Методы и алгоритмы цифровой почвенной картографии –
модели почвенно-ландшафтных связей для категорий номинальной шкалы

Почвенный институт им. В.В. Докучаева
Географический факультет МГУ имени М.В. Ломоносова
© Д.Н. Козлов, Н.И. Лозбенев
daniilkozlov@gmail.com
v.0.99 от 07 декабря 2016

Содержание

Упражнение демонстрирует порядок построения модели почвенно-ландшафтных связей (ПЛС) для номинальных категорий почвенной классификации с использованием инструментов статистического анализа и машинного обучения в среде R: линейного дискриминантного анализа (linear discriminant analysis), ансамбля деревьев решений (Random Forest) и метода опорных векторов (supported vector machine). Использован набор данных, собранных в 2013-2016 гг. при детальном картировании почвенного покрова целинной лесостепи Среднерусской возвышенности (Центрально-Черноземный заповедник, Козлов, 2015; Лозбенев, Козлов, 2016).

Последовательно решаются задачи:

1. загрузить и визуализировать исходные данные почвенно-топографической съемки;
2. провести предварительный экспертный и количественный анализ положения почв в пространстве факторов (оценить информативность индикационных переменных);
3. построить модели почвенно-ландшафтных связей по обучающей выборке с использованием трех методов;
4. на основе модели построить прогнозную карту;
5. оценить неопределенности разных моделей и сравнить их друг с другом.

В первую очередь задание концентрирует внимание на особенностях использования среды R при моделировании ПЛС. Описание сути самих методов не входит в его содержание. Рекомендуется самостоятельно познакомиться с математическим описанием методов – ссылки даны в начале соответствующих разделов.

Время выполнения упражнения зависит от навыков работы в среде R. Опытным пользователям потребуется не более 2 часов, начинающим – до 8 часов. Упражнение можно адаптировать под свои задачи, заменив данные на материалы персональных исследований: точки почвенных описаний и характеристики условий почвообразования, распределенные в ячейках регулярной сетки (свойства рельефа, почвообразующие породы, спектральные индексы и др.).

Данная версия упражнения тестировалась с версией R 3.3.1 (2016-10-10). Актуальную версию задания можно найти в одноименном разделе учебного курса «Анализ данных в среде R» программы подготовки бакалавров кафедры физической географии и ландшафтоведения географического факультета МГУ имени М.В. Ломоносова по адресу http://www.landscape.edu.ru/edu_help4_R.shtml.

1. ПОДГОТОВКА ДАННЫХ И РАБОЧЕЙ СРЕДЫ R

Задача 1.1: скачать набор данных

Архив с данными доступен для скачивания по любой из двух ссылок www.landscape.edu.ru/files/DSM_METdataset_v0.99.zip или https://yadi.sk/i/xEJB1-9DyeqUC. В его составе файлы: 1) met_soilpoints.txt – текстовая таблица результатов почвенного обследования; 2) факторы-индикаторы условий почвообразования – морфометрические характеристики рельефа в растровом формате SAGA (.sdat); 3) R_map_sid_pred.R – скрипт с исполняемым кодом упражнения – его нужно открыть в RStudio и последовательно выполнять команды, сверяясь с пояснениями в руководстве.

Архив необходимо распаковать в рабочую директорию компьютера и отредактировать путь к ней в первой строке исполняемого кода.

```
# назначить рабочую директорию
setwd('D:/Dropbox/0_EDU/R/METsid')
```

После выполнения команды набор данных станет доступным для выполнения задания. Альтернативный способ – выбрать рабочую директорию из меню программы Session – Set As Working Directory – Choose Directory...

Задача 1.2: установить и активировать необходимые библиотеки R

При выполнении упражнения понадобятся дополнительные библиотеки R. Их можно установить командой вида install.packages('car', dep = TRUE). Этой же командой можно переустановить библиотеку, если ее версия устарела или обращение к ней или ее функциям вызывает ошибки. После установки библиотек или после перезагрузки сессии R их нужно разово активировать командой вида library(raster) или галочкой в списке подключаемых библиотек окна Packages программы RStudio. Настоятельно рекомендуется активировать все необходимые библиотеки в начале работы.

```
# необходимые для выполнения упражнения пакеты
library(raster)
library(car)
library(leaflet)
library(rasterVis)
library(randomForest)
library(C50)
library(ellipse)
library(MASS)
library(relaimpo)
library(e1071)
library(ithir)
```

Задача 1.3: загрузить и визуализировать данные

Исходные материалы детальной почвенно-топографической съемки включают цифровую модель рельефа (ЦМР) с разрешением 2,5 м, построенную кригинг-интерполяцией более 12 тыс. пикетов геодезической съемки дифференцированной системой спутникового позиционирования (точность измерения координат и высоты 1 см). В программе SAGA на основе ЦМР рассчитан набор морфометрических величин, описывающий потенциальное перераспределение влаги микрорельефом: крутизна склонов (slp), глубина

замкнутых понижений (cdep), водосборная площадь (facc), топографический индекс влажности (twi), относительные превышения в окрестности 10 и 60 м (tpi10 и tpi60), плановая (plncurv), профильная (prfcurv), общая (gnrcurv) и кроссекционная (cscurv) кривизны, превышение над водотоком (aacn). Средствами модели SIMulated Water Erosion программы Grass GIS рассчитан слой перераспределенных осадков (simwe). Далее для краткости будем называть морфометрические свойства рельефа – ковариатами, подчеркивая стремление описать с их помощью пространственное варьирование почв. Доступ к ковариатам из среды R организован командой stack() (пакет raster), которая считывает в многослойный растровый объект covStack все файлы рабочей директории формата SAGA (.sdat).

```
# загрузить растры формата SAGA-grid и определить их проекцию
covStack = stack(list.files(path = "...", pattern='*.sdat'))
proj4string(covStack) = CRS('+init=epsg:32637') # (UTM (WGS84), зона 37N)
# присвоить ковариатам компактные имена
names(covStack) = c("z","slp","cdep","facc","twi","tpi10","tpi60","plncurv",
"prfcurv","gnrcurv", "cscurv","aacn", "simwe")
```

При возникновении ошибки проверьте число файлов с расширением *.sdat в рабочей директории. Их должно быть 13 с шаблоном имени "??_METq10m_????.sdat".

Каждый слой covStack можно визуализировать командой plot() и, для наглядности, добавить поверх горизонтали, сгенерировав их из ЦМР функцией contour().

```
par(mar=c(2, 2, 1.5, 0.2))  # поля графика снизу по часовой стрелке, см
plot(covStack[[1]], main = "Абс. высота, м", col=terrain.colors(255))

# добавить горизонтали поверх изображения
contour(covStack[[1]], drawlabels = TRUE, axes = FALSE, frame = TRUE,
add=TRUE, col="black")
```

Последовательно постройте изображения для каждой ковариаты, добавляя к ним горизонтали по желанию. Сверяясь с легендой, постарайтесь рассмотреть особенности микрорельефа участка картографирования: соотношение выпуклых и вогнутых форм, положение замкнутых понижений, ориентацию ложбин, их густоту и т.д. Участок включает три элемента мезорельефа: 1) узкое междуречье шириной 100 м с суффозионными западинами диаметром до 20 м и глубиной до 1 м, 2) приводораздельную поверхность крутизной 0,5-1,5°, осложненную эрозионно-суффозионными ложбинами шириной 5-10 м и глубиной до 0,5 м, 3) террасовидную субгоризонтальную поверхность с суффозионными западинами. Повсеместно распространены сурчины – округлые повышения диаметром 5-7 м и высотой до 0,5 м. Почвообразующими породами являются лессовидные суглинки мощностью более 7 м (Целищева, Дайнеко, 1966).

С учетом элементов микрорельефа выполнено бурение для морфологического описания почв и отбора образцов. Часть точек бурения закладывалась по регулярной сетке с шагом 10 м. Всего 157 описаний. Результаты морфологических описаний организованы в текстовую таблицу met_soilpoints.txt, строки в которой – почвенные разрезы, колонки – мощности горизонтов A и AB, глубина вскипания карбонатов (HCL) и индекс таксона почв (sid). По классификации 1977 в границах участка обнаружено четыре рода почв: черноземы типичные обычные (Чт), черноземы типичные карбонатные зоотурбированные (Чтк), черноземы выщелоченные обычные (Чв) и луговато-

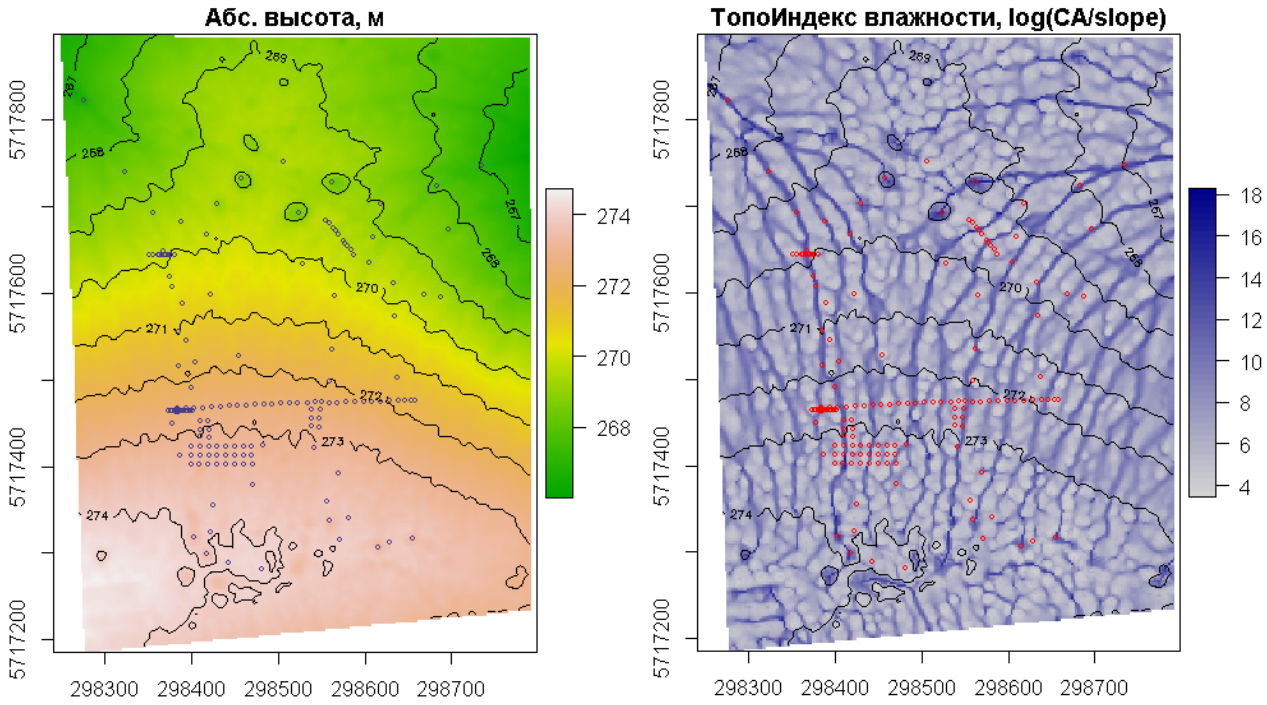
черноземные обычные почвы (Лч). После загрузки текстовой таблицы в объект DSM_data и просмотра ее структуры (функция str()) индексы таксонов преобразуются к номинальной шкале (функция factor) в порядке повышения уровня вторичных карбонатов (levels=c("Лч","Чв","Чт","Чтк")). Функция table() по отношению к номинальной шкале выводит количество элементов каждой категории. В данном случае обучающую выборку образуют 23 точки описания луговато-черноземных почв, 53 точки – черноземов выщелоченных, 64 – черноземов типичных и 17 – черноземы типичные карбонатные.

```
# загрузить таблицу с точками почвенного опробования
DSM_data = read.csv2("met_soilpoints.txt", header = TRUE, sep = "\t",
                     quote = "\"", dec = ".", stringsAsFactors = FALSE)
str(DSM_data) # показать структуру таблицы
DSM_data$sid = factor(DSM_data$sid, ordered=TRUE,
              levels=c("Лч","Чв","Чт","Чтк")) # индекс почв в порядковую шкалу
table(DSM_data$sid) # представленность почв (число точек описаний)
```

Для пространственного сопоставления почвенных описаний со значениями ковариат преобразуем строки таблицы в объекты-точки с координатами Xm, Ym (функция coordinates() библиотеки raster) и определим их географическую проекцию (UTM (WGS84), зона 37N – код EPSG: 32637). Теперь точки можно добавить к изображениям ковариат (функция points()) и подписать при желании каждой из них индекс почв (функция text()).

```
# преобразовать строки таблицы в объекты-точки с координатами Xm, Ym
coordinates(DSM_data) = ~ Xm + Ym # функция пакета "raster"
# и определить проекцию точек (UTM (WGS84), зона 37N)
proj4string(DSM_data) = CRS('+init=epsg:32637')

# добавить точки (элементы обучающей выборки) к активной карте
points(DSM_data, pch = 21:21, cex=0.5, col="darkslateblue")
# и подписать индекс почвы
text(DSM_data$Xm, DSM_data$Ym, labels=DSM_data$sid, cex = 0.7, pos = 3,
offset = 0.5)
```

Пример готовой карты из четырех совмещенных слоев – ковариаты, горизонталей, точек почвенных описаний и индекса почвы.



