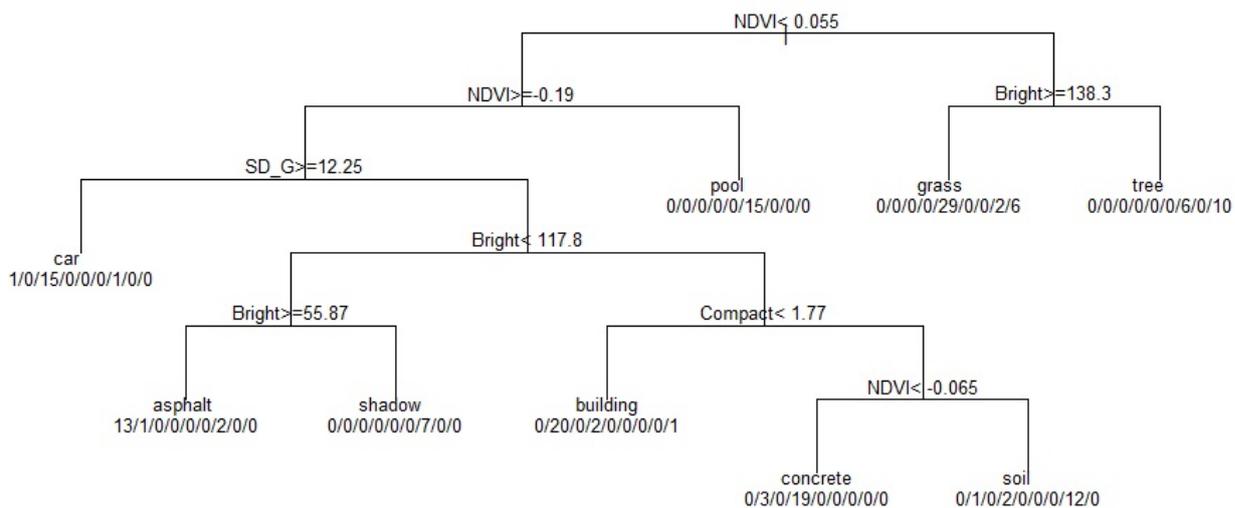


Рис. 4. Дерево решений для значения параметра масштаба 20

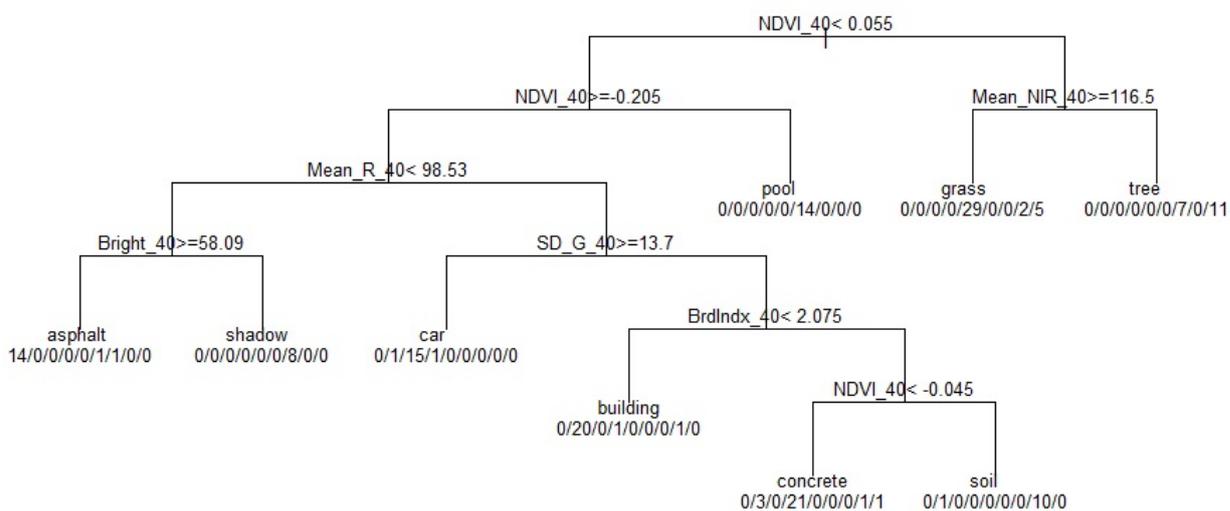


Рис. 5. Дерево решений для значения параметра масштаба 40

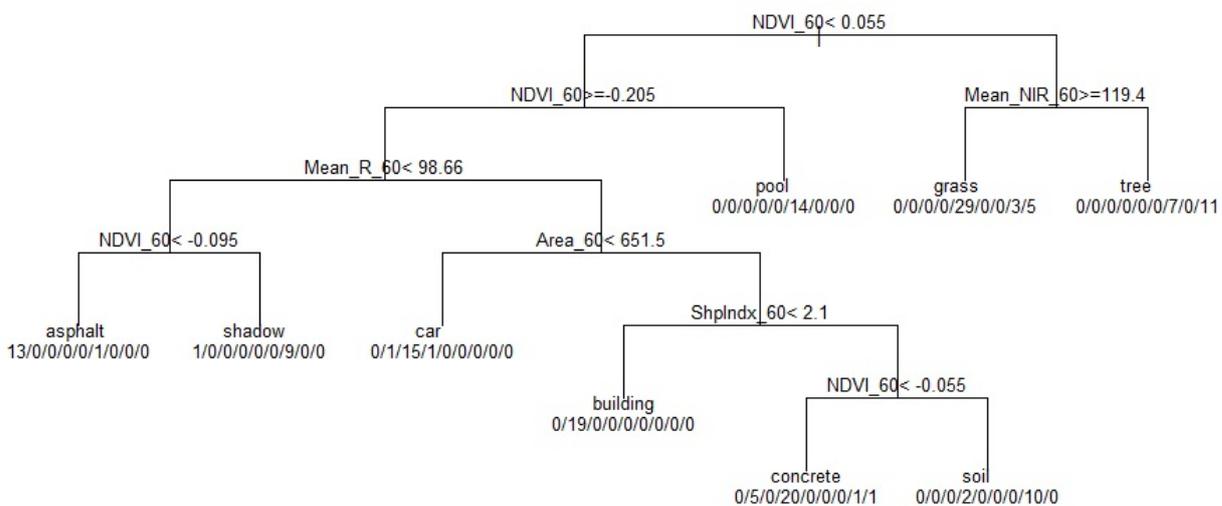


Рис. 6. Дерево решений для значения параметра масштаба 60

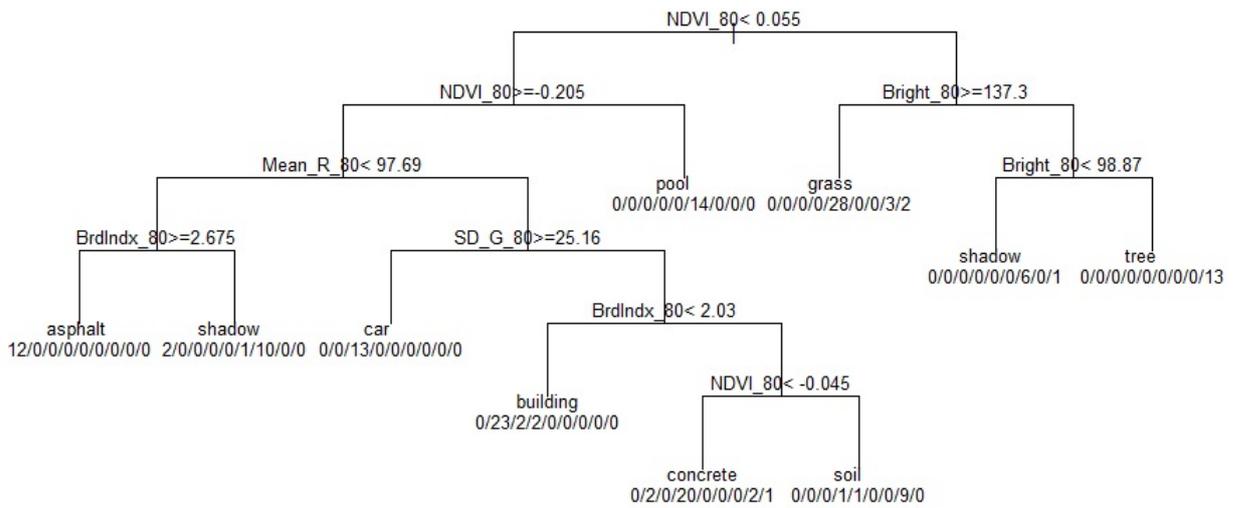


Рис. 7. Дерево решений для значения параметра масштаба 80

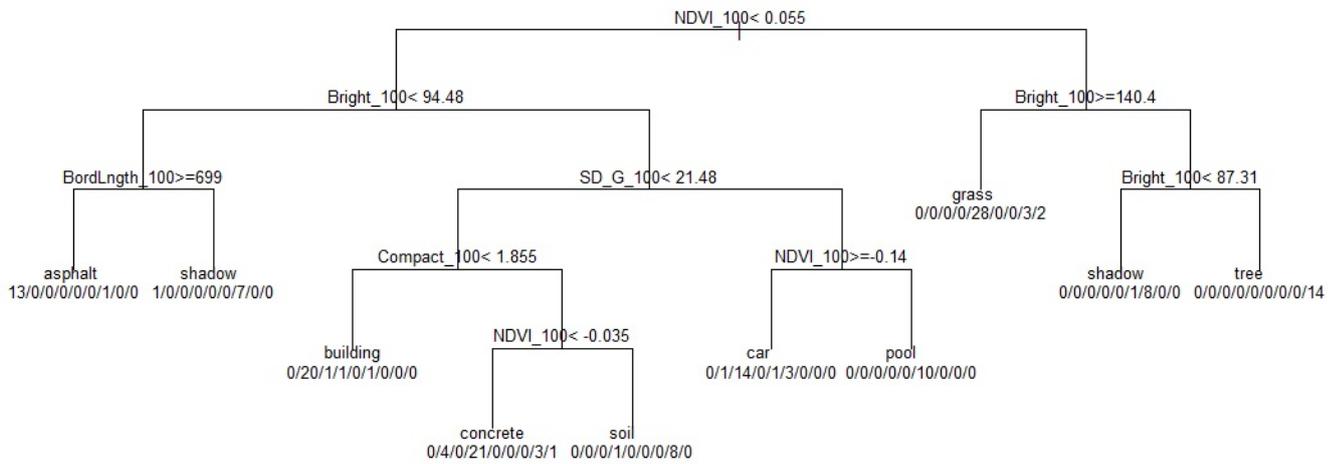


Рис. 8. Дерево решений для значения параметра масштаба 100

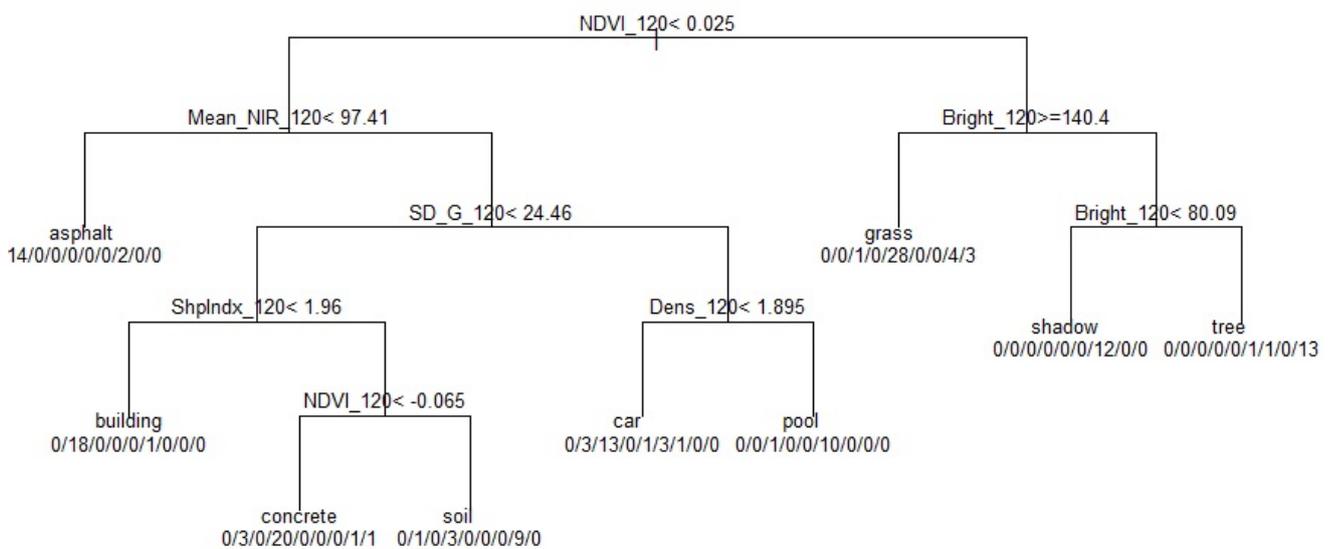


Рис. 9. Дерево решений для значения параметра масштаба 120

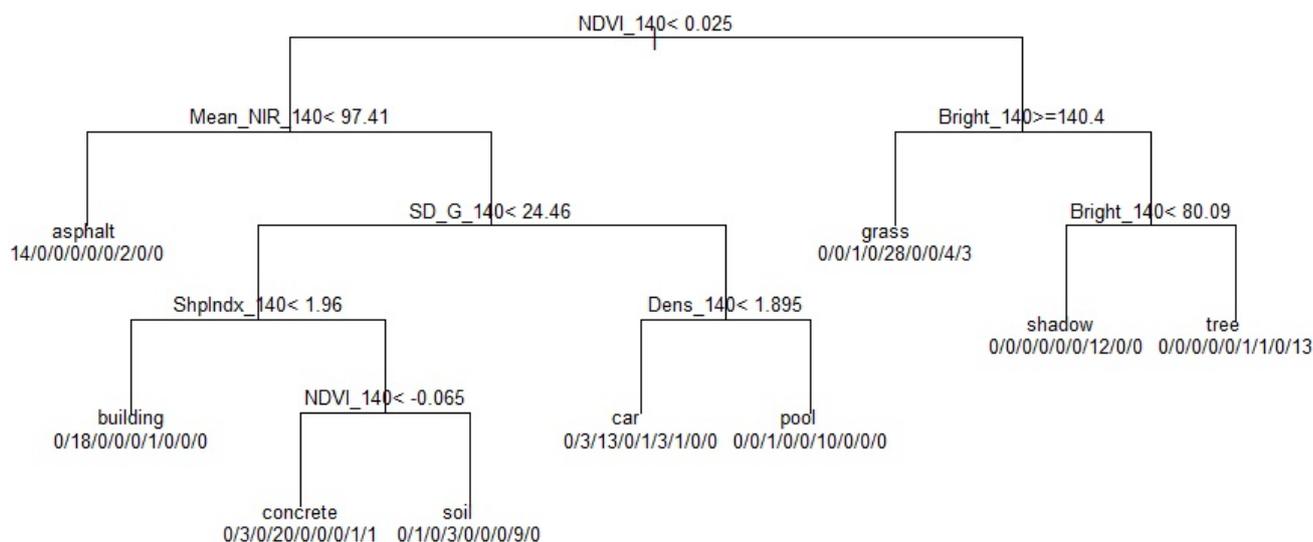


Рис. 10. Дерево решений для значения параметра масштаба 140

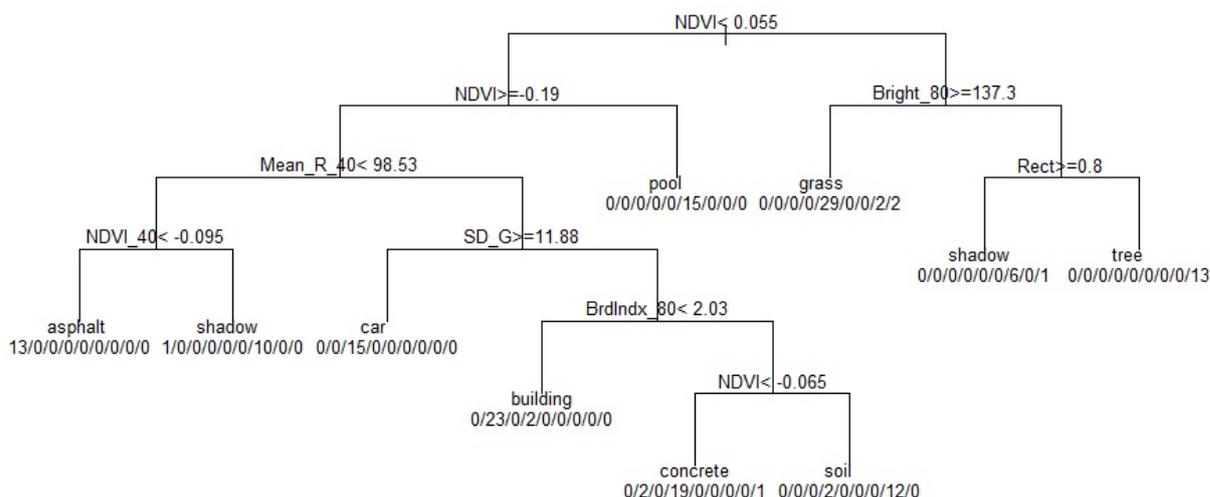


Рис. 11. Дерево решений для всех значений параметра масштаба

В ходе данных исследований было установлено, что для проведенной наивной классификации с использованием деревьев решений (алгоритм CART) оптимальный параметр масштаба (уровень сегментации) имеет значение 100. При более грубой сегментации точность классификации резко падает. Также был установлен закономерный факт, что при менее грубой сегментации лучше классифицируются объекты сравнительно небольшого размера (автомобили, деревья). Всегда относительно «хорошо» классифицируются объекты, принадлежащие классам водоем и газон. Оба этих класса находятся в зеленой части спектра, причем водоемы (синий цвет воды) находятся на внешней границе спектра, и текстуры этих объектов сравнительно однородны.

4. Случайный лес

4.1. Общие принципы

Метод основан на построении большого числа (ансамбля) деревьев решений (их число - параметр метода), каждое из которых строится по выборке, получаемой из исходной обучающей выборки с помощью bootstrap (выборки с возвращением) [31]. При построении каждого дерева на стадиях расщепления вершин используется только фиксированное число случайно отбирае-

мых признаков обучающей выборки (ещё один параметр метода) и строится полное дерево без отсечений. Классификация осуществляется с помощью голосования классификаторов, определяемых отдельными деревьями. Точность (вероятность корректной классификации) зависит от разнообразия классификаторов, составляющих ансамбль (от того, насколько коррелированы их решения) [32]. Чем более разнообразны классификаторы ансамбля (меньше коррелированность их решений), тем выше вероятность корректной классификации. В случайных лесах решения составляющих их деревьев слабо коррелированы вследствие двойной «случайности» в алгоритме построения случайного леса - на стадии выборки с возвращением и на стадии случайного отбора признаков, используемых при расщеплении вершин деревьев.

