

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений

Матрицы расстояний, кластеризация и деревья решений

Г. Мороз

Матрицы расстояний

Матрица расстояний — это матрица $n \times n$, которая содержит значения меры расстояния/сходства между объектами в метрическом пространстве. Существует уйма мер расстояния/сходства, выбор из которых зависит от типа данных. К сожалению, не существует универсального алгоритма выбора метода, так что это остается на откуп исследователям. Кроме того, схожие методы, зародившиеся в биологии, называют string metric: они определяют расстояния между строками (расстояние Хэмминга, расстояние Левинштейна и т. п.)

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений

Бинарные данные: коэффициент Жаккара

Компаративисты сравнивают языки на основе количества общих когнатов в списке Сводеша. Таким образом, для стословника составляются бинарные матрицы, которые отражают, какой когнат в каком идиоме встретился.

```
df <- data.frame(  Lithuanian =      c(1, 1, 1, 1, 0),
                    Latvian  =      c(1, 1, 1, 0, 0),
                    Prussian =      c(1, 1, 0, 0, 0),
                    ChurchSlavonic = c(0, 0, 0, 0, 1))
```

Для каждой пары идиомов строим таблицу сопряженности:

		идиом i	
		1	0
идиом j	1	a	b
	0	c	d

Коэффициент Жаккара рассчитывается по формуле:

$$s(i,j) = \frac{a}{a + b + c} \qquad d(i,j) = \frac{b + c}{a + b + c}$$

В работе [Gower and Legendre 1986] есть и другие методы (14 шт.).
Большинство из них есть в функции `dist.binary()` пакета `ade4`.

презентация доступна: <http://goo.gl/dqocQt>

Бинарные данные: коэффициент Жаккара

метрики
расстояний
метрики расстояний
heatmap
методы
кластеризации
k-means
hierarchical
проблемы
дендрограммы
валидация кластеров
деревья
решений

```
df <- data.frame(  Lithuanian =      c(1, 1, 1, 1, 0),  
                   Latvian   =      c(1, 1, 1, 0, 0),  
                   Prussian  =      c(1, 1, 0, 0, 0),  
                   ChurchSlavonic = c(0, 0, 0, 0, 1))
```

```
df <- t(df) # кластеризации любят держать признаки в строках  
dm <- dist(df, method = "binary")
```

```
dm # матрица расстояний
```

	Lithuanian	Latvian	Prussian
Latvian	0.2500000		
Prussian	0.5000000	0.3333333	
ChurchSlavonic	1.0000000	1.0000000	1.0000000

```
round(100*(dm)) # удобнее смотреть
```

	Lithuanian	Latvian	Prussian
Latvian	25		
Prussian	50	33	
ChurchSlavonic	100	100	100

Небинарные категории: в бинарные

На основе WALS.

```
df <- data.frame(  
  order = c("SVO" "SOV" "SVO" "VOS"),  
  gender = c("3" "0" "0" "0"),  
  future = c("non.inflect" "inflect" "non.inflect" "non.inflect"),  
  row.names = c("English" "Turkish" "Estonian" "Malagasy"))
```

df

	order	gender	future
English	SVO	3	non.infl
Turkish	SOV	0	infl
Estonian	SVO	0	non.infl
Malagasy	VOS	0	non.infl

```
model.matrix( . -1, data=df)
```

	orderSOV	orderSVO	orderVOS	gender3	futurenon.infl
English	0	1	0	1	1
Turkish	1	0	0	0	0
Estonian	0	1	0	0	1
Malagasy	0	0	1	0	1

Числовые категории

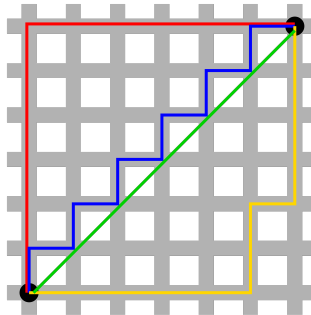
Если категории числовые, то чаще всего используют:

- евклидово расстояние

method = "euclidean"

- расстояние городских кварталов

method = "manhattan"



Картинка из [Википедии](#): зеленое — евклидово, остальные — манхэттенское.