Используем для визуализации точек и ковариат библиотеку Leaflet map widget. Она позволяет совмещать данные пользователя с пространственно распределенной информацией картографических сервисов (необходимо подключение к Интернету). Для начала покажем точку метеостанции Центрально-Черноземного заповедника относительно космического снимка сверхвысокого разрешения (продукт Esri.WorldImagery).

```
leaflet() %>%
  addMarkers(lng = 36.09170, lat = 51.57200, popup = "Метеостанция") %>%
  addProviderTiles("Esri.WorldImagery")
```

Снимок сделан в начале весны, снег стаял на участках косимой степи, но сохранился в понижениях рельефа – днищах западин и ложбин. За счет этого читается характерный микрорельеф целинной лесостепи Среднерусской возвышенности – пятнисто-западинный на субгоризонтальных междуречьях и струйчато-ложбинный на приводораздельных склонах. Сдвиньте мышкой изображение к востоку, что бы убедиться в устойчивости данной закономерности. На участках степи с пастбищным и некосимым использованием снег сохранился лучше за счет удерживающей роли трав и кустарников во время метелевого переноса.

Добавим поверх снимка слой точек почвенных описаний с атрибутом индекса почв.

```
leaflet() %>%
  addProviderTiles("Esri.WorldImagery") %>%
  addCircleMarkers(data = spTransform(DSM_data, CRS('+init=epsg:4326')),
                   popup = paste0("<strong>sid: </strong>", DSM_data$sid), radius =
                    2,    color =   "red")
```

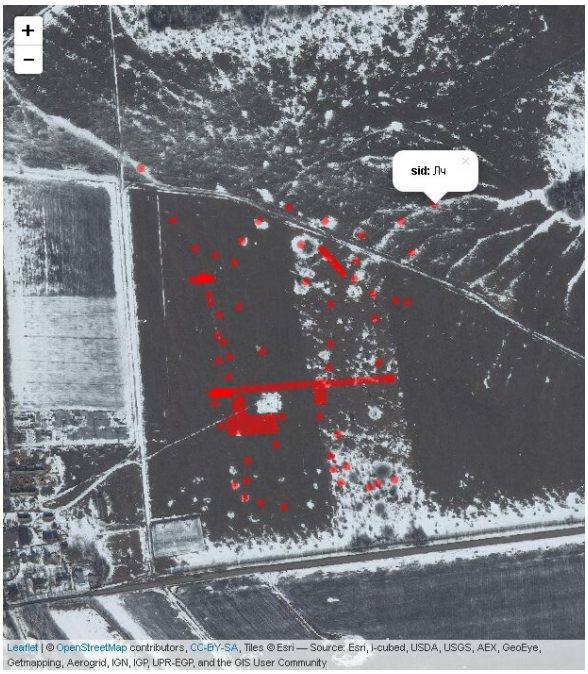
Кликая по точке, можно видеть индекс почвы. Обратите внимание, что в понижениях рельефа с сохранившимся снегом чаще встречаются луговато-черноземные почвы. Ведущим фактором их формирования является дополнительное поверхностное увлажнение, определяющее промывной режим почв.

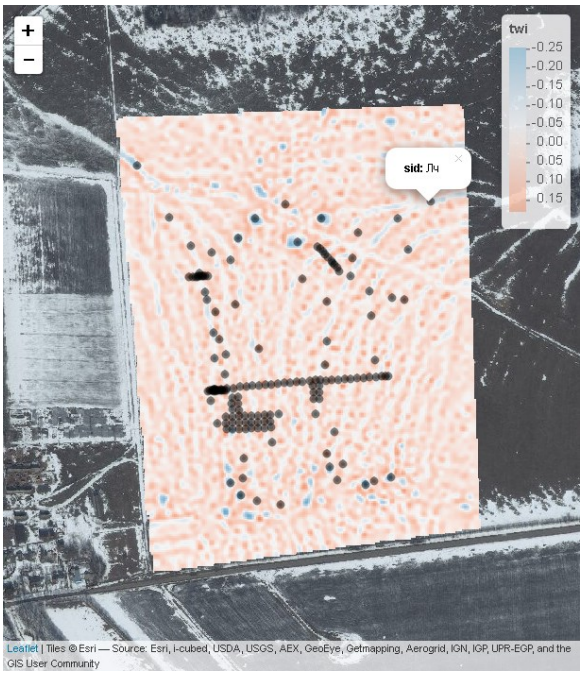
Добавим в виджет растр ковариат, предварительно скопировав его из covStack во временный объект covRST (выбор по номеру слоя covStack) и подобрав информативную палитру.

```
covRST = covStack[[6]] # [6] - относительные превышения в окрестности 10 м
pal = colorNumeric(palette=rev(brewer.pal(3,"RdBu")), domain = values(covRST),
na.color = "transparent") # сформировать палитру, другие – display.brewer.all()

leaflet() %>%
  addProviderTiles("Esri.WorldImagery") %>%
  addRasterImage(covRST, colors = pal, opacity = 0.99) %>%
  addLegend(pal = pal, values = values(covRST), title = "twi") %>%
  addCircleMarkers(data = spTransform(DSM_data, CRS('+init=epsg:4326')), popup
      = paste0("<strong>sid: </strong>", DSM_data$sid), radius = 2, color = "black")
```

Повторите для других свойств рельефа, подбирая для них адаптивную палитру и прозрачность. Варианты палитр можно вызвать командой display.brewer.all().

Сопоставьте роды почв с их положением в микрорельефе и сформулируйте предварительные закономерности – какие почвы в каких элементах микрорельеф чаще всего встречаются, какие действуют механизмы дифференциации почвенного покрова.





Описание всех возможностей пакета Leaflet и список всех доступных картографических сервисов смотрите на https://github.com/leaflet-extras/leaflet-providers и http://leaflet-extras.github.io/leaflet-providers/preview/index.html.

2. АНАЛИЗ ПРИЗНАКОВОГО ПРОСТРАНСТВА

Задача 2.1: извлечь значения ковариат для точек почвенных описаний •

Для начала моделирования почвенно-ландшафтных связей необходимо добавить к атрибутам точек почвенных описаний значения ковариат тех пикселей, внутри которых эти точки расположены (функция extract() пакета «raster»).

```
DSM_data = data.frame(extract(covStack, DSM_data, method="simple", sp=1))
```

Экстракция проводится методом ближайшего соседа по единичному пикселю (method = "simple"). Для вычисления «сглаженной» оценки ковариат можно вычислить среднее значение ближайшего пикселя и его четырех соседей (method = "bilinear"), или задать размер буфера (см. описание команды extract()). Специальный аргумент (sp=1) заставляет добавить значения ковариат к атрибутам объекта-точек DSM_data. Иначе функция extract() возвращает значения ковариат в виде простой таблицы без указания свойств точек. Можно последовательно применять extract() при формировании списка ковариат из разных источников, например, – из стека морфометрических величин на основе ЦМР и стека спектральных характеристик космического снимка. В данном случае, после первого же применения атрибуты точек конвертируются в формат объекта-таблицы командой data.frame().

Для простоты дальнейшей работы оставим в таблице DSM_data только необходимые столбцы: индекс почвы (колонка ‘sid’) и ковариаты. Редактируя список можно сократить или, наоборот, расширить итоговую таблицу исходных данных.

```
DSM_data = data.frame(DSM_data[ ,c("sid","z","slp","cdep","facc","twi","tpi10",
    "tpi60","plncurv","prfcurv","gnrcurv","cscurv","aacn", "simwe")])
format(head(DSM_data), digits=1, scientific=FALSE) # первые 6 строк
```

В итоге должна получить таблица вида:

```
  sid   z slp cdep  facc twi tpi10 tpi60 plncurv  prfcurv gnrcurv  cscurv
1  чв 273 0.5    0  3031  12 -0.06 -0.02    -0.4 -0.00353  -0.017 -0.0024
2  чв 268 0.9    0 15001  13 -0.07 -0.26    -0.3 -0.00397  -0.016 -0.0059
3  чв 273 0.8    0    14   5  0.01  0.05     0.3 -0.00018   0.007  0.0001
4  чв 273 1.1    0   797   9 -0.01 -0.02    -0.1 -0.00003  -0.005 -0.0018
5  чв 272 1.1    0  4429  11 -0.02 -0.04    -0.1  0.00071  -0.004 -0.0027
6  лч 272 0.8    0  9520  13 -0.08 -0.16    -0.6 -0.00163  -0.019 -0.0052
```

Отметим, что, не смотря на то, что технически именно значения ковариат добавляются к атрибутам точек, полученный массив данных нужно рассматривать прямо противоположно – элементами анализа являются не точки описаний, а элементы регулярной сетки, для которых в ходе полевого обследования установлены распространенные в них почвы. Значения ковариат стоит рассматривать как координаты опробованных почв в признаковом пространстве потенциальных факторов почвообразования (в данном случае – рельефа). Предварительный анализ признакового пространства включает проверку независимости переменных и отбор наиболее информативных из них.

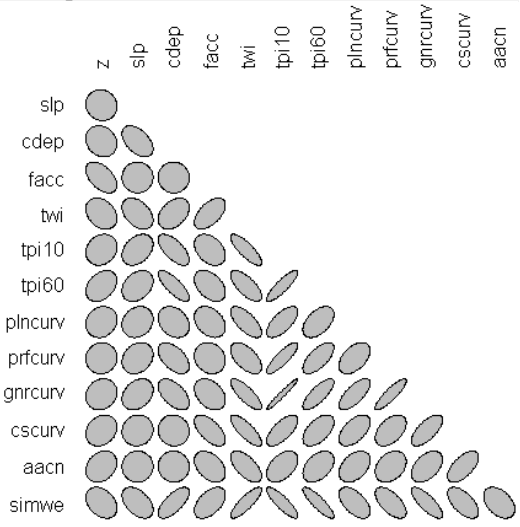
Задача 2.2: проверить независимость переменных •

Зачастую ковариаты коррелируют друг с другом и/или могут содержать экстремальные значения (выбросы), что может исказить модель ПЛС. Проверить независимость переменных можно с помощью матрицы корреляций. Компактный образ матрицы корреляций реализуется функцией symnum() или

построением ее графического образа в виде эллипсов командой plotcorr() из состава пакета ellipse.

```
par(mar=c(0, 0, 0, 0))  # поля графика снизу по часовой стрелке, см
round(cor(DSM_data[ ,-1]), 2)  # рассчитать матрицу корреляций
symnum(cor(DSM_data[ ,-1]))  # компактный образ матрицы корреляций
plotcorr(cor(DSM_data[ ,-1]), type="lower")# образ в виде эллипсов
```

```
         z sl cd f tw t1 t6 pl pr g cs a sm
z        1
slp        1
cdep       .  1
facc     .       1
twi        .  .  . 1
tpi10      .  ,  . +  1
tpi60    . .  ,  . ,  +  1
plncurv          . ,  ,  .  1
prfcurv        .    .  +  ,  .  1
gnrcurv    .  .  . ,  B  ,  ,  +  1
cscurv           , ,  ,  .  .  .  , 1
aacn             . ,  .  .  .  .  . ,  1
simwe    . .  ,  . +  +  +  .  ,  , ,  . 1
attr(,"legend")
[1] 0‘ ’0.3‘.’0.6‘,’0.8‘+’0.9‘*’0.95‘B’1
```



Чем сильнее вытянут эллипс, тем больше доля совместного варьирования признаков. Наиболее независимо от других меняется в пространстве абсолютная высота (Z), крутизна (slp), водосборная площадь (facc) и превышения над местным базисом эрозии (aacn). Наоборот, высокая синхронность в пространственной изменчивости характерна относительным превышениям в окрестности 10 и 60 м (tpi10 и tpi 60), всем четырем мерам кривизны поверхности, а также расчетным значениям перераспределенного слоя осадков (simwe). Глубина замкнутых понижений (cdep) отрицательно связана с относительными превышениями в окрестности 10 и 60 м.