4.1.1. Ансамбли классификаторов

Ансамбль классификаторов представляет собой множество классификаторов, чьи решения комбинируются некоторым образом для получения окончательной классификации наблюдений. Обычно синтез решений отдельных классификаторов, составляющих ансамбль, осуществляется путем их голосования (возможно, взвешенного). Существует несколько групп методов построения ансамблей классификаторов [34]:

- Манипулирование примерами обучающей выборки;
- Манипулирование признаками;
- Включение случайности в индуктивный алгоритм;
- Манипулирование метками классов;
- Байесовское голосование.

Случайные леса синтезируют методы первых трех групп. Первая группа методов состоит либо в прогоне базового индуктивного алгоритма на различных подвыборках исходной обучающей выборки, либо в итеративном перевзвешивании наблюдений. Наиболее простой способ формирования подвыборок предложен Брейманом [35]. Метод основан на формировании обучающей выборки для каждого классификатора ансамбля с помощью bootstrap, то есть случайной выборки (того же объема, что и исходная обучающая выборка) с возвращением из исходной обучающей выборки и использовании метода голосования для агрегирования решений отдельных классификаторов. Метод получил название bagging или агрегированный bootstrap. Теоретический анализ [36], показал, что данный метод приводит к сокращению средней квадратичной ошибки классификации. Однако, это верно не для всех базовых моделей классификаторов. К методам этой группы относится также метод, основанный на формировании множества дизъюнктивных подвыборок обучающей выборки и использовании методики кросс-валидации [37] для формирования ансамбля классификаторов.

Метод, основанный на итеративном перевзвешивании наблюдений обучающей выборки, называется boosting [38]. Идея данного метода состоит в том, что классификаторы ансамбля строятся последовательно и на каждой итерации происходит коррекция (перевзвешивание) наблюдений обучающей выборки (на первой итерации веса всех наблюдений равны). Коррекция осуществляется таким образом, чтобы соответствующий классификатор делал меньше ошибок на тех наблюдениях, на которых часто делали ошибки классификаторы, построенные на предыдущих итерациях алгоритма. Кроме того, каждому классификатору приписывается некоторый вес исходя из количества допущенных им ошибок.

К методам второй группы относится тот [39], в котором исходное множество признаков разбивается на несколько дизъюнктивных подмножеств и строится ансамбль нейронных сетей, каждая из которых включает признаки только из одного подмножества.