презентация доступна: <http://goo.gl/dqocQt>

Смешанные категории

Для данных содержащих как числовые, так и категориальные данные используется алгоритм предложенный в работе [Gower 1971]. В целом, если в данных нет пропущенных значений, эта мера достаточно близка к евклидову расстоянию. В R она реализована функцией `daisy` пакета `cluster`. Вот пример на основе данных по количеству согласных и наличию абруптивных (**график**):

```
df <- read.csv("http://goo.gl/919qoS row.names = 1)
df <- df[sample(1:27, 5),] # выборка из данных
```

```
library(cluster)
dm <- daisy(df); dm # строит матрицу и вызывает ее
```

Dissimilarities :

	Japanese	Hawaiian	Lakota	Pomo
Hawaiian	0.15909091			
Lakota	0.84090909	1.00000000		
Pomo	0.75000000	0.90909091	0.09090909	
Turkish	0.20454545	0.36363636	0.63636364	0.54545455

Metric : mixed ; Types = I, N
Number of objects : 5

Метрики расстояний для строк

Для решения ряда проблем NLP было создано несколько метрик для измерения расстояний между строками. Для подсчета этих метрик в R есть несколько пакетов, я приведу примеры использования **пакета stringdist**. Наиболее популярные в лингвистике расстояния:

- Хэмминга # method = "h"
- Левенштейна (**см. визуализацию**) # method = "lv"
- косинусное # method = "c"

```
library(stringdist)
str1 <- "мама"
str2 <- "папа"
stringdist(str1, str2, method = "h")
[1] 2
```

```
str3 <- "мама"
str4 <- c("папа", "рампа", "лада", "рама")
stringdist(str3, str4, method = "lv")
[1] 2 2 2 1
```


Еще способы уменьшения размерностей?

- регрессионный анализ
- кластеризация
- многомерное шкалирование (multidimensional scaling)
- компонентный анализ (principal component analysis)
- метод k ближайших соседей (k-NN)

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений

heatmap

Как обычно, есть несколько способов визуализации:

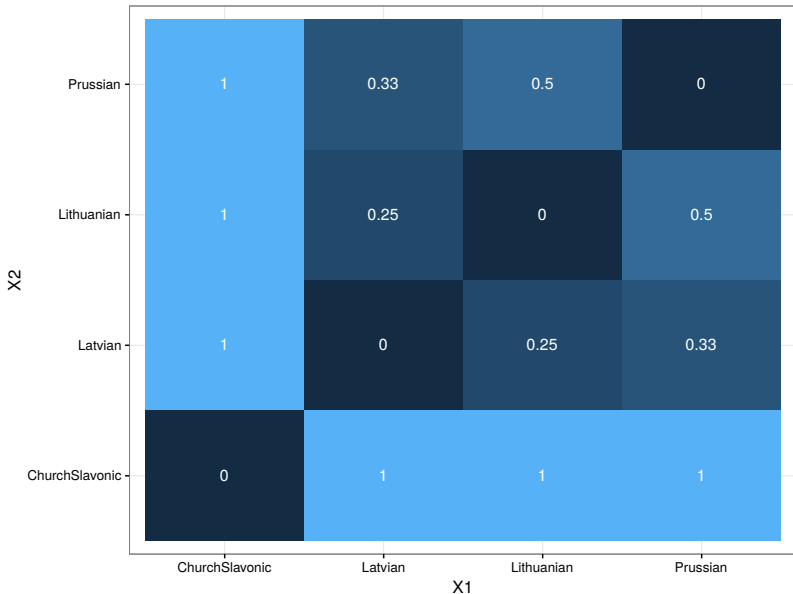
- в Rbase есть функция `heatmap()`, но ее настройка — сплошное мучение
- в `ggplot2` есть `geom_tile()`