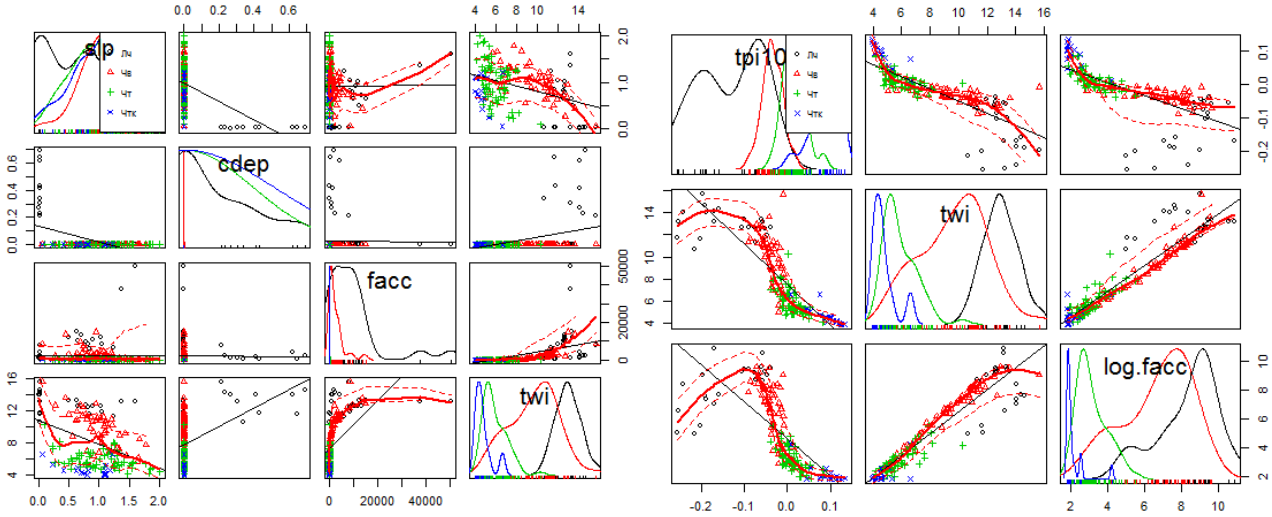
Дополнительные возможности по снижению размерности признакового пространства предоставляет факторный анализ и его упрощенный вариант – метод главных компонент. Однако, рассмотрение их возможностей выходит за рамки данного упражнения. Сейчас ограничимся общим пониманием связанности ковариат друг с другом, что позволит отобрать наиболее значимые переменные для построения модели ПЛС.

Задача 2.3: оценить значимость переменных в разделении почв •

Для построения модели ПЛС был рассчитан заведомо избыточный перечень ковариат с расчетом на то, что на этапе анализа из них будут отобраны две-три, имеющие наиболее тесную связь с изменчивостью почв. Не все средства моделирования предоставляют такую возможность, поэтому полезно сделать это на предварительном этапе. Оценить значимость переменных при разделении почвенных категорий в признаковом пространстве можно визуально и строго статистически. Первый способ наиболее наглядно реализован в функции scatterplotMatrix() библиотеки «car».

```
scatterplotMatrix(~ slp + simwe + log(facc) + twi| sid, data=DSM_data, cex=0.8)
scatterplotMatrix(~ tpi10 + twi + log(facc) | sid, data=DSM_data, cex=0.8)
```

Должны получиться группы графиков, показывающие как таксоны почв разделяются в пространстве пар ковариат и по плотности распределения значений каждой ковариаты. Чем четче пики плотности распределения отделены друг от друга, тем лучше значения данной ковариаты отделяет таксоны почв.



Наиболее четко почв разделяются в пространстве значений топографического индекса влажности (twi), логарифма площади водосбора (log.facc), относительных превышений в окрестности 10 м (tpi10). Ряд почв в порядке снижения уровня залегания вторичных карбонатов (Чтк-Чт-Чв-Лч) характеризуется увеличением значений топографического индекса влажности, водосборной площади и снижением относительных превышений в окрестности 10 м от положительных к отрицательным значениям. Рекомендуется самостотельно построить подобные графики для остальных ковариат. Обратите внимание на возможность произвольно комбинировать ковариаты друг с другом в поиске мультипликативного эффекта, например:

```
scatterplotMatrix(~ twi + log(facc*(tpi10+1)) | sid, data=DSM_data, cex=0.8)
```

Формальная оценка значимости переменных производится при помощи линейной модели lm(), реализованной в пакете relaimpo.

```
# рассчитать разделимость таксонов почв в значениях всех ковариат
lmMod = lm(as.numeric(sid) ~ . , data = DSM_data)
# рассчитать значимость переменных в долях от 1
relImportance = calc.relimp(lmMod, type = "lmg", rela = TRUE)
# вывести относительную значимость переменных в порядке убывания
sort(round(relImportance$lmg, 2), decreasing=TRUE)
```

Результат:

```
twi    tpi10  cscurv gnrcurv   simwe   tpi60 plncurv prfcurv    aacn
0.13    0.13    0.13    0.12    0.10    0.09    0.07    0.07    0.07
```

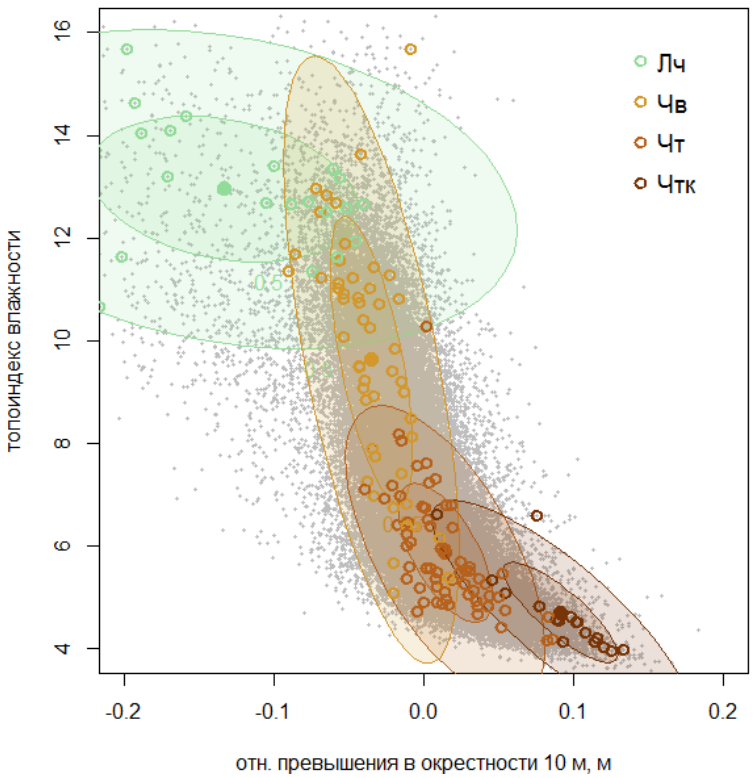
Наиболее значимыми переменными при разделении почв в признаковом пространстве оказались tpi10, twi и две меры формы поверхности (cscurv и gnrcurv). Из равнозначных ковариат cscurv, gnrcurv и tpi10, помня что они скоррелированы, отдадим предпочтение tpi10 – как более простой в интерпретации. Сократим набор ковариат до наиболее значимых и повторим проверку.

```
lmMod = lm(as.numeric(sid) ~ twi+tpi10+cdep+cscurv+aacn, data = DSM_data)
relImportance = calc.relimp(lmMod, type = "lmg", rela = TRUE)
sort(round(relImportance$lmg, 2), decreasing=TRUE)
```

Самыми значимыми ковариатами и при этом простыми в интерпретации оказались tpi10 и twi. Первая выражает превышение каждого пикселя над средней высотой его окрестности радиусом 10 м. Отрицательные значения соответствуют понижениям, положительные – повышениям микрорельефа, область нулевых значений – плоским участкам, либо местам перегиба выпуклых/вогнутых форм. Топографический индекс влажности (twi) равен логарифму отношения водосборной площади и крутизны, то есть выражает соотношение потенциального стока и скорости его транзита через ячейку ЦМР. Для большей наглядности используем при моделировании только эти две ковариаты, помня что они объясняют большую часть изменчивости почв, но не дают полного его описания.

Показать, как точки почвенного опробования расположены в пространстве двух самых значимых ковариат позволяет диаграмма рассеяния.

```
par(mar=c(4, 4, 1, 0.2))  # поля графика снизу по часовой стрелке, см
# сформировать палитру (HEX-цвета почв "Лч", "Чв", "Чт", "Чтк")
sid_col = c("#92DC99", "#D39829", "#B7621C", "#793405")
palette(sid_col)
# все пиксели в пространстве двух ковариат
plot(as.vector(covStack$tpi10), as.vector(covStack$twi), col="grey", pch=20,
     cex=0.6, xlim = c(-0.2,0.2), ylim = c(4,16),  xlab="отн. превышения в
     окрестности 10 м, м", ylab="топоиндекс влажности")
# отметить пиксели, опробованные при почвенном обследовании
points(DSM_data$tpi10, DSM_data$twi, col=DSM_data$sid, lwd=2, cex=1.2)
# добавить эллипс доверительного интервала таксонов почв 50 и 90%
dataEllipse(DSM_data$tpi10, DSM_data$twi, DSM_data$sid, c(0.5,0.95),
     add=TRUE, plot.points=TRUE, lwd=1, fill=TRUE, fill.alpha=0.15,
     pch=c(21,21,21,21), col=sid_col)
# добавить легенду
legend("topright",bg="white",pch=c(21,21,21,21), legend=c("Лч","Чв","Чт",
"Чтк"), inset = .02, bty="n", col= sid_col, cex=1.2, pt.lwd=2, pt.bg="white")
```



Из графика видно, что области разных таксонов почв в пространстве ковариат пересекаются, т.е. при одних и тех же значениях ковариат наблюдаются разные почвы. Цель исследования – найти лучшее разделение почв в пространстве признаков и выразить это разделение формальной моделью почвенно-ландшафтных связей, имеющий адекватный физический смысл.

Вы можете исследовать другие проекции многомерного признакового пространства ПЛС на двухмерную плоскость, меняя ковариаты у аргументов X и Y функций points() и plot(). Во избежание недоразумений удалите аргументы xlim и ylim функции plot().

3. МОДЕЛИРОВАНИЕ ПОЧВЕННО-ЛАНДШАФТНЫХ СВЯЗЕЙ (ПЛС)

Цель моделирования ПЛС – используя выборку пикселей с опробования, выразить зависимость почв от факторов почвообразования в виде математической формы, специфичной для каждого вида анализа (дискриминантной функции, свода правил и др.). Строгая постановка исследовательской задачи и обзор методов ее решения наглядно сформулированы в вводной лекции К.В. Воронцова школы анализа данных Yandex.

Полный цикл цифрового тематического картографирования предполагает сравнения результатов моделирования ПЛС, полученных разными методами, и выбора из них лучшего, дающего наиболее полное описание разнообразия почв. Из большого числа методов в упражнении использованы только три: линейного дискриминантного анализа (Linear Discriminant Analysis), ансамбля деревьев решений (Random Forest) и метода опорных векторов (Supported Vector Machine). Для понимания математической основы каждого метода настоятельно рекомендуется познакомиться с его описанием по ссылкам, указанным в начале каждого раздела.

Задача 3.1: подготовка обучающей выборки •

Обычно, построение модели ПЛС включает обучение и верификацию, для чего весь массив точек разбивается на 2 части, – большая часть (обычно 70 % точек) используется для обучения и оценки точности модели; а меньшая (30%) – только для независимой проверки точности (верификации). Такое разделение позволяет оценить устойчивость модели к составу обучающей выборки и является обязательным этапом числового анализа данных.