Методы третьей группы основаны на том, что случайность вводится непосредственно в базовый алгоритм. Так, в работах [40], [41] алгоритм построения деревьев решений, составляющих ансамбль, был рандомизирован следующим образом: на каждом шаге расщепления вычислялось 20 наилучших расщеплений и затем осуществлялся случайный выбор одного из них.

4.1.2. Схематический алгоритм построения случайного леса

Алгоритм индукции случайного леса может быть представлен в следующем виде [42]:

- 1) Для $i = 1, 2, \dots, B$ (здесь B — количество деревьев в ансамбле) выполнить:
 - Сформировать bootstrap выборку S размера l по исходной обучающей выборке $D = \{x_i, y_i\}_{i=1}^l$;
 - По bootstrap выборке S индуцировать неусеченное дерево решений T_i с минимальным количеством наблюдений в терминальных вершинах, равным n_{min} , рекурсивно следуя следующему подалгоритму:
 - из исходного набора n признаков случайно выбрать p признаков;
 - из p признаков выбрать признак, который обеспечивает наилучшее расщепление;
 - расщепить выборку, соответствующую обрабатываемой вершине, на две подвыборки;

2) В результате выполнения шага 1 получаем ансамбль деревьев решений $\{T_i\}_{i=1}^B$;

3) Предсказание новых наблюдений осуществлять следующим образом:

- для регрессии:

$$\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{i=1}^B T_i(x); \quad (5)$$

- для классификации: пусть $\hat{\omega}_i(x) \in \{\omega_1, \omega_2, \dots, \omega_c\}$ — класс, предсказанный деревом решений T_i , то есть $T_i(x) = \hat{\omega}_i(x)$; тогда $\hat{\omega}_{rf}^B(x)$ — класс, наиболее часто встречающийся в множестве $\{\hat{\omega}_i(x)\}_{i=1}^B$.

4.1.3. Преимущества случайных лесов

- Высокая точность классификации;
- Гарантирует защиту от переобучения (ситуации, при которой классификатор хорошо классифицирует наблюдения обучающей выборки, но непригоден для классификации наблюдений, не входящей в неё) даже в случае, когда количество признаков значительно превышает количество наблюдений;
- Для построения случайного леса по обучающей выборке требуется задание всего двух параметров, которые требуют минимальной настройки;
- Метод Out-Of-Bag (OOB) [33] обеспечивает получение естественной оценки вероятности ошибочной классификации случайных лесов на основе наблюдений, не входящих в обучающие bootstrap выборки, используемые для построения деревьев;
- Обучающая выборка для построения случайного леса может содержать признаки, измеренные в разных шкалах: числовой, порядковой и номинальной, что недопустимо для многих других классификаторов.

4.2. Применение случайного леса к рассматриваемой задаче

Для классификации объектов городского ландшафта был использован алгоритм случайного леса. Инструментом также являлся язык программирования R [27] и библиотека случайного леса «randomForest» [43], являющаяся основным пакетом, реализующим классический алгоритм случайных лесов Бреймана. Результаты сравнивались с данными, полученными с помощью языка программирования MATLAB [29] с использованием класса TreeBagger [44].

Для каждого уровня сегментации объектов был построен случайный лес. В таблице 3 приведена достигнутая точность классификации и ООВ оценка вероятности ошибочной классификации случайного леса на основе наблюдений, не входящих в обучающие bootstrap выборки, используемые для построения деревьев.

Для одного значения параметра сегментации в состав случайного леса вошло 500 деревьев, каждое из которых использовало по 4 параметра. В случае объединения всех уровней сегментации число параметров для построения каждого из 500 деревьев равно 12.

Таблица 3. Результаты построения случайного леса для разного уровня сегментации

Параметр масштаба (уровень сегментации)	ООВ оценка вероятности ошибочной классификации, R (%)	Достигнутая точность классификации на тестовой выборке, R (%)	Достигнутая точность классификации на тестовой выборке, Matlab (%)
20	17.26	77.32	81.07
40	20.24	79.88	80.08
60	19.64	77.91	77.12
80	18.45	74.75	75.15
100	21.43	72.78	74.16
120	25.6	70.61	69.43
140	23.21	68.24	67.06
All scales	16.07	80.87	81.07

При использовании случайного леса была достигнута лучшая точность классификации в сравнении с применением одного дерева решений. Для всей совокупности параметров достигнута точность выше, чем при рассмотрении параметров, соответствующих отдельным уровням сегментации. Это может быть вызвано тем, что используется большее количество параметров при построении отдельных деревьев леса.

На рисунках 12 — 19 изображены зависимости ошибки по каждому из классов от числа деревьев в лесу на тренировочной выборке для каждого уровня сегментации.