Обе функции принимают на вход матрицы, так что результат работы функции `dist()` надо трансформировать:

```
df <- data.frame(  Lithuanian =      c(1, 1, 1, 1, 0),  
                   Latvian   =      c(1, 1, 1, 0, 0),  
                   Prussian  =      c(1, 1, 0, 0, 0),  
                   ChurchSlavonic = c(0, 0, 0, 0, 1))
```

```
df <- t(df)          # кластеризации любят держать признаки в строках  
dm <- as.matrix(dist(df, method = "binary"))
```

heatmap: ggplot



heatmap: ggplot

```
df <- data.frame(  Lithuanian =      c(1, 1, 1, 1, 0),  
                   Latvian   =      c(1, 1, 1, 0, 0),  
                   Prussian  =      c(1, 1, 0, 0, 0),  
                   ChurchSlavonic = c(0, 0, 0, 0, 1))
```

```
df <- t(df)          # кластеризации любят держать признаки в строках  
dm <- as.matrix(dist(df, method = "binary")) # считает расстояния
```

```
library(reshape)  
dm.m <- melt(dm)      # преобразования матрицы для ggplot
```

```
library(ggplot2)  
ggplot(dm.m, aes(X1, X2, fill=value)) +  
  geom_tile() + # делает heatmap  
  geom_text(aes(X1, X2, label = round(value, 2)), # пишет значения  
            color = "white", size = 4)
```

Кластеризация

Кластеризация — это не метод, а задача, для решение которой придумано множество алгоритмов. Не существует "правильных" методов кластеризации, так как "clustering is in the eye of the beholder"[Estivill-Castro 2002]. В презентации рассказывается о представителях двух семейств алгоритмов:

- метод k -средних (k -means)
- иерархическая кластеризация (hierarchical clustering)

Алгоритм k -means

Алгоритм k -means был разработан в статье [Lloyd 1982]:

- на вход алгоритму подаются данные и k — количество кластеров, на которые эти данные надо поделить;
- произвольно выбираются k точек (центроидов) и рассчитываются ближайшие расстояния (евклидово) от данных точек до центроидов, точки которые ближе всего к некоторому центроиду образуют кластер;
- на основе точек вошедших в кластер строится новый центроид, так чтобы расстояние от всех точек до нового центроида было минимально;
- часть точек становится ближе к новому центроиду и входят в его кластер, а часть от центроида отдаляется и начинают входить в другой/другие кластер/кластеры;
- ... все это повторяется, пока на некоторой итерации не происходит изменение положения центроидов.

Naftali Harris сделал [визуализация \$k\$ -means](http://goo.gl/dqocQt).
презентация доступна: <http://goo.gl/dqocQt>

метрики
расстояний
метрики расстояний
heatmap

методы
кластеризации

k -means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений

Задача

В описании нанайского языка есть гласные *i*, *ɪ* и *ə* (в данных закодированы *i*, *I*, *e* соответственно), однако совсем не понятно, одинаково ли произносят гласные *i* и *ɪ* современные носители. В датасет записаны *F1* и *F2* этих трех гласных, произнесенных в нанайских словах шестью нанайцами из двух селений Найхин и Джуен. Если *F1* и *F2* достаточно для описания разницы между этими гласными, то тогда они должны кластеризоваться.