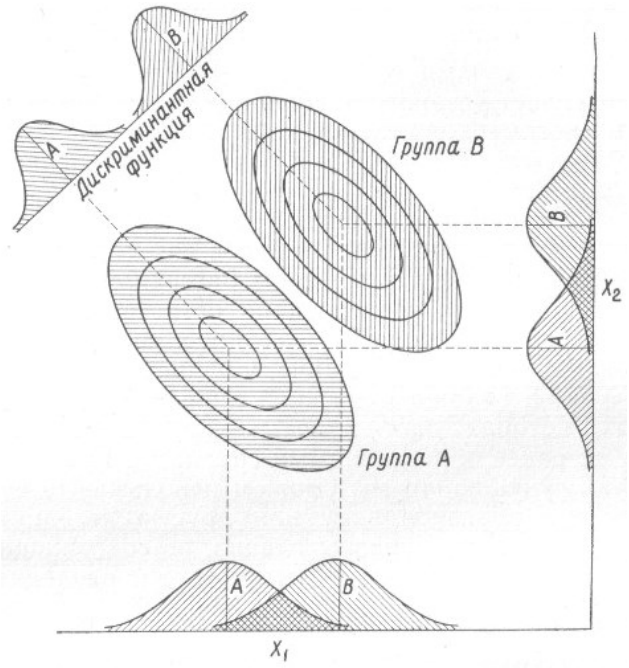
```
set.seed(123) # способ отбора случайной выборки
# выбор 70% элементов выборки
training = sample(nrow(DSM_data), 0.70 * nrow(DSM_data))
training # выводит номера элементов обучающей выборки DSM_data
DSM.train = DSM_data[ training, ] # обучающая выборка (70%)
DSM.verif = DSM_data[-training, ] # выборка для верификации (30%)
points(DSM.train$tpi10, DSM.train$twi, col='black', lwd=1, cex=1.2)
```

Последняя строка добавляет точки обучающей выборки к графику, построенному в конце предыдущего раздела. Это позволяет оценить конфигурацию точек для обучения и верификации. Меняя значение функции set.seed() конфигурацию можно случайным образом изменять, что может использоваться для исследования влияния выбросов на точность модели. В данном случае выбросы (точки за пределами эллипса доверительного интервала) включены в обучающую выборку.

В следующих разделах на одних и тех же данных, но разными методами строятся модели ПЛС: 1) по полному набору данных из 157 точек и 2) только по их части (70%) с последующей независимой проверкой модели по 30%.

Задача 3.2: дискриминантная модель ПЛС •

Применение дискриминантного анализа в задачах тематического картографировании имеет более чем сороколетнюю историю (Webster, Burrough, 1974; Аэрокосмический мониторинг…, 1991; Пузаченко и др., 2006; Козлов и др., 2008; Конюшкова, 2014). Наглядное описание его сути дано Дж. Дэвисом в книге «Статистика и анализ геологических данных» (1977). Основой



дискриминантного анализа является нахождение преобразования пространства ковариат, которое дает минимум отношения разности средних значений к дисперсии двух и более групп элементов.

Наиболее простое преобразование связано с нахождением линейной дискриминантной функции с помощью уравнения регрессии (Linear Discriminant Analysis – lda). В нашем случае зависимой переменной будут разности между многомерными средними таксонов почв, а независимыми – значения ковариат.

```
mod.lda = lda(sid ~ tpi10 + twi, data=DSM_data)
mod.lda # структура модели
```

```
Prior probabilities of groups:
      Лч       Чв       Чт      Чтк
0.1464968 0.3375796 0.4076433 0.1082803

Group means:
          tpi10       twi
Лч  -0.13335995 12.952854
Чв  -0.03502889  9.625900
Чт   0.01430986  5.862323
Чтк  0.09114457  4.702371

Coefficients of linear discriminants:
             LD1         LD2
tpi10 15.7351110 -25.1488950
twi   -0.3679679  -0.5613848

Proportion of trace:
   LD1    LD2
0.9449 0.0551
```

априорная вероятность (доля описаний каждого таксона от общего числа описаний)

среднее значение ковариат таксонов почв (координаты центров эллипсов на рисунке в конце предыдущего раздела)

коэффициенты пересчета значений ковариат в значения дискриминантных функций

вклад каждой функций в разделение таксонов почв в пространстве ковариат

Граничные значения дискриминантной функции каждого таксон почв зависят от величины их априорной вероятности (*prior probabilities of groups*), т.е. вероятности принадлежности любого элемента к каждой группе при прочих равных условиях (не зависимо от значений ковариат). По умолчанию априорная вероятность принимается пропорциональной доле ее элементов в обучающей выборке. Такой способ занижает площадь Чтк и Лч почв, поскольку доля точек их описаний меньше фактической доли этих почв в площади участка картографирования (при полевом обследовании повторные описания Лч и Чтк почв не проводились из-за их однозначного положения в микрорельефе). Аргумент prior функции lda() позволяет экспертно задать априорные вероятности для каждой категории почв. Руководствуясь региональными работами, они приняты для Чт – 0.40, Чв – 0.30, Чтк – 0.15 и Лч – 0.15 (Целищева, Дайнеко, 1966; Фишман, 1977).

```
mod.lda = lda(sid ~ tpi10 + twi, data=DSM_data, prior=c(0.15,0.30,0.40,0.15))
mod.lda
```

Дискриминантная модель mod.lda выражает зависимость четырех таксонов почв от двух свойств рельефа. Функция predict(), используя параметры модели

mod.lda, позволяет предсказать свойства почв для элементов растра на основе значений tpi10 и twi. Такой прогноз, выполненный для исходных элементов обучающей выборки predict(mod.lda), позволит оценить точность модели как целиком, так и по каждому таксону.

```
pred.lda = as.data.frame(predict(mod.lda))
format(head(pred.lda), digits=1, scientific=FALSE) # первые 6 строк
```

```
  class posterior.Лч posterior.Чв posterior.Чт posterior.Чтк x.LD1 x.LD2
1   Чв    0.0997245         0.90       0.0052     0.0000076  -2.0  -1.0
2   Чв    0.4324221         0.57       0.0004     0.0000003  -2.8  -1.4
3   Чт    0.0000006         0.02       0.9427     0.0357508   1.3   0.9
4   Чв    0.0016589         0.83       0.1648     0.0019749  -0.5  -0.7
5   Чв    0.0132386         0.97       0.0120     0.0001119  -1.4  -1.6
6   Чв    0.4602738         0.54       0.0006     0.0000003  -2.8  -1.1
```

Для каждого элемента получен: 1) наиболее вероятный таксон почв (class), 2) постериорные вероятности его принадлежности к каждой группе, 3) положение в координатах дискриминантных функций (x.LD1 и x.LD2). Третий из первых пяти элементов, исходно описанных в поле как Чв (см. п.2.1), с вероятностью 0.94 отнесен моделью к Чт. Шестой элемент с вероятностью 0,54 против 0.46 отнесен моделью к Чв вместо диагностированного в поле Лч.

Общую статистику соответствия исходных и модельных значений удобно посмотреть с использованием функции goofcat() из пакета ithir.

```
goofcat(observed=DSM_data$sid, predicted=pred.lda$class) # качество модели
```

Результат:

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  14  0  0   0
Чв   9 40  4   0
Чт   0 13 57   3
Чтк  0  0  3  14
```
– матрица соответствия, где строки – предсказанные значения (predicted), колонки – исходные (observed), например: из 23 пикселей с Лч почвами 14 – предсказаны верно, а 9 – как Чв.

```
$overall_accuracy
[1] 80
```
– общая точность: 80% совпадений исходных и модельных значений

```
$producers_accuracy
 Лч  Чв  Чт Чтк
 61  76  90  83
```
– точность по колонкам, например, 61% совпадений исходных и модельных значений по Лч почвам (14 из 23)

```
$users_accuracy
 Лч  Чв  Чт Чтк
100  76  79  83
```
– точность по строкам, например, 100% модельных значений Лч почв и исходно были Лч (14 из 14), в то время как только 76% модельных Чв исходно были Чв

```
$kappa
[1] 0.6965634
```
– индекс точности Каппа

Убедившись в достоверности дискриминантной модели (соответствие модельных и фактических описаний – 80%), применим параметры mod.lda к предсказанию таксона почв для каждого элемента растра covStack, независимо от того, опробован он во время полевой съемки или нет. Функция predict() выбирает по названию только те ковариаты covStack, что использованы при расчете модели mod.lda. В отличие от predict(mod.lda) результаты предсказаний predict(covStack, mod.lda) можно сохранить лишь раздельно для наиболее вероятного таксона почв (map.lda.c) и постериорных вероятностей (map.lda.p1 – map.lda.p4), изменяя значение аргумента index, где под кодами 2, 3, 4, 5 подразумеваются Лч, Чв, Чт и Чтк.

```
map.lda.c   = predict(covStack, mod.lda) # наиболее вероятный класс
map.lda.p1 = predict(covStack, mod.lda, type = "probs", index=2) # Лч
map.lda.p2 = predict(covStack, mod.lda, type = "probs", index=3) # Чв
map.lda.p3 = predict(covStack, mod.lda, type = "probs", index=4) # Чт
map.lda.p4 = predict(covStack, mod.lda, type = "probs", index=5) # Чтк
# объединим вероятности в общий стек
map.lda.p   = stack(map.lda.p1, map.lda.p2, map.lda.p3, map.lda.p4)
names (map.lda.p) = c('Лч', 'Чв', 'Чт', 'Чтк') # переименуем
```

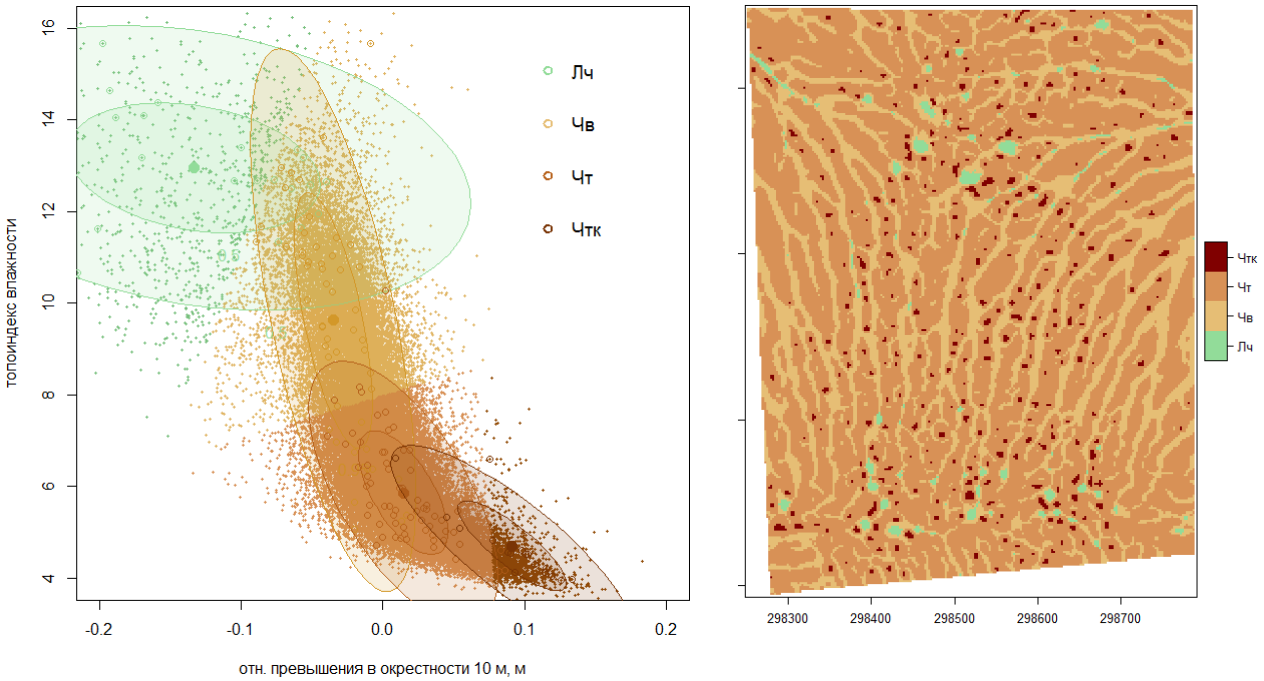
Посмотрим, как линейный дискриминантный анализ разделил пространство ковариат tpi10 и twi на подобласти четырех таксонов почв.