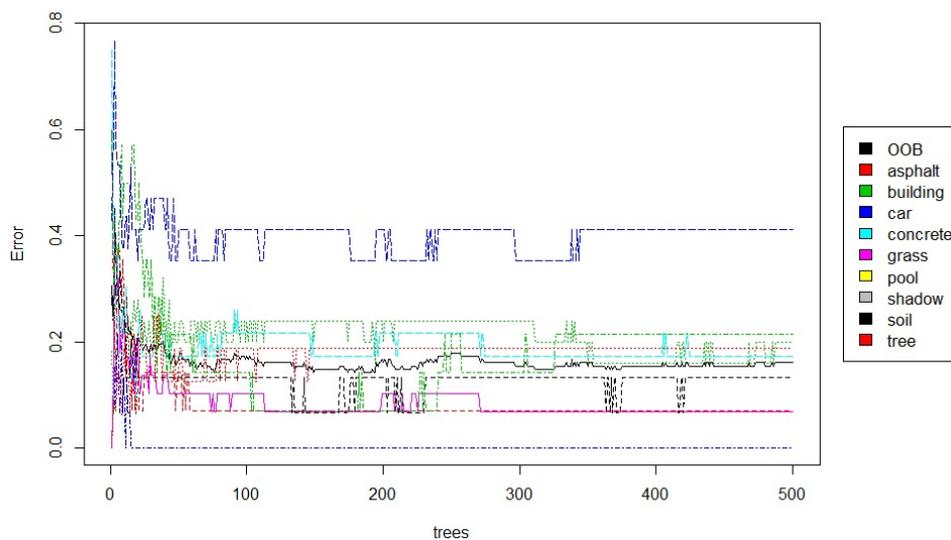


Рис. 12. Ошибки леса для каждого типа объекта при параметре масштаба 20

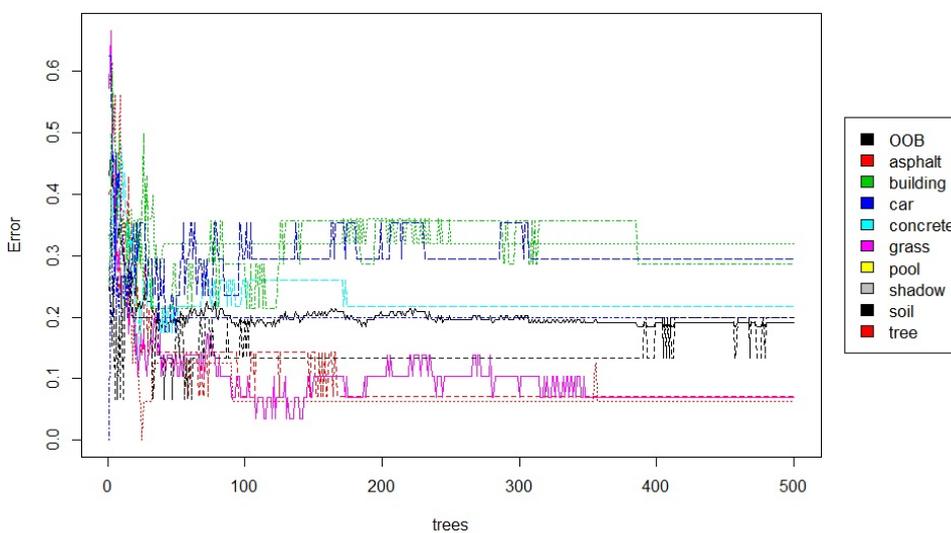


Рис. 13. Ошибки леса для каждого типа объекта при параметре масштаба 40

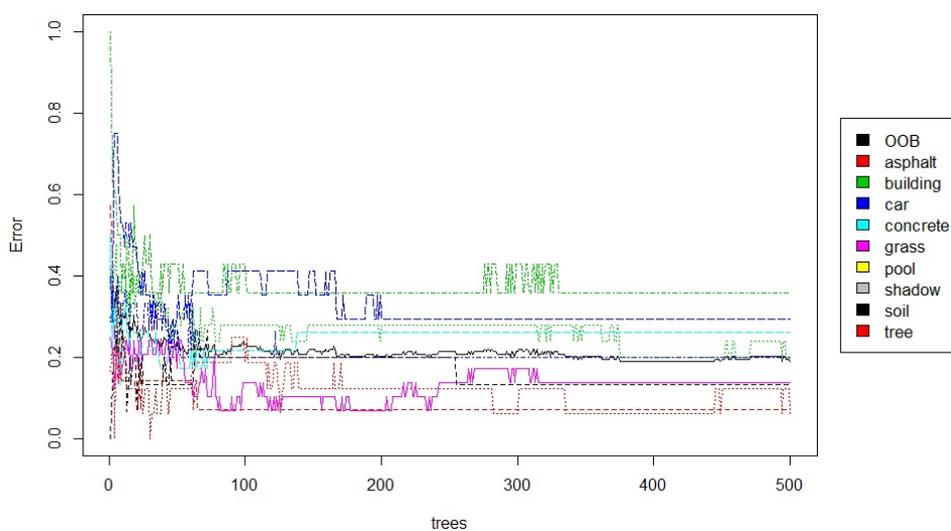


Рис. 14. Ошибки леса для каждого типа объекта при параметре масштаба 60

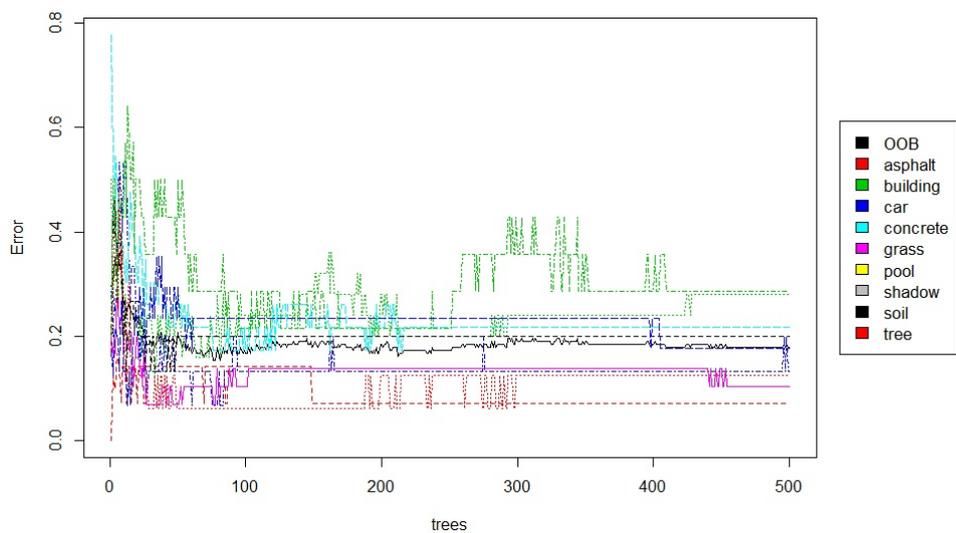


Рис. 15. Ошибки леса для каждого типа объекта при параметре масштаба 80

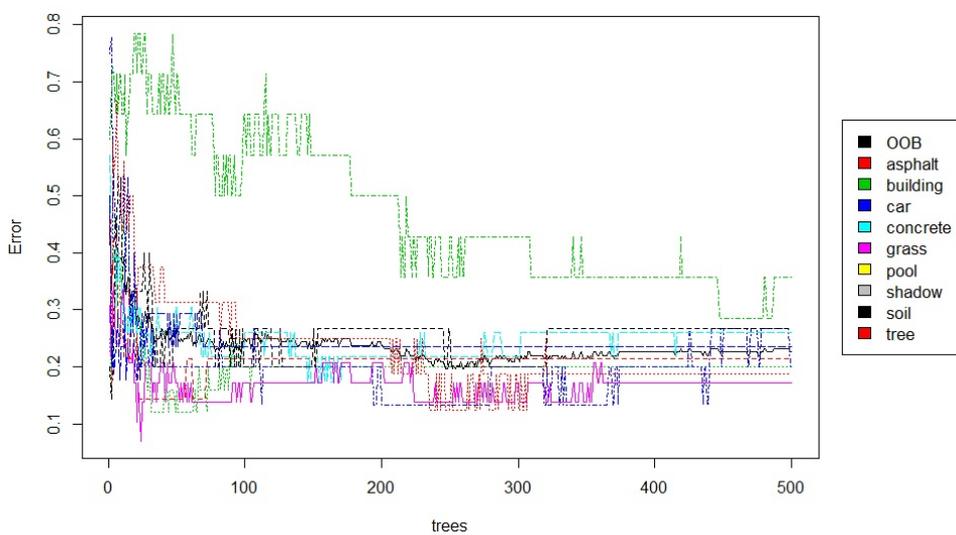


Рис. 16. Ошибки леса для каждого типа объекта при параметре масштаба 100

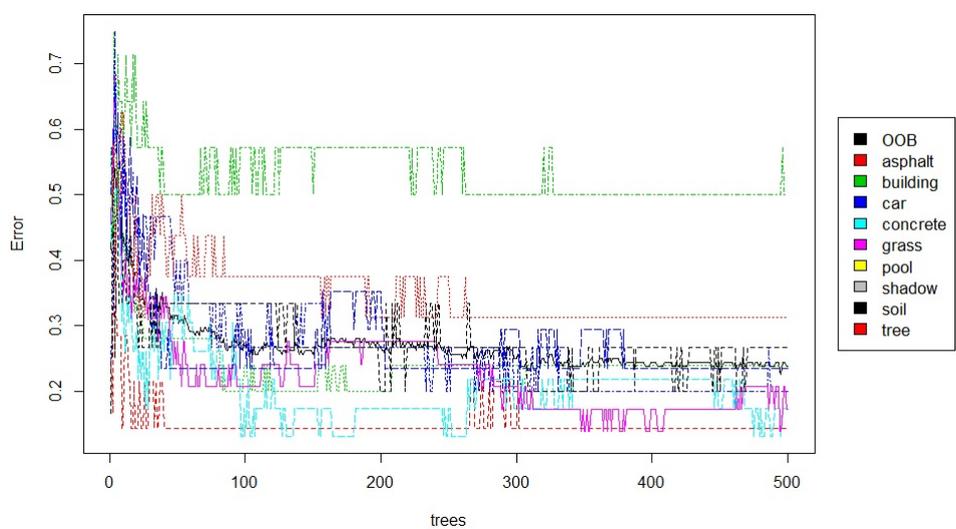


Рис. 17. Ошибки леса для каждого типа объекта при параметре масштаба 120

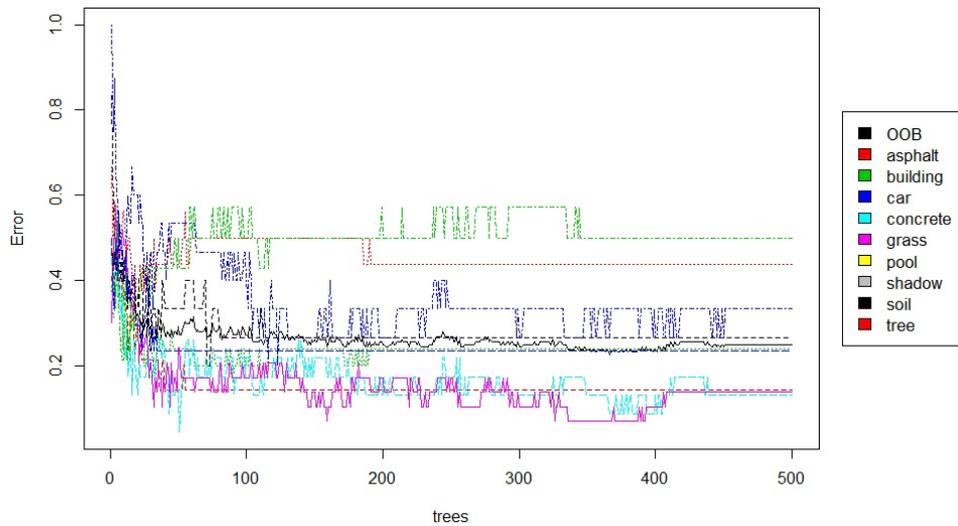


Рис. 18. Ошибки леса для каждого типа объекта при параметре масштаба 140

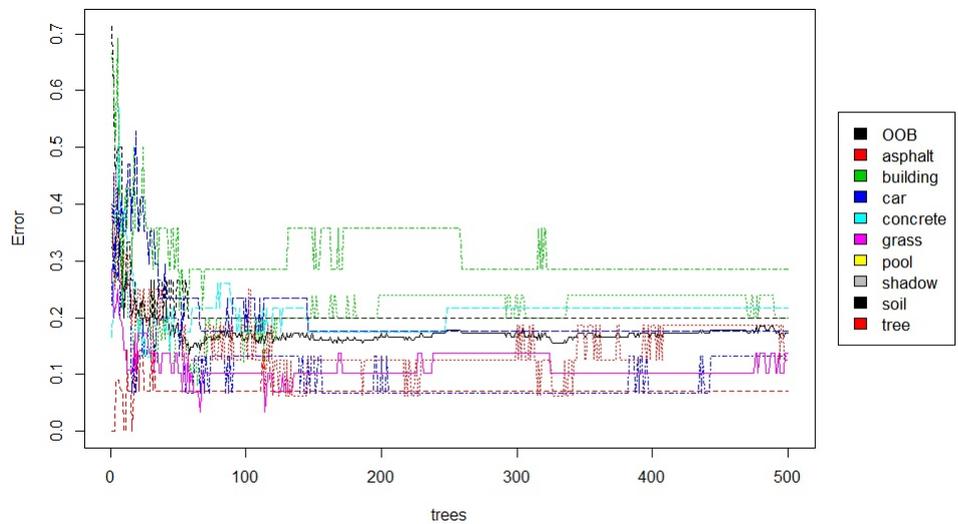


Рис. 19. Ошибки леса для каждого типа объекта при всех значениях параметра масштаба

Также были получены различные значения точности классификации в зависимости от выбранных параметров метода. На Рис. 20 представлена зависимость точности классификации от числа деревьев случайного леса B на выборке, включающей данные со всеми параметрами масштаба. Можно видеть, что, начиная примерно с $B = 100$, точность классификации колеблется около одного значения. С увеличением числа деревьев амплитуда колебаний уменьшается, так как полученный результат всё меньше зависит от отдельных случайных деревьев, но увеличивается время вычислений. Установленное по умолчанию значение $B = 500$ является хорошим компромиссом между ожидаемой точностью и временем работы метода.

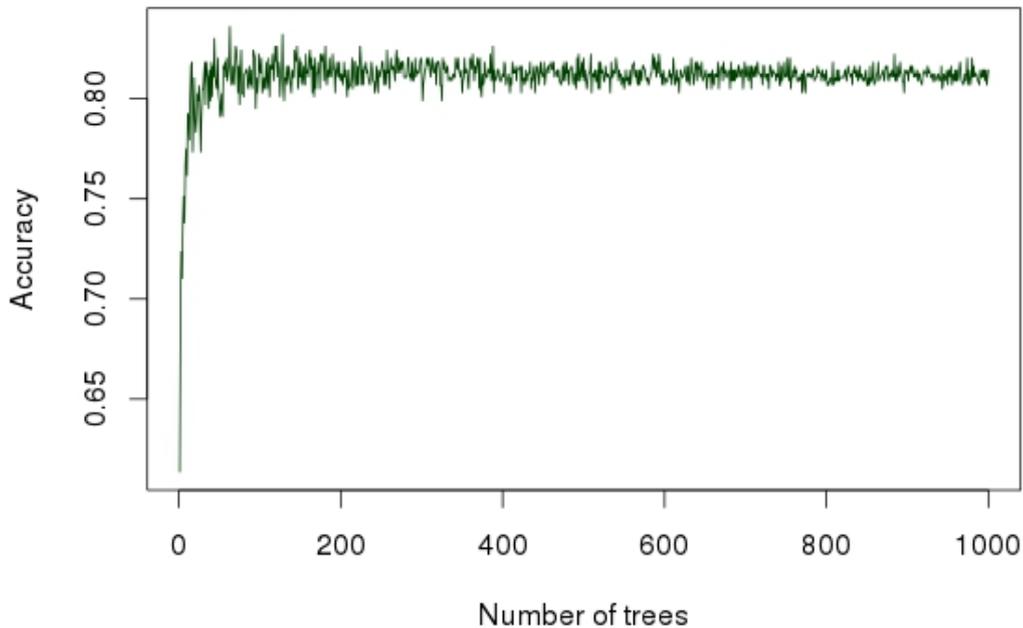


Рис. 20. Зависимость точности классификации от числа деревьев случайного леса на выборке, включающей данные со всеми параметрами масштаба. График построен по 1000 точек.