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений

Задача

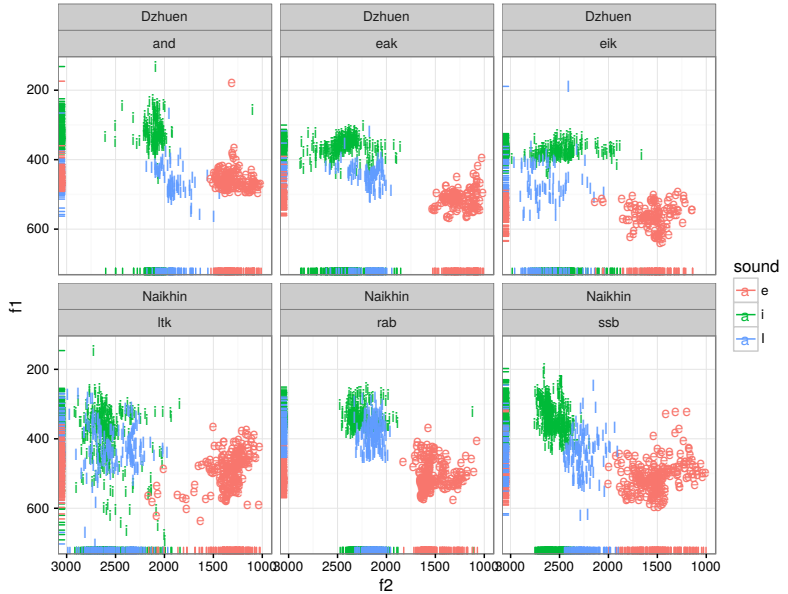
метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений



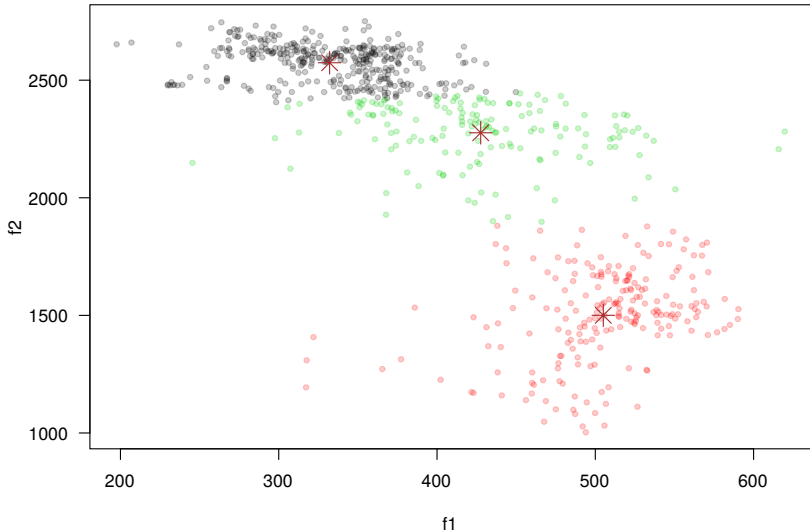
k -means

Нашим примером будет носитель ssb:

```
n <- read.csv("http://goo.gl/YPMyl2", sep = ";")
n <- n[n$dicator == "ssb",]
```

```
set.seed(5)      # устанавливаем определенное значение рандомизатора
n.cl <- kmeans(n[, c(5,6)], centers = 3)                # датафрейм, k
n.cl$cluster      # кластер каждой точки
n.cl$centers      # координаты центроидов
```

Визуализация k -means: R-base



```
plot(n[, c(5,6)],  
     col = n.cl$cluster)  
points(n.cl$centers, col = "brown", pch = 8, cex = 2)
```

данные
раскрашиваем по кластеру
центры

презентация доступна: <http://goo.gl/dqocQt>

Визуализация k -means: ggplot2

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

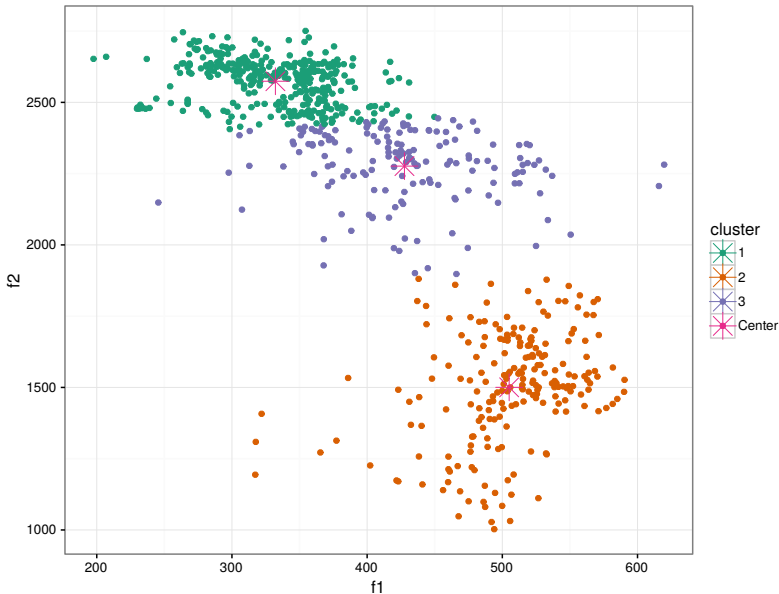
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Визуализация k -means: ggplot2