```
par(mar=c(4, 4, 1, 0.2))  # поля графика снизу по часовой стрелке, см
# сформировать палитру (HEX-цвета почв "Лч", "Чв", "Чт", "Чтк")
sid_col = c (("#5AA75F", "#B77E29", "#A25817", "#793405") # цвета эллипсов
palette(c("#79C17E", "#E0B462", "#D89156", "#934D08"))   # цвета областей
# все пиксели в пространстве двух ковариат
plot(as.vector(covStack$tpi10), as.vector(covStack$twi),
     col=as.vector(map.lda.c), pch=20, cex=0.6, xlim = c(-0.2,0.2), ylim = c(4,16),
     xlab="отн. превышения в окрестности 10 м, м", ylab="топоиндекс
     влажности")
# добавить эллипс доверительного интервала таксонов почв 50 и 90%
dataEllipse(DSM_data$tpi10, DSM_data$twi, DSM_data$sid, c(0.5,0.95),
     add=TRUE, plot.points=TRUE, lwd=1, fill=TRUE, fill.alpha=0.15,
     pch=c(21,22,24,25), col=sid_col)
# добавить легенду
legend("topright",bg="white",pch=c(21,22,24,25), legend=c("Лч","Чв","Чт",
"Чтк"), inset = .02, bty="n", col= sid_col, cex=1.2, pt.lwd=2, pt.bg="white")
```

И построим прогнозную карту четырех таксонов почв изобразительными средствами пакет rasterVis (проекция модели ПЛС в территориальном пространстве).

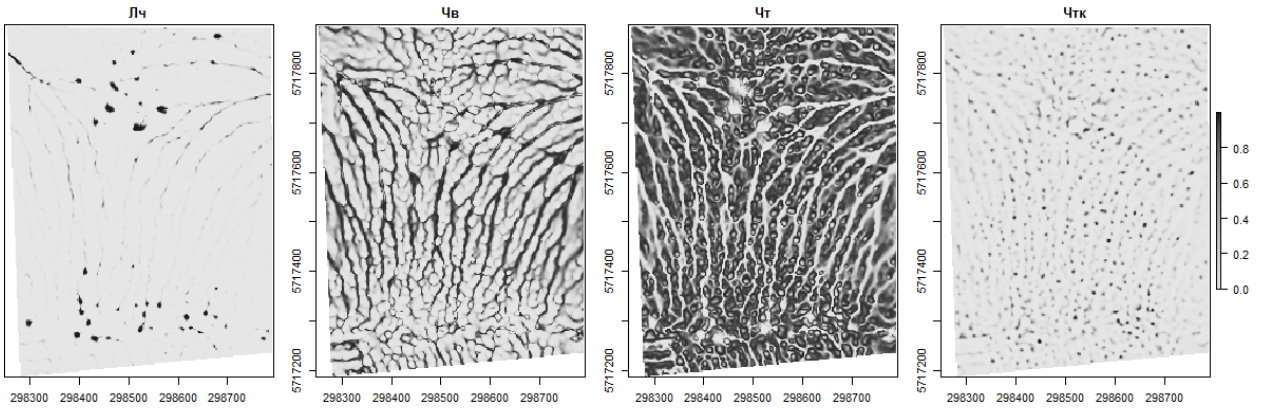
```
cat = levels(as.factor(map.lda.c))[[1]] # числовые коды атрибутов растра
cat[["sid"]] = c("Лч","Чв", "Чт","Чтк")  # их расшифровка (легенда)
levels(map.lda.c) = cat   # сформировать легенду к атрибутам растра
sid_col = c("#92DC99", "#E6BE75", "#D89156", "#934D08") # цвета HEX
levelplot(map.lda.c, col.regions= sid_col) # построить карту
```

Результаты моделирования соответствуют ожиданиям – Чтк имеют максимальные значения tpi при минимальных значениях twi и локализованы в виде пятнистых элементов, обусловленных структурой реликтовых сурчин. Напротив, Лч почвы отличаются минимальными значениями tpi при максимальных значениях twi, соответствующих днищам западин и ложбин. Чт и Чв имеют близкие значения tpi, но различаются по величине водосборной площади. Чв имеют большие значения twi и их ареалы вытянуты вдоль ложбинообразных понижений. Фон почвенного покрова участка составляют черноземы типичные. В пределах субгоризонтального междуречья сочетания почв образует пятнистую структуру, переходящую в ложбинно-древовидную на слабонаклонной приводораздельной поверхности.



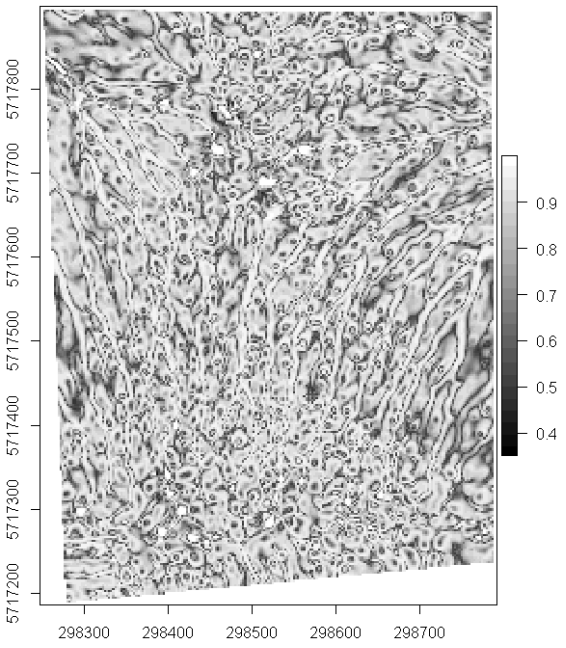
Континуальное отображение почвенного покрова можно получить, раскрасив пиксели по значениям постериорных вероятностей.

```
par(mfrow=c(1,4)) # разместить карты по четыре в один ряд
par(mar=c(2.5, 0, 1, 2))  # поля графика снизу по часовой стрелке, см
for (i in 1:4) plot(map.lda.p[[i]], col = gray.colors(100, start = 0.9, end = 0.1))
```



Интенсивность черного цвета выражает для элементов изображения вероятность принадлежности его почвы (степень подобия) к каждому из четырех таксонов почв. Такое представление подчеркивает континуальный характер почвенного покрова, т.к. смена почв от пикселя к пикселю происходит постепенно. Дискретное и континуальные отображения связаны, поскольку являются отображениями одной и той же модели ПЛС. Так прогнозная карта четырех почв (см. выше) есть результат выбора для каждого пикселя таксона, вероятность принадлежности к которому максимальна. Чем меньше по величине это значение, тем выше неопределенность прогноза почвы. Найдем эту величину для каждого пикселя стека map.lda.pmax с помощью функции stackApply() пакета raster. Аргумент fun задает тип операции (в данном случае – max, найти максимальное значение), а indices – указывает для каких слоев стека эту операцию выполнить. Визуализируем неопределенность прогноза наиболее вероятного таксона в виде карты функцией plot().

```
par(mfrow=c(1,1))
map.lda.pmax = stackApply(map.lda.p, indices=c(1,1,1,1), fun=max)
cellStats(map.lda.pmax, mean) # средняя определенность прогноза
plot(map.lda.pmax, col = grey(seq(0, 1, len = 25)))
```



Белый цвет соответствует участкам с однозначным соответствием почв и значений ковариат дискриминантной модели ПЛС. Напротив, темные участки – области максимальной неопределенности ПЛС, где только на основе двух ковариат трудно однозначно определить доминантную почву. Средняя достоверность прогноза четырех таксонов 82%. Для снижения неопределенности необходимо включить в состав модели ПЛС другие ковариаты и/или провести дополнительное полевое опробование почв в «темных областях». Одно из заданий в конце упражнения предлагает вам самостоятельно рассчитать индекс спутанности (*confusion index*), как еще одну меру неопределенности цифрового моделирования.

В завершении оценим устойчивость дискриминантной модели к составу обучающей выборки. Для этого проведем обучение модели на части выборки (70% DSM.train) и проверим достоверность ее прогноза на независимых элементах (30% DSM.verif).

```
# обучение по части обучающей выборки (70%)
mod.lda.trn = lda(sid ~ tpi10 + twi, data=DSM.train)
# расчет модельных значений
pred.lda.trn = as.data.frame(predict(mod.lda.trn))          # 70%
pred.lda.ver = as.data.frame(predict(mod.lda.trn, DSM.verif)) # 30%
# оценка точности и верификация модели
goofcat(observed = DSM.train$sid, predicted = pred.lda.trn$class) # 70%
goofcat(observed = DSM.verif$sid, predicted = pred.lda.ver$class) # 30%
```

Точность модели (70% выборки)

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  12  0  0   0
Чв   6 32  2   0
Чт   0  7 36   1
Чтк  0  0  2  11

$overall_accuracy
[1] 84

$producers_accuracy
 Лч  Чв  Чт Чтк
 67  83  90  92

$users_accuracy
 Лч  Чв  Чт Чтк
100  80  82  85

$kappa
[1] 0.7604103
```

Верификация (30% выборки)

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч   4  1  0   0
Чв   1  8  2   0
Чт   0  5 21   3
Чтк  0  0  1   2

$overall_accuracy
[1] 73

$producers_accuracy
 Лч  Чв  Чт Чтк
 80  58  88  40

$users_accuracy
 Лч  Чв  Чт Чтк
 80  73  73  67

$kappa
[1] 0.5586987
```

Точность модели для 70% обучающей выборки возросла (84%) по сравнению с полным набором. В то же время точность соответствия модельных и полевых таксонов при верификации (73%) падает не значительно, что говорит об устойчивости модели ПЛС. Самостоятельно проводя эксперименты с вариантами разделения обучающей выборки на части в п.3.1 (произвольно меняя параметр функции set.seed()), можно убедиться, что состав обучающей выборки влияет на точность моделирования ПЛС в пределах 10%.

Задача 3.3: модель ПЛС на основе ансамбля деревьев решений •

Линейный дискриминантный анализ – метод статистического анализа, разработанный в первой половине XX века. В отличие от него ансамбль случайных деревьев (random forest или RF) – эффективный метод машинного обучения, сочетающий достижения двух последних десятилетий: бинарное дерево решений, бутстреп аггрегирование, метод случайных подпространств и декорреляция (Нестеров, 2014; Чистяков, 2013). В отличие от других методов машинного обучения RF имеет небольшое число управляющих параметров (В.Л.Аббакумов Лекция №11 курса "Анализ данных на R в примерах и задачах").

Для наглядности проведем моделирование с двумя наборами параметров: 1) принятыми по умолчанию, 2) с аргументами, дающими лучшее соответствие модельных и фактических значений. Аргумент ntree задает общее число деревьев решений, mtry – число переменных на каждом шаге построения дерева, sampsize – число наблюдений в случайной подвыборке, nodesize – минимальное число наблюдений в узле, replace – подвыборка с возвращением или нет.

```
# модель с настройками по умолчанию
mod.rf  = randomForest(sid~tpi10 + twi, data=DSM_data)
# модель с адаптивными настройками
mod.rf  = randomForest(sid~tpi10 + twi, data=DSM_data, ntree = 100, mtry = 1,
                       sampsize = 50, nodesize = 5, replace = TRUE)
```

Поскольку отбор переменных и подвыборок происходит случайно, даже повторное построение модели с одними и теме же параметрами будет давать несколько отличные результаты. Поэтому естественно, что результаты ваших вычислений будут несколько отличаться от приведенных ниже.

```
print(mod.rf) # показать основные результаты
```

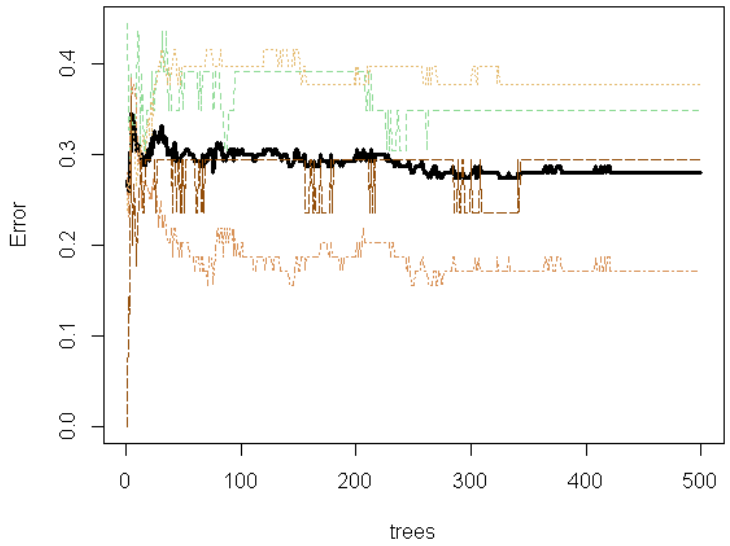
```
Call:
 randomForest(formula = sid ~ tpi10 +
twi, data = DSM_data)


Type of random forest: classification
Number of trees: 500
No. of variables tried at each split:
1

OOB estimate of  error rate: 28.03%

Confusion matrix:
     Лч Чв Чт Чтк class.error
Лч   15  8  0   0   0.3478261
Чв    8 33 12   0   0.3773585
Чт    0 10 53   1   0.1718750
Чтк   0  0  5  12   0.2941176
```

```
Call:
 randomForest(formula = sid ~ tpi10 +
        twi, data = DSM_data, ntree =
        300, mtry = 1, sampsize = 40,
        nodesize = 5, replace = TRUE)
Type of random forest: classification
Number of trees: 300
No. of variables tried at each split:
1

OOB estimate of  error rate: 24.84%

Confusion matrix:
     Лч Чв Чт Чтк class.error
Лч   17  6  0   0   0.2608696
Чв    6 35 12   0   0.3396226
Чт    0  7 54   3   0.1562500
Чтк   0  0  5  12   0.2941176
```

Эксперименты со значениями управляющих параметров не позволили добиться снижения ошибки модели больше чем на несколько процентов (24,85 против 27.39%). Это говорит об устойчивости ансамблей случайных деревьев к начальным настройкам алгоритма. Остальные шаги и результаты моделирования относятся только к варианту с настройками по умолчанию.

Важным управляющим параметром метода следует считать число случайных деревьев. Рекомендуется сначала провести анализ при заведомо избыточном числе деревьев (по умолчанию – 500), а затем на основании графика зависимости ошибки модели от числа деревьев обосновать их оптимальное число.