Таким же образом ведет себя точность классификации на выборках, включающих только данные с определенными параметрами масштаба. Однако, на Рис. 21 можно видеть, что значения, около которых она колеблется, неодинаковы для разных параметров масштаба. Наилучшая точность классификации достигается при использовании всех данных. Это обусловлено тем, что объекты определенных классов лучше распознаются на снимках с одними параметрами масштаба и хуже на снимках с другими. По данным Рис. 22 и Таб. 4 можно сделать следующие выводы:

- значение параметра масштаба, равное 20, позволяет с достаточной точностью распознавать объекты наибольшего числа классов;
- при значениях параметра масштаба, равных 140 и 120, достаточно успешно определяются только объекты класса concrete (бетонные конструкции);
- объекты класса building (строения) плохо распознаются при любых параметрах масштаба (возможно, это связано с тем, что здания могут сильно отличаться друг от друга);
- объекты класса tree (деревья) достаточно хорошо распознаются только при значении параметра масштаба, равном 20.

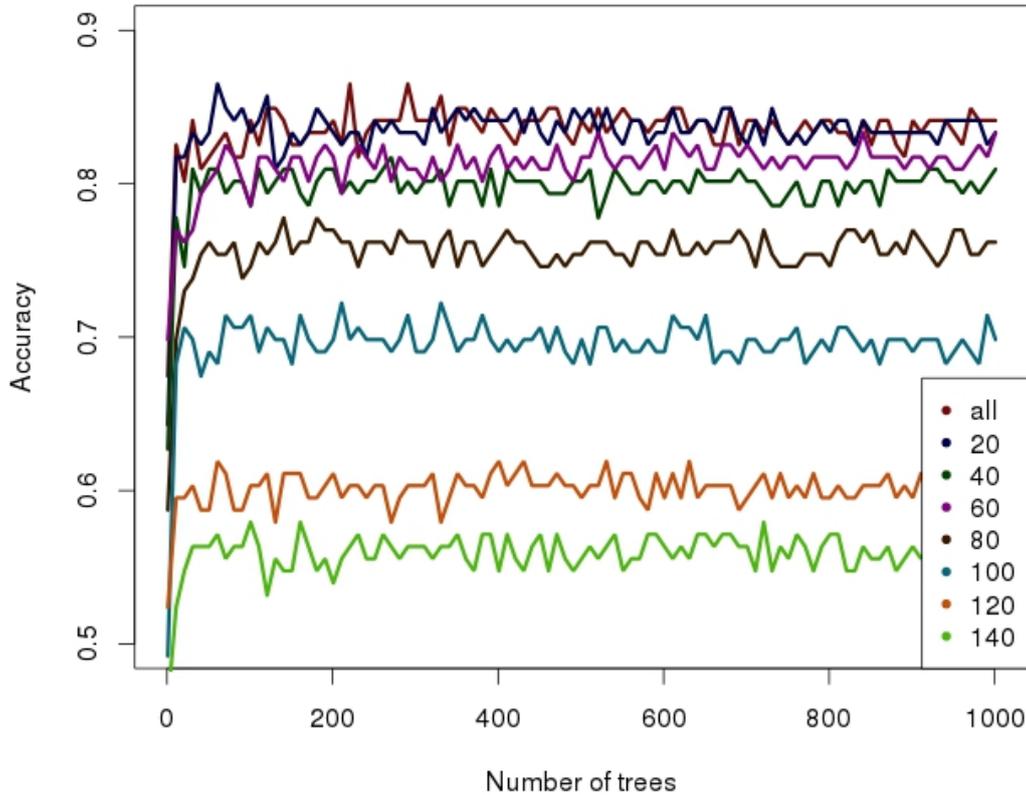


Рис. 21. Зависимость точности классификации от числа деревьев случайного леса на выборках, включающих только данные с определенными параметрами масштаба. Каждый график построен по 100 точкам.

Таблица 4. Точности распознавания объектов определенных классов на данных с разными параметрами масштаба, превышающие 0,8.

Параметр масштаба	concrete	shadow	tree	asphalt	building	grass	pool	car	soil
20		✓	✓	✓			✓	✓	✓
40		✓		✓		✓	✓	✓	
60	✓	✓				✓		✓	
80	✓	✓				✓		✓	
100	✓	✓							
120	✓								
140	✓								

Кроме этого была исследована зависимость точности классификации от параметра p - количества случайно выбираемых признаков, используемых для построения отдельных деревьев леса. На Рис. 23 можно видеть, что, начиная с некоторого момента, точность классификации уменьшается с увеличением параметра p . Для большинства наборов данных удачным является значение $p = 5$.

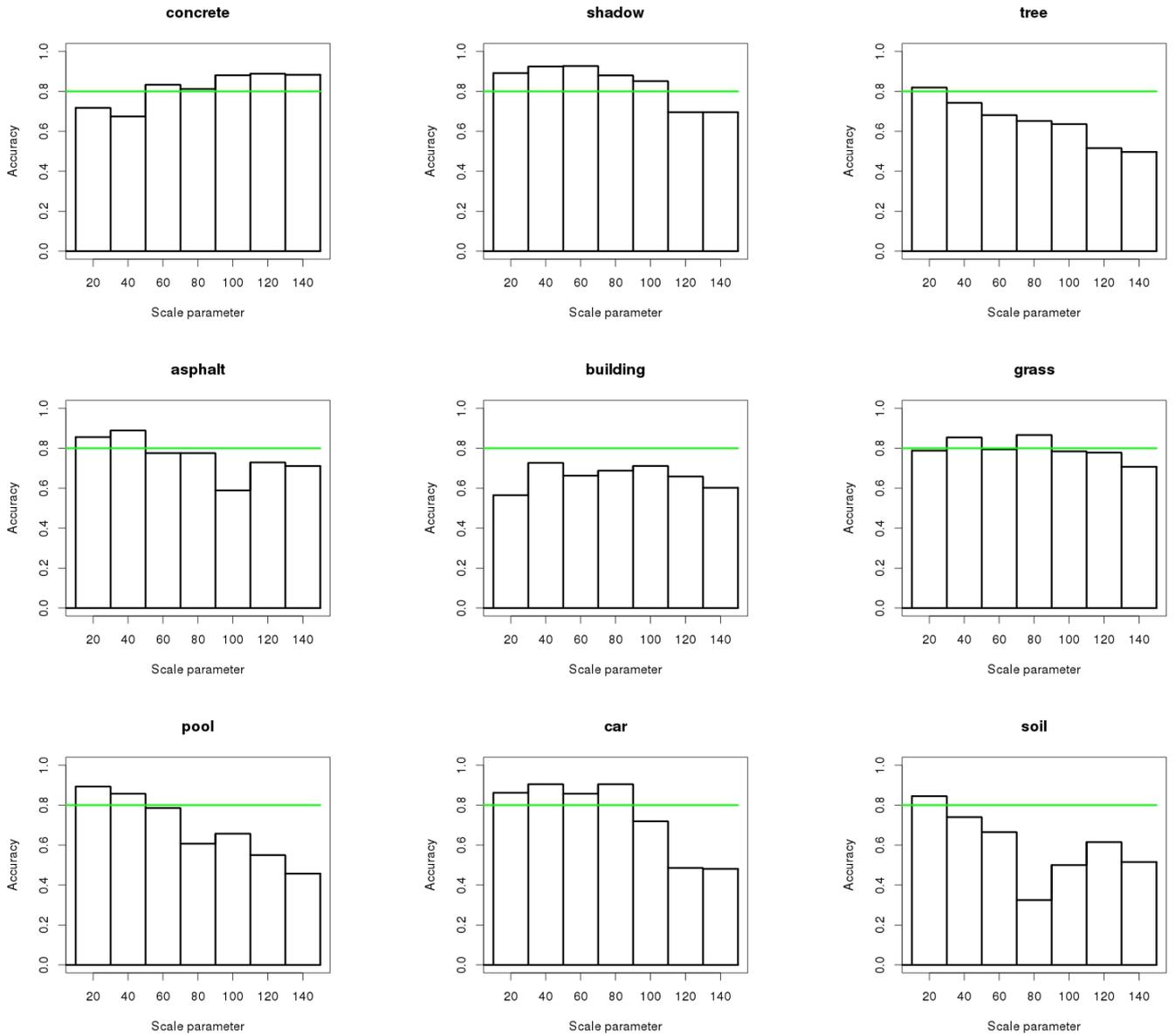


Рис. 22. Средняя точность распознавания объектов каждого из классов случайным лесом из 500 деревьев для наборов данных с разными параметрами масштаба.

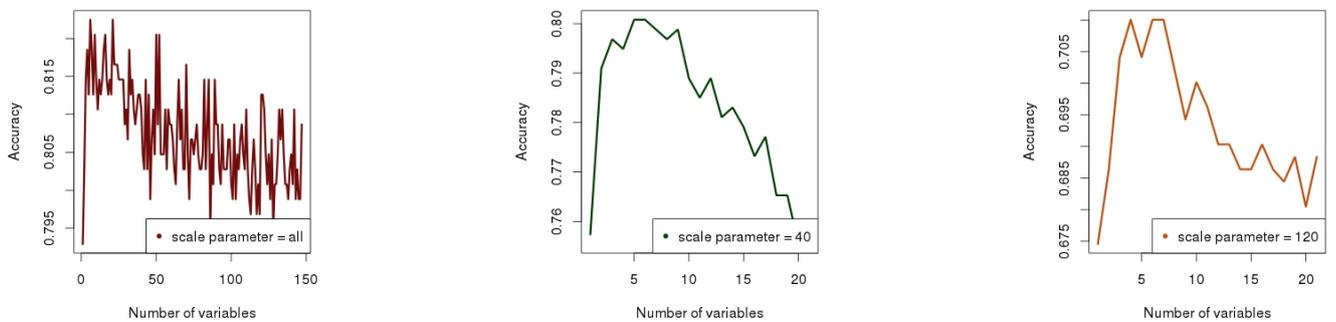


Рис. 23. Зависимость точности классификации от значения параметра p для наборов данных с различными параметрами масштаба.

5. Метод опорных векторов (SVM)

5.1. Общие принципы

Рассматривается задача обучения по прецедентам $\langle X, Y, y^*, X^l \rangle$, где X — пространство объектов, Y — множество ответов, $y^* : X \rightarrow Y$ — целевая зависимость, значения которой известны только на объектах обучающей выборки $X^l = (x_i, y_i)_{i=1}^l, y_i = y^*(x_i)$. Строится алгоритм $a : X \rightarrow Y$, аппроксимирующий целевую зависимость на всём пространстве X [45].