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений

```
n$cluster <- factor(n.cl$cluster)      # добавляет кластеризацию в дф
centers <- as.data.frame(n.cl$centers)  # создает дф с центроидами
```

```
library(ggplot2)
ggplot(data=n, aes(x=f1, y=f2, color=cluster)) +
  geom_point() +
  geom_point(data=centers, aes(x=f1,y=f2, color='Center'),
             shape = 8, size = 5)
```

k-means: продолжение

И что дальше? В наших данных есть информация о произнесениях, так что можно сравнить (следующий слайд) результат работы *k*-means (обозначено цветом) с тем, что ожидалось в данных словах (обозначено буквой) и посмотреть сколько раз *k*-means ошибся (слайд через один):

	and	eak	eik	ltk	rab	ssb
correct	393	426	278	515	549	682
mistaken	27	61	99	148	102	43

	and	eak	eik	ltk	rab	ssb
correct	0.94	0.87	0.74	0.78	0.84	0.94
mistaken	0.06	0.13	0.26	0.22	0.16	0.06

Кластеры k -means

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

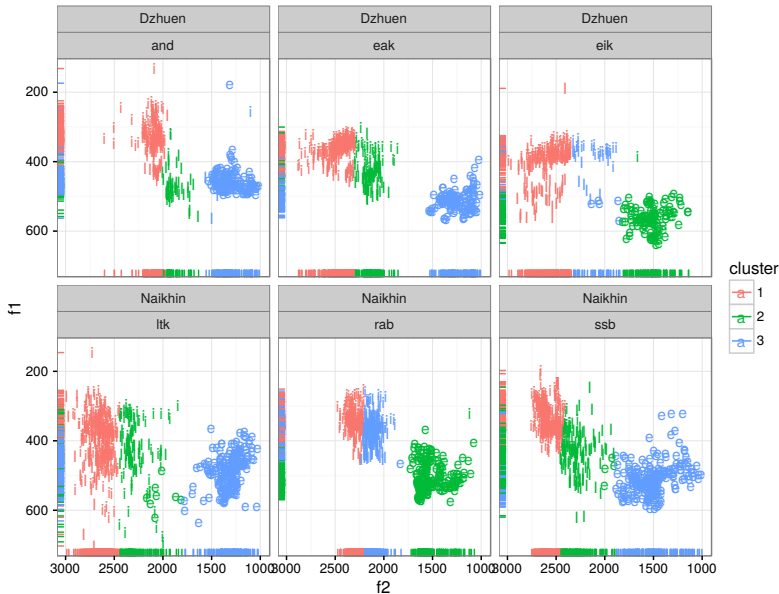
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Ошибки алгоритма k -means