```
plot(mod.rf, col=c("black", "#92DC99", "#E6BE75", "#D89156", "#934D08"))
```



В данном случае устойчивое снижение ошибки наблюдается при 250 и более деревьев. И хотя при меньшем их числе общая ошибка (черная линия) возрастает не более чем на 2-5%, для отдельных таксонов ошибка может быть больше равновесной на 10 и более процентов. Она выше для Чв (светлокоричневая) и Лч почв (зеленая), напротив, для Чт (коричневая) характерна минимальная ошибка. Высоки риски получения искаженной модели при использовании менее 70 деревьев.

Random forest позволяет оценить вклад каждой переменной с помощью коэффициента Джинни, т.е. ранжировать ковариаты по их значимости в разделении таксонов почв.

```
importance(mod.rf) # относительная важность каждой ковариаты
```

```
      MeanDecreaseGini
tpi10         58.11778
twi           48.49398
```

В данном случае эта информация не столь наглядна, поскольку ради простоты визуализации результатов моделирования мы используем только две самые информативные ковариаты. Они обе значимы, причем относительные превышения в окрестности 10 м (tpi10) несколько важнее топографического индекса влажности (twi). Повторить анализ для полного набора ковариат предлагается в одном из заданий для самостоятельной работы.

Дальнейшие шаги связаны с проверкой точности модели, а также с визуализацией ее результатов и уже знакомы вам по предыдущим разделам. Используем функцию predict() для расчета модельных значений сначала для элементов обучающей выборки, а затем и для всех элементов сетки.

```
pred.rf = as.data.frame(predict(mod.rf)) # наиболее вероятный класс
head(pred.rf)
```

```
  predict(mod.rf)
1              Лч
2              Лч
3              Чт
4              Чв
5              Чв
6              Чв
```

В отличие от дискриминантного анализа RF по умолчанию не дает оценки постериорной вероятности для элементов анализа, ограничиваясь прогнозом таксона почв. Сравнение модельных значений показывает их отличие, как от фактических (п.2.1), так и от предсказанных дискриминантным анализом (п.3.2). Подробную статистику по соответствию модельных и фактических значений дает функция goofcat().

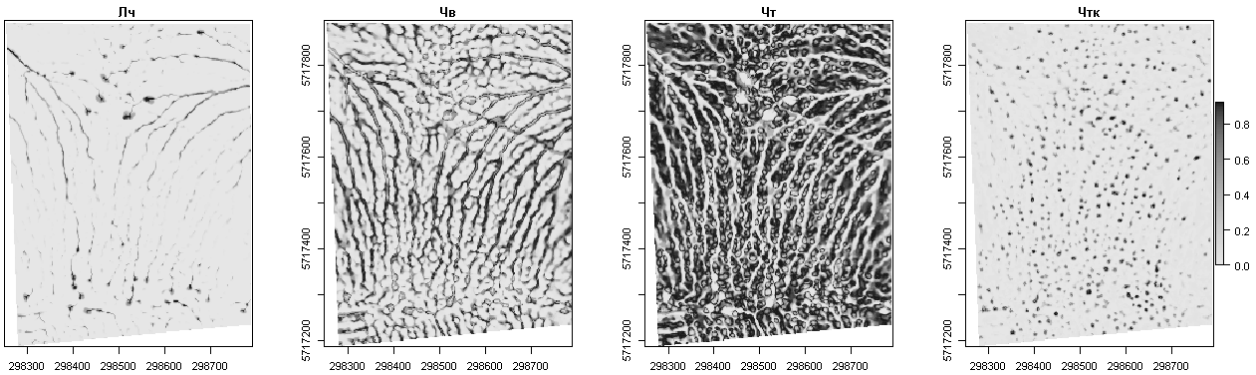
```
goofcat(observed = DSM_data$sid, predicted = pred.rf) # качество модели
```

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  17  6  0   0
Чв   6 36  7   0
Чт   0 11 53   5
Чтк  0  0  4  12

$overall_accuracy
[1] 76

$producers_accuracy
 Лч  Чв  Чт Чтк
 74  68  83  71

$users_accuracy
 Лч  Чв  Чт Чтк
 74  74  77  75

$kappa
[1] 0.6362934
```

В сравнении с дискриминантным анализом, RF демонстрирует менее точное описание положения таксонов почв в пространстве двух ковариат (76% против 81%). Несмотря на это, используем параметры модели для прогноза почв для всех ячеек сетки с шагом 2,5 м.

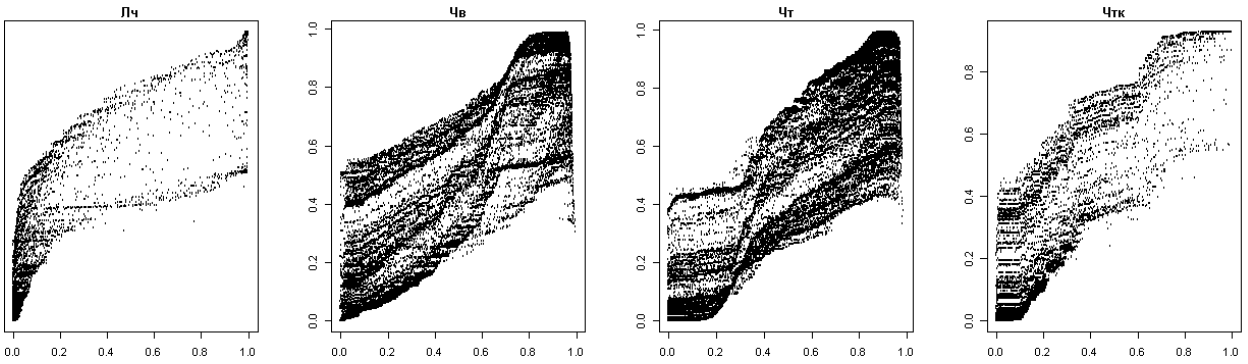
```
map.rf.c  = predict(covStack, mod.rf) # наиболее вероятный класс
map.rf.p1 = predict(covStack, mod.rf, type = "prob", index=1) # Лч
map.rf.p2 = predict(covStack, mod.rf, type = "prob", index=2) # Чв
map.rf.p3 = predict(covStack, mod.rf, type = "prob", index=3) # Чт
map.rf.p4 = predict(covStack, mod.rf, type = "prob", index=4) # Чтк
# объединим вероятности в общий стек
map.rf.p  = stack(map.rf.p1, map.rf.p2, map.rf.p3, map.rf.p4)
names (map.rf.p) = c('Лч', 'Чв', 'Чт', 'Чтк') # переименуем
par(mfrow=c(1,4)) # разместить карты по четыре в один ряд
par(mar=c(2.5, 0, 1, 2))  # поля графика снизу по часовой стрелке, см
for (i in 1:4) plot(map.rf.p[[i]], col = gray.colors(100, start = 0.9, end = 0.1))
```

Обратите внимание на некоторые отличия синтаксиса управляющих команд (type="prob" вместо type="probs" и начало индексации постериорных вероятностей с 1, а не с 2, как у дискриминантного анализа).

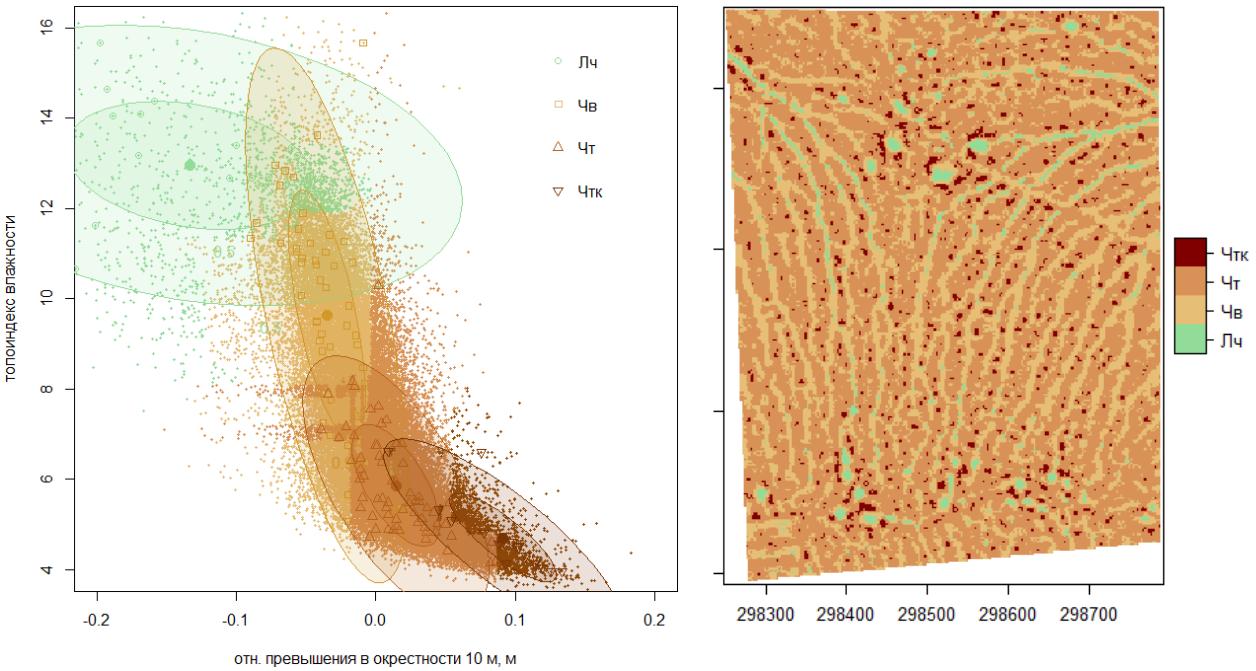


Несмотря на внешнюю схожесть полученных карт, постериорные вероятности lda и RF моделей различаются. Это отчетливо видно на диаграмме рассеяния функции plot(X, Y, …), у которой по оси X – значения постериорных вероятностей дискриминантного анализа (map.lda.p), а по оси Y – ансамбля случайных деревьев (map.rf.p).

```
for (i in 1:4) plot(map.lda.p[[i]], map.rf.p[[i]])
```



Области таксонов в признаковом и территориальном пространстве строим по аналогии с управляющим кодом соответствующего этапа дискриминантного анализа. Замены требует только имя растра с предсказанным моделью таксоном почв (map.rf.c вместо map.lda.c). Перед началом визуализации необходимо отменить деление графической области на четыре части командой dev.off().



По сравнению с дискриминантным анализом границы таксонов в признаковом пространстве RF модели приобрели разрывную форму замысловатых ступеней, причудливо оконтуривая элементы обучающей выборки. Сравнение карт показывает небольшое увеличение площади Чтк и Лч почв и общую «зашумленность» изображения.