В базовом алгоритме рассматривается задача классификации на два непересекающихся класса, в которой объекты описываются n -мерными вещественными векторами: $X = \mathbb{R}^n, Y = \{-1, +1\}$.

Строится линейный пороговый классификатор:

$$a(x) = \text{sign} \left(\sum_{j=1}^n \omega_j x^j - \omega_0 \right) = \text{sign} (\langle \omega, x \rangle - \omega_0), \quad (6)$$

где $x = (x^1, \dots, x^n)$ — признаковое описание объекта x ; вектор $\omega = (\omega^1, \dots, \omega^n) \in \mathbb{R}^n$ и скалярный порог $\omega_0 \in \mathbb{R}$ являются параметрами алгоритма.

Уравнение $\langle \omega, x \rangle = \omega_0$ описывает гиперплоскость, разделяющую классы в пространстве \mathbb{R}^n .

Предположим, что выборка линейно разделима, то есть существуют такие значения параметров ω, ω_0 , при которых функционал числа ошибок

$$Q(\omega, \omega_0) = \sum_{i=1}^l [y_i (\langle \omega, x_i \rangle - \omega_0) < 0] \quad (7)$$

принимает нулевое значение. В этом случае разделяющая гиперплоскость не единственна, поскольку существуют и другие положения разделяющей гиперплоскости, реализующие то же самое разбиение выборки. Идея метода заключается в том, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней точек обоих классов. Этот принцип получил мощное теоретическое обоснование [46], [47], [48].

Параметры линейного порогового классификатора определены с точностью до нормировки: алгоритм $a(x)$ не изменится, если ω и ω_0 одновременно умножить на одну и ту же положительную константу. Удобно выбрать эту константу таким образом, чтобы для всех пограничных (ближайших к разделяющей гиперплоскости) объектов x_i из X^l выполнялись условия:

$$\langle \omega, x \rangle - \omega_0 = y_i. \quad (8)$$

При оптимальном положении разделяющей гиперплоскости все пограничные объекты находятся от неё на одинаковом расстоянии, остальные объекты находятся дальше. Таким образом, для всех $x_i \in X^l$:

$$\langle \omega, x \rangle - \omega_0 \begin{cases} \leq -1, & \text{если } y_i = -1; \\ \geq 1, & \text{если } y_i = +1. \end{cases} \quad (9)$$

Условие $-1 < \langle \omega, x \rangle - \omega_0 < 1$ задаёт полосу, разделяющую классы. Ни одна из точек обучающей выборки не может лежать внутри этой полосы. Границами полосы служат две параллельные гиперплоскости с направляющим вектором ω . Точки, ближайšie к разделяющей гиперплоскости, лежат в точности на границах полосы. При этом сама разделяющая гиперплоскость проходит ровно посередине полосы.

Чтобы разделяющая гиперплоскость как можно дальше отстояла от точек выборки, ширина полосы должна быть максимальной. Пусть x_- и x_+ — две произвольные точки классов -1 и $+1$ соответственно, лежащие на границе полосы. Тогда ширина полосы:

$$\langle (x_+ - x_-), \frac{\omega}{\|\omega\|} \rangle = \frac{\langle \omega, x_+ \rangle - \langle \omega, x_- \rangle}{\|\omega\|} = \frac{(\omega_0 + 1) - (\omega_0 - 1)}{\|\omega\|} = \frac{2}{\|\omega\|}. \quad (10)$$

Ширина полосы максимальна, когда норма вектора ω минимальна. В итоге требуется найти такие значения параметров ω и ω_0 , при которых норма вектора ω минимальна при условии 9.

5.2. Применение метода опорных векторов к рассматриваемой задаче

Для классификации объектов городского ландшафта был использован метод опорных векторов. Так как задача мультиклассовая, применялось решающее правило, основанное на разбиении задачи на бинарные по схеме «один против остальных». Инструментом также являлся язык программирования R [27] и библиотека метода опорных векторов «e1071» [49].

Для каждого уровня сегментации объектов была построена система опорных векторов. В таблице 5 приведена достигнутая точность классификации и количество опорных векторов, построенных в ходе работы алгоритма.

Таблица 5. Результаты построения системы опорных векторов для разного уровня сегментации

Параметр масштаба (уровень сегментации)	Достигнутая точность классификации на тестовой выборке (%)	Количество опорных векторов
20	73.17	152
40	71.00	150
60	70.80	155
80	70.21	155
100	67.85	156
120	66.27	159
140	63.11	159
All scales	75.73	159

Также для иллюстрации приведены примеры графиков — проекций всего пространства параметров на плоскости «Bright_80 - BrdIndx_80» (Рис. 24) и «NDVI - Mean_R» (Рис. 25). Данные параметры являются наиболее значимыми при классификации с помощью деревьев решений для всех значений параметра сегментации.

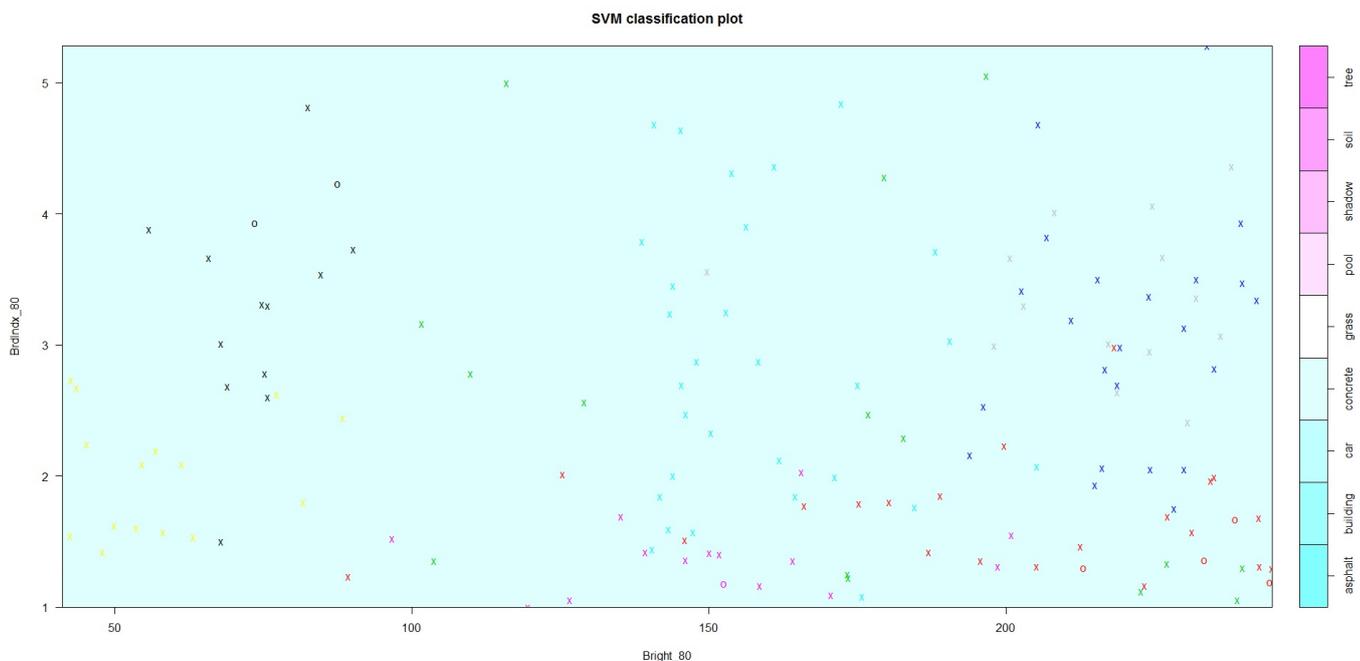


Рис. 24. Проекция пространства параметров на плоскость «Bright_80 - BrdIndx_80»

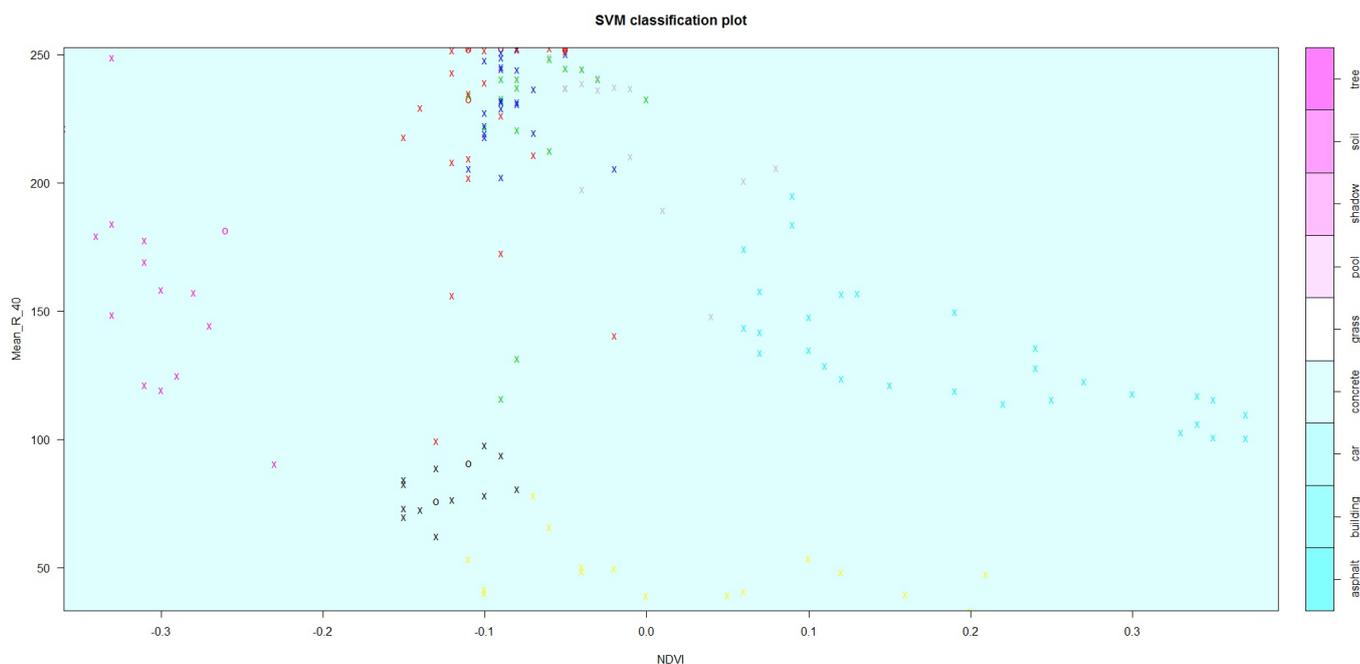


Рис. 25. Проекция пространства параметров на плоскость «NDVI - Mean_R_40»

6. Нейронные сети

6.1. Общие принципы

Искусственная нейронная сеть — математическая модель, а также её программная или аппаратная реализация, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Искусственная нейронная сеть представляют собой систему соединённых и взаимодействующих между собой искусственных нейронов. Каждый нейрон получает и передает сигналы другим

нейронам. Нейроны делятся на входные, выходные и нейроны скрытого слоя. На входные нейроны подаются данные, которые требуется проанализировать, а с выходных нейронов поступает отклик сети.