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

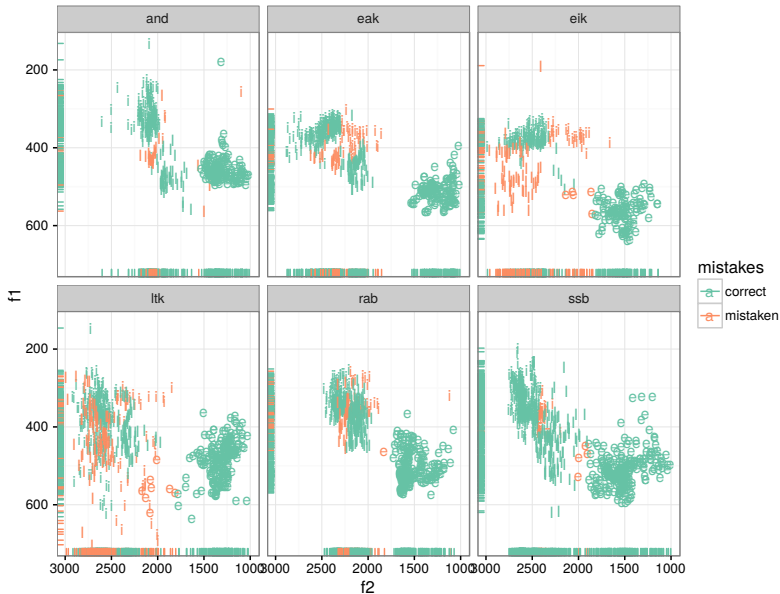
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Иерархическая кластеризация

Иерархические кластеризации имеют два типа:

- **снизу вверх (agglomerative)**: каждое наблюдение в начальной позиции является кластером, дальше два ближних кластера соединяются в один, а дендограмма отображает порядки таких соединений.
- **сверху вниз (divisive)**: все наблюдения в начальной позиции являются кластером, который дальше делится на более мелкие, а дендограмма отображает порядки таких разъединений.

Алгоритмы иерархической кластеризации требуют на вход матрицы расстояний. Алгоритмов кластерного анализа очень много, так что имеет смысл заглянуть в работу [Gordon 1987] и [на страницу CRAN](#).

Иерархическая кластеризация

Нашим примером снова будет носитель ssb:

```
n <- read.csv("http://goo.gl/YPMyl2", sep = ";")
n <- n[n$dicator == "ssb",]
```

Функция hclust принимает на вход матрицу расстояний:

```
hc <- hclust(dist(n[,c(5,6)])) # agglomerative clustering
plot(hc) # график получившихся кластеров
plot(hc, labels = F) # график без подписей
rect.hclust(hc, k=3) # выделить k кластеров
```

Функция cutree возвращает вектор номеров кластеров в соответствии с данными, так что можно строить все предыдущие графики:

```
cluster <- cutree(hc, k=3)
```

	and	eak	eik	ltk	rab	ssb
correct	393	345	234	478	494	439
mistaken	27	142	143	185	157	286

	and	eak	eik	ltk	rab	ssb
correct	0.94	0.71	0.62	0.72	0.76	0.61
mistaken	0.06	0.29	0.38	0.28	0.24	0.39

Иерархическая кластеризация

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

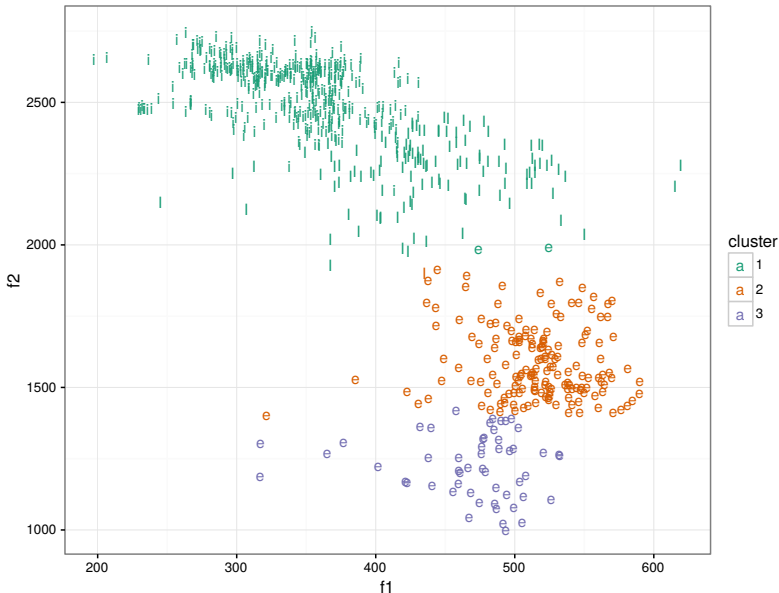
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Кластеры иерархическая кластеризации