Неопределенность прогноза элементарных почвенных ареалов рассчитывается по аналогии с тем, как это делалось в задаче 3.2. Здесь же приведем результаты проверки устойчивость модели ансамбля случайных деревьев к составу обучающей выборки (70% DSM.train против 30% DSM.verif).

```
# обучение по части обучающей выборки (70%)
mod.rf.trn = randomForest(sid~tpi10+twi, data=DSM.train, ntree=300,mtry=2)
# расчет модельных значений
pred.rf.trn = as.data.frame(predict(mod.rf.trn))              # 70%
pred.rf.ver = as.data.frame(predict(mod.rf.trn, DSM.verif)) # 30%
# оценка точности и верификация модели
goofcat(observed = DSM.train$sid, predicted = pred.rf.trn$predict) # 70%
goofcat(observed = DSM.verif$sid, predicted = pred.rf.ver$predict) # 30%
```

Точность модели (70% выборки)

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  14  4  0   0
Чв   4 29  6   0
Чт   0  6 32   2
Чтк  0  0  2  10

$overall_accuracy
[1] 78

$producers_accuracy
 Лч  Чв  Чт Чтк
 78  75  80  84

$users_accuracy
 Лч  Чв  Чт Чтк
 78  75  80  84

$kappa
[1] 0.6845152
```

Верификация (30% выборки)

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч   5  5  0   0
Чв   0  5  3   0
Чт   0  4 21   3
Чтк  0  0  0   2

$overall_accuracy
[1] 69

$producers_accuracy
 Лч  Чв  Чт Чтк
100  36  88  40

$users_accuracy
 Лч  Чв  Чт Чтк
 50  63  75 100

$kappa
[1] 0.5068493
```

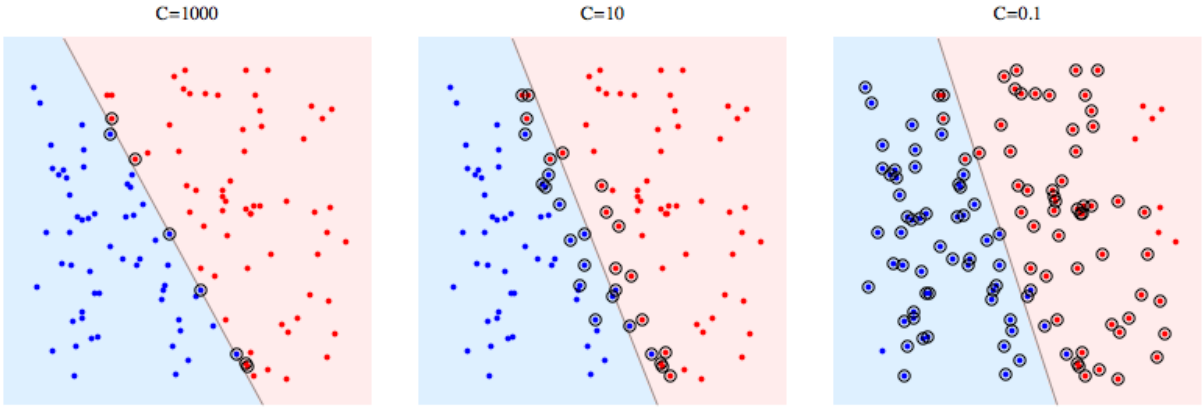
Точность модели, как и в случае с дискриминантным анализом, выше для обучающей выборки (78%) по сравнению с результатами независимой проверки (69%).

Задача 3.4: модель ПЛС методом опорных векторов •

Метод опорных векторов, или машина опорных векторов (Support Vector Machine или SVM) – один из наиболее популярных методов обучения для классификации и регрессионного анализа. Основная идея метода – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве (Н.Анохин, Лекция 7: Машина опорных векторов учебного курса «Алгоритмы интеллектуальной обработки больших объемов данных» НОУ ИНТУИТ, 2016).

В среде R реализация метода опорных векторов доступна в библиотеке e1071. Основным управляющим параметром является аргумент cost. В примере (SVM – hard or soft margins?, 2011) показано, как значения cost влияют на число опорных векторов, используемых при проведении границы классов (обведены кружком). Большие значения аргумента cost используются при описании

линейно разделимых классов, элементы которых не пересекаются в пространстве признаков (hard-margin SVM). Наоборот, при выраженной линейной неразделимости классов используют малые значения cost (soft-margin SVM). При прочих равных условиях, меньшие значения cost будут давать более «мягкие» границы классов, слабо реагирующие на отдельные опорные вектора.




```
# svm-модель с начальными настройками (по умолчанию)
mod.svm = svm(sid ~ tpi10 + twi, data = DSM_data)
summary(mod.svm)
```

```
Parameters:
SVM-Type:  C-classification
SVM-Kernel:  radial
       cost:  100
      gamma:  0.5
Number of Support Vectors:  80
 ( 29 15 7 29 )
Number of Classes:  4
Levels:
 Лч Чв Чт Чтк
```

- параметры классификации (тип, управляющие параметры, характеристики исходных данных)

- число опорных векторов, использованное для обособления области каждого класса

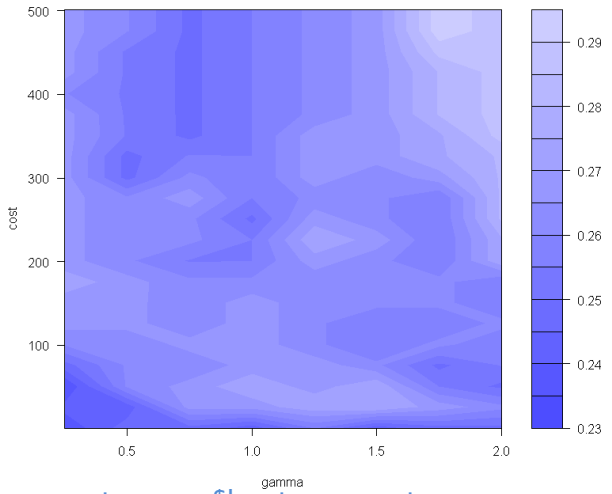
Для описания не прямолинейных границ svm использует так называемые ядра (kernel) – функции, преобразующие систему координат в пространство большей размерности с потенциальной линейной разделимостью классов. Наиболее распространенные ядра – полиномиальное (polynomial), радиальная базисная функция (radial) и сигмовидная (sigmoid). У каждой из них свои параметры, оптимальные значения которых не известны и требуют подбора. Команда tune.svm() пакета e1071 автоматически производит последовательный перебор значений из заданного диапазона управляющих параметров. Она вычисляет ошибку модели для каждого сочетания управляющих параметров и находит лучшее их них – svmtunepar$best.parameters. Эту процедуру целесообразно сначала провести в широком диапазоне значений и грубом шаге его изменении, а затем только в потенциальном диапазоне лучших значений с детальным шагом. Впрочем, небольшие изменения параметров чаще всего слабо влияют на итоговый результат.

Наиболее распространённые ядерные функции:

$$\phi = \begin{Bmatrix} x_i * x_j & \text{линейная} \\ \left(\gamma x_i x_j + coefficient\right)^{degree} & \text{полиномиальная} \\ exp\left(-\gamma\left|x_i - x_j\right|^2\right) & RBF \\ tanh\left(\gamma x_i x_j + coefficient\right) & \text{сигмовидная} \end{Bmatrix}$$

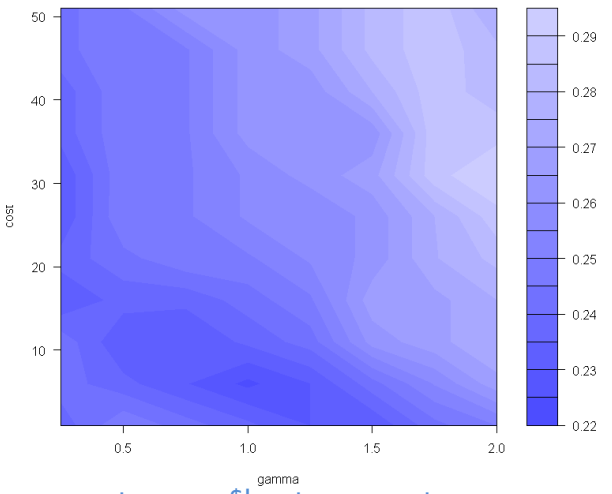
Например, гамма-параметр (γ) радиальной базисной функции (в svm-модели используемой по умолчанию) выражает радиус влияния каждой точки (обратно пропорционально расстоянию). Чем меньше его значение, тем выше роль каждой точки и тем рельефнее поверхность будет изогнута в пространство новой размерности. Напротив, большие значения используют для «сглаженного выгибания».

```
svmtunepar  = tune.svm(sid ~ tpi10 + twi, data =
              DSM_data, gamma =
              seq(0.25,2,0.25),cost=seq(1,501,25))
plot (svmtunepar,  xlab="gamma", ylab="cost")
```




```
svmtunepar$best.parameters
   gamma cost
17  0.25   51
```

```
svmtunepar = tune.svm(sid ~ tpi10 + twi, data =
             DSM_data, gamma =
             seq(0.25,2,0.25), cost = seq(1,51,5))
plot (svmtunepar,  xlab="gamma", ylab="cost")
```




```
svmtunepar$best.parameters
   gamma cost
12     1    6
```

Поскольку при подборе параметров отбор опорных векторов происходит случайно, повторные вызовы функции tune.svm() будут давать несколько отличные результаты. В целом ошибка svm-моделирования (см. легенду к предыдущим графикам) зависит от начальных параметров не более чем на 10%.

При необходимости аргумент class.weights svm-модели позволяет задать вес для каждой категории почв (по аналогии с априорными вероятностями lda-модели). По умолчанию они равны 1. Меняя значения, можно увеличивать/уменьшать площади каждого таксона на прогнозной карте.

```
costs = table(DSM_data$sid)
costs['Лч']   = 1.15
costs['Чв']   = 1.25
costs['Чт']   = 1.35
costs['Чтк'] = 1.15
costs  # веса таксонов
```

Итоговые настройки svm-модели выглядят так:

```
mod.svm = svm(sid ~ tpi10 + twi, data = DSM_data, class.weights = costs, cost =
              6, gamma = 1)
```

Они не исчерпывают всех возможностей svm-анализа, с остальными предлагается познакомиться самостоятельно. Дальнейшие шаги повторяют этапы, подробно разобранные на примере дискриминантного анализа (п.3.2).

Общая статистика соответствия исходных и модельных значений практически идентична результатам дискриминантного анализа. Очень близки и

результаты проверки устойчивости SVM модели к составу обучающей выборки (70% DSM.train против 30% DSM.verif).

```
goofcat(observed=DSM_data$sid,predicted=predict(mod.svm,newdata=DSM_data))
```

. Для наглядности результаты всех трех проверок точности и устойчивости модели (100%=70%+30% объема точек обучающей выборки) расположены рядом.

Точность модели (100% выборки)

```
$confusion_matrix
     Лч Чв Чт Чтк
Лч   14  0  0    0
Чв    9 38  1    0
Чт    0 15 63    5
Чтк   0  0  0   12

$overall_accuracy
[1] 81

$producers_accuracy
 Лч  Чв  Чт Чтк
 61  72  99  71

$users_accuracy
 Лч  Чв  Чт Чтк
100  80  76 100

$kappa
[1] 0.7104568
```

Точность модели (70% выборки)

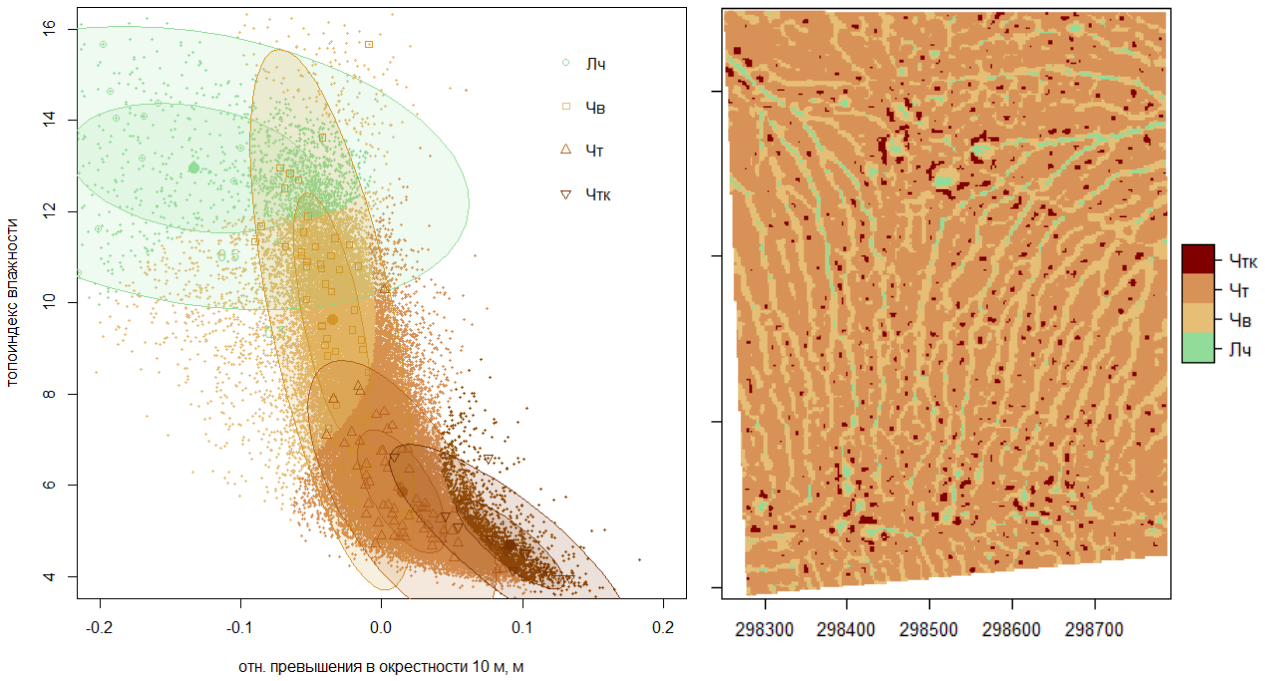
```
$confusion_matrix
     Лч Чв Чт Чтк
Лч   17  3  0    0
Чв    1 30  2    0
Чт    0  6 37    1
Чтк   0  0  1   11

$overall_accuracy
[1] 88

$producers_accuracy
 Лч  Чв  Чт Чтк
 95  77  93  92

$users_accuracy
 Лч  Чв  Чт Чтк
 85  91  85  92

$kappa
[1] 0.8168067
```