В данной работе рассматривались только многоуровневые сети прямого распространения, то есть такие, в которых связи между нейронами не образуют циклов, а узлы сети можно разбить на слои, которые удастся пронумеровать так, что дуги будут вести только от вершин из i -го слоя в вершины $(i + 1)$ -го слоя.

Функция ошибки (E) — функция, характеризующая разницу между требуемым ответом и ответом, полученным от нейронной сети. Функция активации ($g_j(x_1, x_2, \dots, x_q)$, где x_1, \dots, x_q — входы j -го нейрона) — функция, в соответствии с которой нейроны скрытого слоя преобразуют взвешенную сумму своих входов.

Обучение искусственной нейронной сети осуществляется итеративно путем изменения весов связей между нейронами на величины, зависящие от значения функции ошибки.

6.2. Нейронные сети с одним скрытым слоем (пакет nnet)

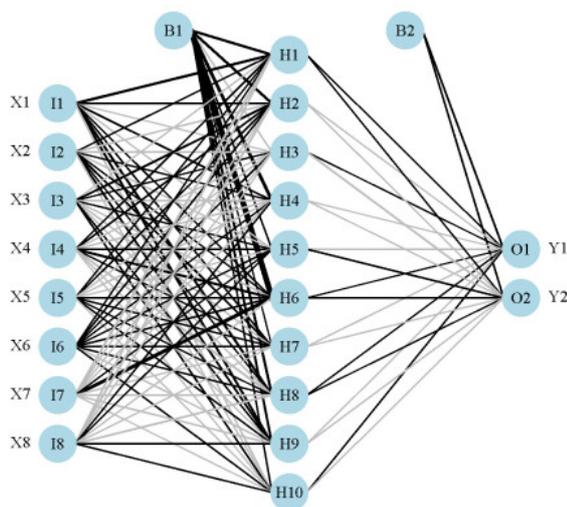


Рис. 26. Схема нейронной сети с одним скрытым слоем

Схема нейронной сети реализованного в пакете nnet типа представлена на Рис. 26. Использовались логистическая функция активации:

$$g_j(x_1, x_2, \dots, x_q) = \frac{1}{1 + e^{\beta_0 + \sum_{i=1}^q \beta_i x_i}},$$

и квадратичная функция ошибки:

$$E = \sum_{i=1}^n (y_i - d_i)^2,$$

где y_i — отклик i -го выходного нейрона, d_i — ожидаемый отклик. Для обучения сети использовался алгоритм обратного распространения ошибки, характеризующийся следующим правилом изменения весов:

$$\beta_k^{(t+1)} = \beta_k^{(t)} - \eta \cdot \frac{\partial E^{(t)}}{\partial \beta_k^{(t)}},$$

где t — номер шага обучения.

6.3. Нейронные сети с несколькими скрытыми слоями (пакет neuralnet)

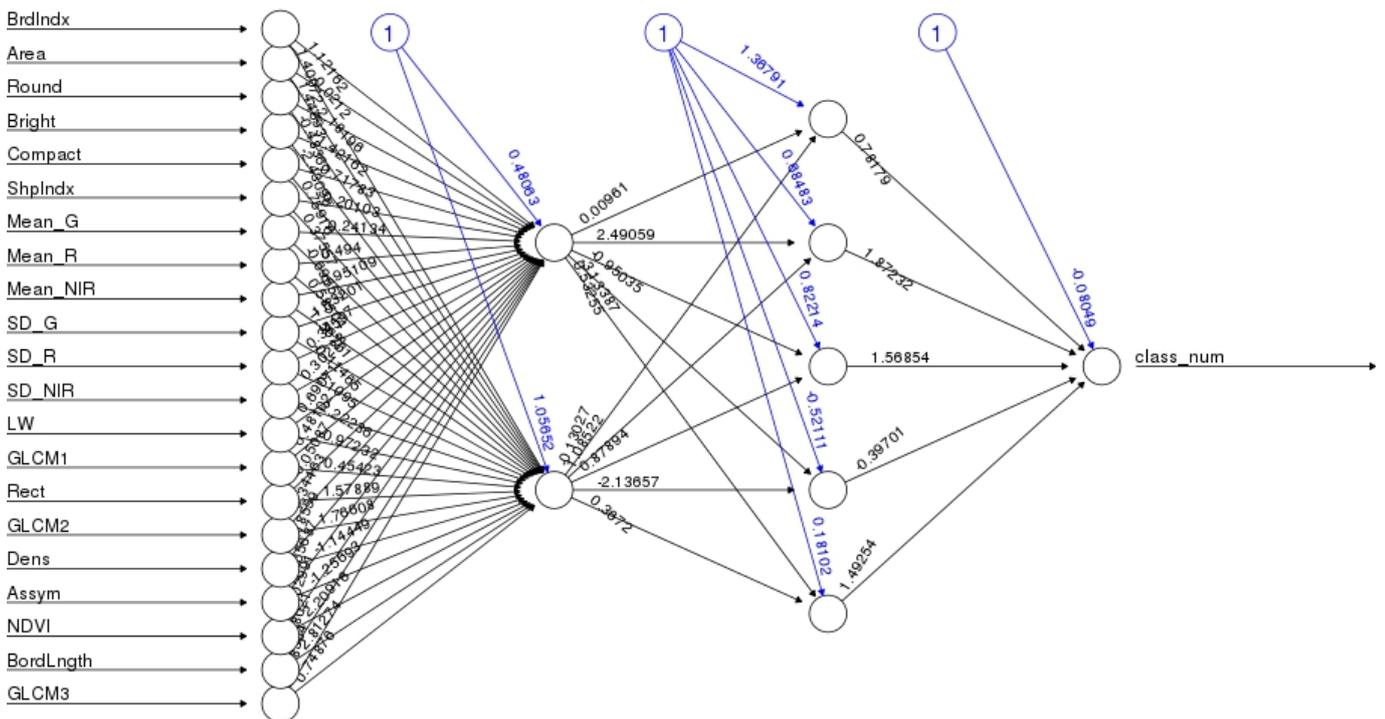


Рис. 27. Схема нейронной сети с двумя скрытыми слоями

Пакет `neuralnet` дает возможность использовать нейронные сети с произвольным количеством скрытых слоев. Схема одной из таких сетей представлена на Рис. 27. Также как и в предыдущем случае, использовались логистическая функция активации и квадратичная функция ошибки. Для обучения сети использовался устойчивый алгоритм обратного распространения ошибки (`resilient backpropagation`), характеризующийся следующим правилом изменения весов:

$$\beta_k^{(t+1)} = \beta_k^{(t)} - \eta_k^{(t)} \cdot \text{sign} \left(\frac{\partial E^{(t)}}{\partial \beta_k^{(t)}} \right),$$

где t — номер шага обучения.

6.4. Применение нейронных сетей к рассматриваемой задаче

Для классификации объектов городского ландшафта были использованы нейронные сети с одним и двумя скрытыми слоями. Инструментами являлись язык программирования R (традиционно) [27], а также библиотеки `nnet` [50] и `neuralnet` [51].

В связи с тем, что возможное количество связей между нейронами в библиотеке `nnet` ограничено, для набора данных, содержащего изображение со всеми параметрами масштаба, нельзя построить сеть с более чем 6 нейронами в скрытом слое. Это сказывается на качестве классификации. На Рис. 28 можно видеть, что начиная с 10 нейронов в скрытом слое точность классификации колеблется в диапазоне $[0.65, 0.75]$, принимая значение, зависящее от случайно генерируемых начальных весов сети. Уменьшения точности с увеличением числа нейронов не наблюдается, поэтому расчеты проводились при максимально возможном количестве нейронов в скрытом слое.

Лучшие результаты классификации были достигнуты при значениях параметра масштаба равных 60, 40 и 20. Для распознавания объектов на снимках с другими параметрами масштаба данная нейронная сеть подходит плохо.



Рис. 28. Графики зависимости точности классификации от числа нейронов в скрытом слое для параметра масштаба равного 20 (синий) и для всех параметров масштаба (красный).

Таблица 6. Результаты классификации наборов данных с разными параметрами масштаба с помощью нейронной сети с одним скрытым слоем

Параметр масштаба (уровень сегментации)	Достигнутая точность классификации на тестовой выборке	Количество нейронов в скрытом слое
20	70.41	31
40	70.41	31
60	71.01	31
80	66.86	31
100	67.65	31
120	62.33	31
140	56.21	31
All scales	64,89	6

На Рис. 29 можно видеть, что точность классификации, производимой двухслойной нейронной сетью, уменьшается с увеличением числа нейронов в скрытых слоях. В связи с этим, расчеты проводились при 10 нейронах в каждом из скрытых слоев, хотя пакет `neuralnet` не накладывает ограничений на их количество. Лучшие результаты классификации были достигнуты при использовании всех данных и использовании данных с параметром масштаба, равным 20. Но даже для этих данных значения точности классификации недостаточно велики.

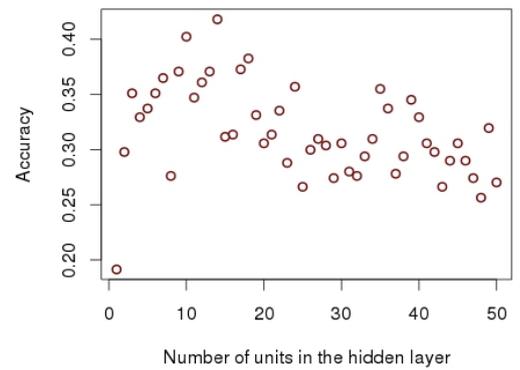
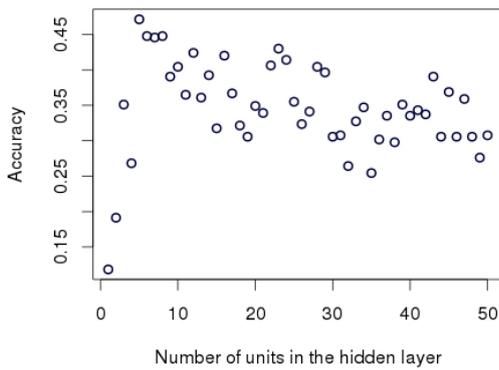


Рис. 29. Графики зависимости точности классификации от числа нейронов в каждом из двух скрытых слоев для параметра масштаба равного 20 (синий) и для всех параметров масштаба (красный).