Верификация (30% выборки)

```
$confusion_matrix
     Лч Чв Чт Чтк
Лч    4  3  0    0
Чв    1  7  2    0
Чт    0  4 21    2
Чтк   0  0  1    3

$overall_accuracy
[1] 73

$producers_accuracy
 Лч  Чв  Чт Чтк
 80  50  88  60

$users_accuracy
 Лч  Чв  Чт Чтк
 58  70  78  75

$kappa
[1] 0.5728953
```

Прогноз наиболее вероятного таксона почв для каждого элемента сетки covStack в соответствии с моделью mod.svm.

```
map.svm.c = predict(covStack, mod.svm) # наиболее вероятный класс
```

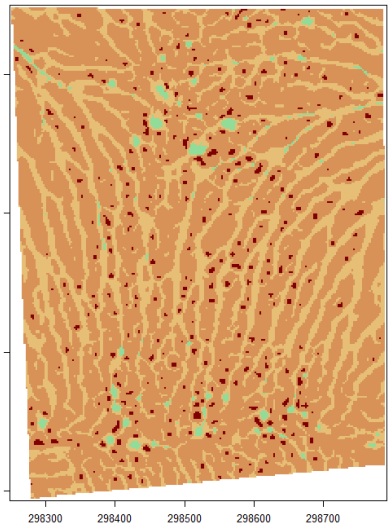


При общей схожести результатов, SVM модель в пространстве двух ковариат формирует криволинейные границы областей соседних таксонов, отдаленно повторяющие контуры эллипсов доверительных интервалов.

Задача 3.5: сравнить результаты разных моделей ПЛС •

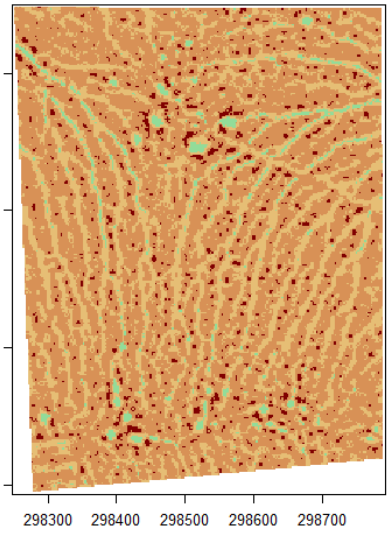
Помимо оценки достоверности модели ПЛС по формальным критериям (соответствие фактическим описаниям, устойчивость при верификации) ее качество определяется с учетом экспертного мнения специалиста картографа. Сопоставление трех картографических моделей демонстрирует их качественную идентичность. Различия связаны с цельностью ареала Чв по элементам древовидно-ложбинной сети (лучше у LDA модели), площадью и густотой Чтк, по реликтовым сурчинам и краям суффозионных западин (реалистичной у RF, SVM моделей и заниженной у LDA) и областью Лч почв по днищам ложбин (выше по водосбору поднимаются у RF и SVM моделей). Карта на основе RF модели отличается некоторой «зашумленностью».

Приведенное сопоставление формальных и экспертных критериев позволяют отказаться от карты на основе RF-модели. Выбирая из LDA и SVM моделей, предпочтение стоит отдать LDA, поскольку при одинаковых формальных значениях точности и подобия воспроизведенной структуры почвенного покрова, LDA-модель обладает лучшей интерпретируемостью почвенно-ландшафтных связей. Экспертное замечание по заниженной площади Лч почв в днищах ложбин можно попробовать устранить, увеличив значение априорной вероятности в начальных настройках при расчете LDA-модели.



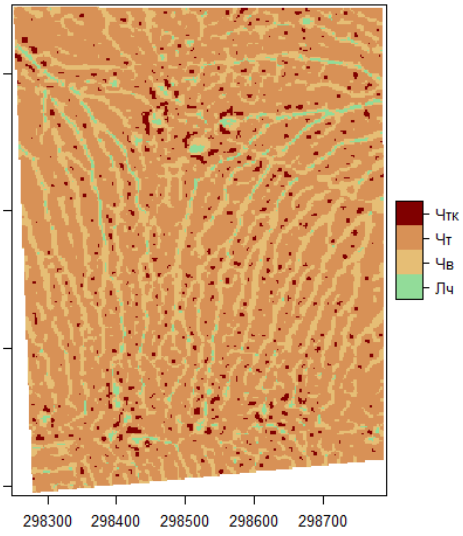
LDA – линейный дискриминантный анализ

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  14  0  0   0
Чв   9 40  4   0
Чт   0 13 57   3
Чтк  0  0  3  14

$overall_accuracy
[1] 80
```



RF – ансамбль случайных деревьев

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  17  6  0   0
Чв   6 36  7   0
Чт   0 11 53   5
Чтк  0  0  4  12

$overall_accuracy
[1] 76
```



SVM – метод поддерживающих векторов

```
$confusion_matrix
    Лч Чв Чт Чтк
Лч  14  0  0   0
Чв   9 38  1   0
Чт   0 15 63   5
Чтк  0  0  0  12

$overall_accuracy
[1] 81
```

4. ЭКСПОРТ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

При необходимости сохранить прогнозные карты в одном из распространенных растровых форматах можно использовать возможности библиотеки raster одним из двух вариантов. Первый, – одновременно с прогнозом наиболее вероятного таксона с помощью аргументов filename и format функции predict(), например:

```
map.lda.c = predict(covStack, map.lda, type = "class", filename = "map_lda_c.tiff",
                    format = "GTiff", overwrite = TRUE, datatype = "FLT4S")
```

Второй, – сохранить уже готовую прогнозную карту в файл функцией writeRaster(), например:

```
writeRaster(map.lda.c, filename = "map_lda_c.tiff", format = "GTiff", overwrite =
            TRUE, datatype = "FLT4S")
```

Поскольку по умолчанию файлы сохраняются в рабочую директорию, во избежание недоразумений, растры не стоит сохранять в формате SAGA (.sdat), а использовать форматах GeoTiff, .grd и др. (см. помощь к функции writeRaster()). В противном случае результаты моделирования будут считаны при следующем обращении к исходным данным в начале упражнения, что приведет к ошибке несовпадения числа слоев в стеке при их переименовании.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Самостоятельно постройте матрицы соответствия между прогнозными значениями разных моделей по образцу:

```
goofcat(as.vector(map.lda.c), as.vector(map.rf.c))
```

   В какой степени совпадают площади таксонов почв разных моделей? Площадь какого таксона наиболее/наименее устойчива к методу построения модели ПЛС?

2. Как изменится точность lda-модели ПЛС, если задать априорные вероятности равными Чт – 0,35, Чв – 0,15, Чтк и Лч – 0,25? Увеличится или уменьшится площадь Чтк и Лч почв на карте элементарных почвенных ареалов?

3. Постройте svm-модель ПЛС на основе линейной ядровой функции и сравните результаты с lda-моделью. Оптимальные значения управляющих параметров обоснуйте самостоятельно.

```
mod.svm = svm(sid ~ tpi10 + twi, data = DSM_data, kernel="linear")
```

4. Рассчитайте индекс спутанности (CI, от англ. *confusion index*) как меру неопределенности прогноза наиболее вероятного таксона для каждого элемента сетки (ЭТЕ – элементарной территориальной единице):

$$CI = 1 - [m_{max} - m_{(max-1)}],$$

   где m_{max} и $m_{(max-1)}$ – два первых по величине значения постериорной вероятности к таксонам почв для каждой ЭТЕ. Значения CI, близкие к 0, соответствуют ЭТЕ, почвы которых однозначно относятся к какому-либо таксону. Чем ближе значение CI к единице, тем неопределеннее модель для ЭТЕ. Сравните значения CI со значениями максимальной постериорной вероятностью, как меры неопределенности.

5. Исследуйте влияние на результаты моделирования ПЛС расширения набора ковариат, руководствуясь своим экспертным мнением и формальными параметрами (точность, значимость и др.). Например, для RF-модели можно оценить относительную значимость всех ковариат, а затем использовать первые три из них для расчета трех моделей ПЛС тремя методами.

```
mod.rf  = randomForest(sid ~ ., data=DSM_data, ntree = 300, mtry = 2)
importance(mod.rf) # относительная важность каждой ковариаты
```

   Цель – получить карту с максимальными показателями соответствия фактическим описаниям при минимальном перечне ковариат (две-три-четыре), обеспечивающими физическую интерпретируемость модели ПЛС. Используйте диаграмму рассеяния, чтобы визуализировать положение почв в пространстве отобранных ковариат. Предложите механизмы связи таксонов почв и отобранных свойств рельефа.

6. Попробуйте предложить новый метод разделения признакового пространства на подобласти на основе обучающей выборки.

СПИСОК ЛИТЕРАТУРЫ

Аэрокосмический мониторинг лесов / Исаев А.С., Сухих В.И., Калашников Е.Н. и др. – М.: Наука, 1991. – 240 с.

Девис Дж. Статистика и анализ геологических данных. – М.: Мир, 1977. – 573 с.

Козлов Д.Н., Пузаченко М.Ю., Федяева М.В., Пузаченко Ю.Г. Отображение пространственного варьирования свойств ландшафтного покрова на основе дистанционной информации и цифровой модели рельефа // Известия РАН. Сер. геогр., 2008, № 4. – С. 112-124.

Козлов Д.Н. На пути к картированию и пространственному моделированию запасов углерода и потоков парниковых газов объектов RusFluxnet и сопряженных экосистем Европейской России // Функционально-экологический мониторинг почвенного углерода и парниковых газов в условиях Центрального региона России / Б. Бадарч, Р. Валентини, В. И. Васенев и др. — Издательство САМполиграфист Москва, 2015. – С. 320.

Конюшкова М.В. Цифровое картографирование почв солонцовых комплексов Северного Прикаспия. – КМК Москва, 2014. – С. 318.

Лозбенев Н.И., Козлов Д.Н. Водно-миграционные особенности структурно-функциональной организации целинных лесостепных ландшафтов Среднерусской возвышенности // Почвоведение – продовольственной и экологической безопасности страны: тезисы докладов VII съезда Общества почвоведов им. В.В. Докучаева и Всероссийской с международным участием научной конференции (Белгород, 15-22 августа 2016г), часть 1 / Отв. Ред. С.А. Шоба, И.Ю.Савин. – Москва-Белгород: Изд. дом «Белгород», 2016. – с. 187-188.

Пузаченко М.Ю., Пузаченко Ю.Г., Козлов Д.Н., Федяева М.В. Картографирование мощности органогенного и гумусового горизонтов лесных почв и болот южнотаежного ландшафта (юго-запад Валдайской возвышенности) на основе трехмерной модели рельефа и дистанционной информации (Landsat 7) // Исследование Земли из космоса, 2006, №4. – С.1-9.

Целищева Л.К., Дайнеко Е.К. Очерк почв Стрелецкого участка Центрально-Черноземного заповедника // Тр. ЦЧГЗ им. В.В.Алехина, вып. X. М.: Лесная промышленность,1967. С. 154-186.

Чистяков С.П. Случайные леса: обзор // Труды Карельского научного центра РАН № 1. 2013. С. 117–136 http://cyberleninka.ru/article/n/sluchaynye-lesa-obzor.pdf

Фишман М.И. Черноземные комплексы и их связь с рельефом на Среднерусской возвышенности // Почвоведение, 1977, №5. с. 17-30.

Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p.

Webster R., Burrough P.A. Multiple discriminant analysis in soil survey // Journal of Soil Science, 1974, vol. 25. – P. 120-134.

Материалы сети Интернет

Аббакумов В.Л. Лекция №11 «Случайные леса. Gradient boosting machine» курса "Анализ данных на R в примерах и задачах" https://compscicenter.ru/courses/data-mining-r-problems/2016-spring/classes/2602/

Анохин Н., Лекция 7: Машина опорных векторов учебного курса «Алгоритмы интеллектуальной обработки больших объемов данных» НОУ ИНТУИТ, 2016. https://www.youtube.com/watch?v=HFj128PAZek

Воронцов К.В. Вводной лекции школы анализа данных Yandex https://www.youtube.com/watch?v=qLBkB4sMztk