Таблица 7. Результаты классификации наборов данных с разными параметрами масштаба с помощью нейронной сети с двумя скрытыми слоями

Параметр масштаба (уровень сегментации)	Достигнутая точность классификации на тестовой выборке	Количество нейронов в каждом скрытом слое
20	42.49	10
40	38.56	10
60	37.85	10
80	34.1	10
100	30.99	10
120	27.06	10
140	25.13	10
All scales	40.53	10

7. Выводы

В данной работе рассматривалась задача классификации объектов городского ландшафта по аэрофотоснимкам высокого разрешения. Рассмотрены различные подходы выделения значимых параметров из полученных изображений и их дальнейшее использование для классификации. Особенностью выборки данных является большое число параметров каждого объекта и малый объем самой выборки (15-20 экземпляров на каждый класс).

7.1. Сравнение методов классификации

Были рассмотрены несколько подходов для классификации объектов в данной работе, а именно: деревья решений (CART [25], пакет «rpart» [28]) и построенные на их основе случайные леса Бреймана (пакет «randomForest» [43]). Наибольшая точность классификации была достигнута при использовании сразу всех параметров масштаба (уровней сегментации) для построения случайного леса ($\approx 81\%$). Кроме того, для решения данной задачи классификации был использован алгоритм построения системы опорных векторов, в котором наилучшая точность также была выявлена при использовании сразу всех уровней сегментации, но сам ре-

зультат хуже, чем при применении случайного леса. Это может объясняться тем, что задача мультиклассовая, поэтому для эффективного применения метода опорных векторов необходимы дополнительные допущения (бинарный принцип «один против многих»), которые могут влиять на точность классификации. Для сравнения, в работе [19] для классификации объектов в этой задаче как раз использовались алгоритмы на основе системы опорных векторов (SVM [52], [53]) и максимальная достигнутая точность $\approx 83\%$. Однако, в этой работе объединялись в один класс потенциально похожие объекты на снимках: бетон, асфальт и автомобиль. В данной работе объединение классов не рассматривалось.

Кроме этого, была рассмотрена возможность применения нейронных сетей прямого распространения с одним или двумя скрытыми слоями для решения данной задачи классификации. Эксперименты показали, что использование рассмотренных моделей нейронных сетей в данном случае нецелесообразно.

7.2. Влияние уровня сегментации на точность классификации

При малых значениях параметра масштаба (уровня сегментации) достаточно хорошо классифицируются объекты наибольшего числа классов, в особенности tree (деревья). При больших значениях параметра масштаба приемлемо классифицируются только бетонные конструкции, что может быть связано с их относительно равномерной поверхностью и большой площадью.

Список литературы

- [1] Walton, J., Nowak, D., Greenfield, E., 2008. Assessing urban forest canopy cover 704 using airborne or satellite imagery. *Arboriculture and Urban Forestry* 34 (6), 705–334–340.
- [2] Lang, S., Schopfer, E., Holbling, D., Blaschke, T., Moeller, M., Jekel, T., Kloyber, E., 2008. Quantifying and qualifying urban green by integrating remote sensing GIS 673 and social science methods. In: Petrosillo et al. (Eds.), *Use of Landscape Sciences 674 for the Assessment of Environmental Security*. Springer, Netherlands, pp. 93–675–105.
- [3] Zhou, Y., Wang, Y., 2008. Extraction of impervious surface areas from high spatial resolution imagery by multiple agent segmentation and classification. *Photogrammetric Engineering and Remote Sensing* 74 (7), 857–868.
- [4] Jin, X., Davis, C., 2005. Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP Journal on Applied Signal Processing* 14, 2196–2206.
- [5] Herold, M., Liu, X., Clarke, K., 2003. Spatial metrics and image texture for mapping urban land use. *Photogrammetric Engineering and Remote Sensing* 69 (9), 991–1001.
- [6] Thomas, N., Hendrix, C., Congalton, R., 2003. A comparison of urban mapping methods using high-resolution digital imagery. *Photogrammetric Engineering and Remote Sensing* 69 (9), 963–972.
- [7] Blaschke, T., Burnett, C., Pekkarinen, A., 2004. New contextual approaches using image segmentation for object-based classification. In: De Meer, F., de Jong, S (Eds.), *Remote Sensing Image Analysis: Including the Spatial Domain*. Kluwer Academic Publishers, Dordrecht, pp. 211–236.
- [8] Myint, S., Gober, P., Brazel, A., Grossman-Clarke, S., Weng, Q., 2011. Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment* 115 (5), 1145–1161.
- [9] Yu, Q., Gong, P., Clinton, N., Biging, G., Kelly, M., Schirokauer, D., 2006. Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. *Photogrammetric Engineering and Remote Sensing* 72 (7), 799–811.
- [10] Kim, M., Madden, M., Warner, T., 2008. Estimation of optimal image object size for the segmentation of forest stands with multispectral IKONOS imagery. In: Blaschke, T., Lang, S., Hay, G. (Eds.), *Object-based Image Analysis; Spatial Concepts for Knowledge-driven Remote Sensing Applications*. Springer, Heidelberg, Berlin, New York, pp. 291–307.
- [11] Trias-sanz, R., Stamon, G., Louchet, J., 2008. Using colour, texture, and hierarchical segmentation for high-resolution remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing* 63 (2), 156–168.
- [12] Dorren, L., Maier, B., Seijmonsbergen, A., 2003. Improved Landsat-based forest mapping in steep mountainous terrain using object-based classification. *Forest Ecology and Management* 183 (1–3), 31–46
- [13] Kim, M., Madden, M., Xu, B., 2010. GEOBIA vegetation mapping in Great Smoky Mountains National park with spectral and non-spectral ancillary information. *Photogrammetric Engineering and Remote Sensing* 76 (2), 137–149.

- [14] Liu, D., Xia, F., 2010. Assessing object-based classification: advantages and limitations. *Remote Sensing Letters* 1 (4), 187–194.
- [15] Novack, T., Esch, T., Kux, H. and Stilla, U., 2011, Machine learning comparison between WorldView-2 and Quickbird-2-simulated imagery regarding object-based urban land cover classification. *Remote Sensing*, 3, pp. 2263–2282.
- [16] Zhou, W., Troy, A., 2009. Development of an object-based framework for classifying and inventorying human-dominated forest ecosystems. *International Journal of Remote Sensing* 30 (23), 6343–6360.
- [17] Kim, M., Warner, T., Madden, M., Atkinson, D., 2011. Multi-scale GEOBIA with very high spatial resolution digital aerial imagery: scale, texture and image objects. *International Journal of Remote Sensing* 32 (10), 2825–2850
- [18] Definiens, 2006. *Definiens Professional 5 Reference Book*. Definiens AG, Munchen, Germany.
- [19] Johnson, B., Xie, Z., 2013. Classifying a high resolution image of an urban area using super-object information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83, 40-49.
- [20] Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I., Heynen, M., 2004. Multiresolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing* 58 (3–4), 239–258.
- [21] [Urban Land Cover Data Set](#)
- [22] Walker, J., Blaschke, T., 2008. Object-based land-cover classification for the Phoenix metropolitan area: optimization vs. transportability. *International Journal of Remote Sensing* 29 (7), 2021–2040.
- [23] Hunt, E.B., Marin, J., Stone, P.J. (1966). *Experiments in induction*. New York: Academic Press.
- [24] [Деревья решений](#)
- [25] L. Breiman, J.H. Friedman, R.A. Olshen, and C.T. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.
- [26] L. Hyafil and R. Rivest. Constructing optimal decision trees is NP-complete. *IPL*, 5:1(1976), 15-17.
- [27] [The R Project for Statistical Computing](#)
- [28] [rpart: Recursive Partitioning and Regression Trees](#)
- [29] [MathWorks - MATLAB and Simulink for Technical Computing](#)
- [30] [ClassificationTree class](#)
- [31] Чистяков С.П. Случайные леса: обзор. Труды Карельского научного центра РАН. – № 1. – 2013. – С. 117–136.
- [32] Kuncheva L. I. *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, New Jersey: John Wiley Sons, 2004. 349 p.

- [33] Breiman L. Out-of-bag estimation. Technical report, Statistics Department University of California, Berkeley. 1996. P. 1–13.
- [34] Dietterich T. G. Ensemble methods in machine learning. First International Workshop on Multiple Classifier Systems. Lecture Notes in Computer Science. New York: Springer, 2000. P. 1–15.
- [35] Breiman L. Bagging predictors. *Machine Learning*. 1996. Vol. 24, N 2. P. 123–140.
- [36] Behlmann P., Yu B. Analyzing bagging. *Annals of Statistics*. 2002. Vol. 30. P. 927–961.
- [37] Parmanto B., Munro P. W., Doyle H. R. Improving committee diagnosis with resampling technique. *Advances in Neural Information Processing Systems*. 1996. Vol. 8. P. 882–888.
- [38] Freund Y., Shapire R. E. Experiments with a new boosting algorithm *Proceedings of the 13rd International Conference on Machine Learning*. P. Morgan Cauffman. 1996. P. 146–148.
- [39] Cherkauer K. G. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. Eds. P. Chan *Working Notes on the AAAI Workshop on Integrating Multiple Learned Models*. 2006. P. 15–21.
- [40] Dietterich T. G., Kong E. B. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms Technical report, Department of computer science Oregon State University. 1995. P. 1–22.
- [41] Dietterich T. G. An Experimental Comparison of Three Methods for Constructing Ansembles of Decision Trees: Bagging, Boosting and Randomization *Machine Learning*. 1999. P. 1–20.
- [42] Siroky D. Navigating Random Forests and related advances in algorithmic modeling. *Statistics Surveys*. 2009. Vol. 3. P. 147–163.
- [43] [randomForest: Breiman and Cutler’s random forests for classification and regression](#)
- [44] [TreeBagger Class](#)
- [45] [К.В. Воронцов. Лекции по методу опорных векторов](#)
- [46] Bartlett P., Shawe-Taylor J. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel Methods*. MIT Press, Cambridge, USA, 1998.
- [47] Shawe-Taylor J., Cristianini N. Robust bounds on generalization from the margin distribution: Tech. Rep. NC2-TR-1998-029: Royal Holloway, University of London, 1998.
- [48] Vapnik V., Chapelle O. Bounds on error expectation for support vector machines. *Neural Computation*. 2000. Vol. 12, no. 9. Pp. 2013–2036.
- [49] [e1071: Misc Functions of the Department of Statistics](#)
- [50] [nnet: Feed-forward Neural Networks and Multinomial Log-Linear Models](#)
- [51] [neuralnet: Training of neural networks](#)
- [52] Burdges, C., 1998, A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, pp. 121–167.
- [53] Mountrakis, G., Im, J. and Ogole, C., 2011, Support vector machines in remote sensing: a review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66, pp. 247–259.