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

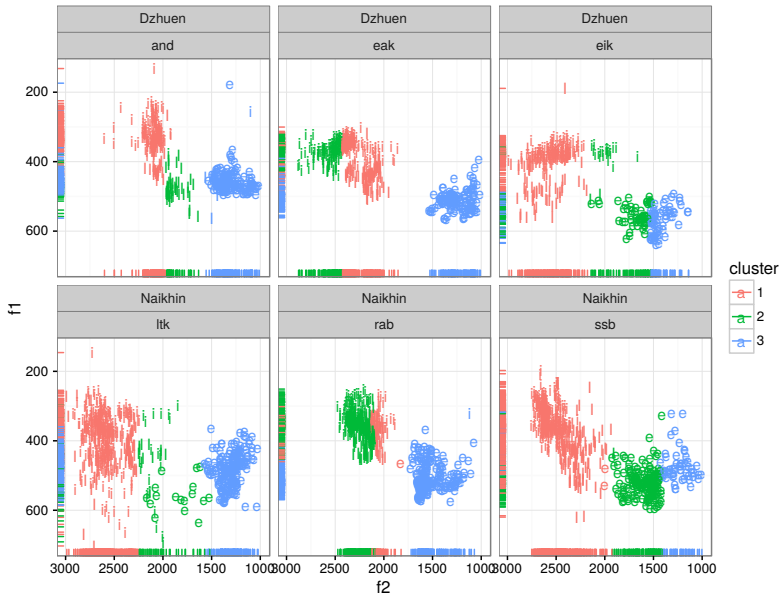
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Ошибки алгоритма иерархической кластеризации

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

hierarchical

проблемы

дендрограммы

валидации кластеров

деревья
решений



Проблемы приведенных методов

- k -means может давать разные результаты на одних и тех же данных
- при использовании k -means нужно знать k
- иерархическая кластеризация не может исправить ошибки, сделанные на предыдущих шагах: в работе [Hawkins 1982] приводится пример вектора $c(2.2, 2, 1.8, 0.1, 0.1, 1.8, 2, 2.2)$, в котором очевидны три кластера, однако если на первом этапе алгоритм разобьет все на $c(2.2, 2, 1.8, 0.1)$ и $c(0.1, 1.8, 2, 2.2)$, то дальше это исправлено не будет.

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений

Дендрограммы

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений

Дендограммой обычно называют граф, отображающий некоторые расстояния между единицами. Существует достаточно много методов построения графов на основе матрицы расстояний, напрямую связанный с используемым методом кластеризации. Надо отметить, что дендограмма это всего лишь **семейство визуализаций матрицы расстояний**. Примером для построения дендрограмм послужат данные фонетических особенностей адыгских идиомов:

```
ad <- read.csv("http://goo.gl/Rj92fh")
rownames(ad) <- ad[,1]
ad.d <- dist(ad, method = "binary")
ad.c <- hclust(ad.d)
library(ape)
ad.c <- as.phylo(ad.c)
plot(ad.c, type = "phylogram")
```

этот формат лучше воспринимается

Дендрограммы: type = "phylogram"

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

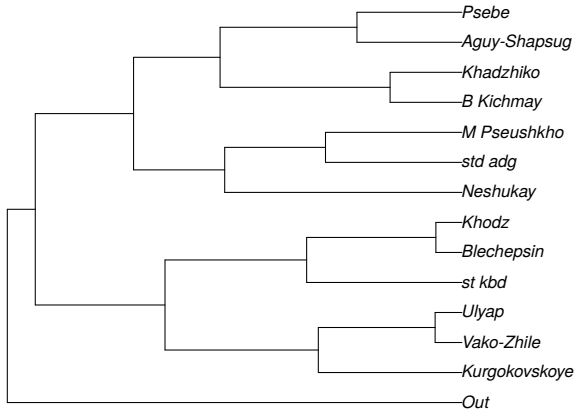
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Дендрограммы: type = "cladogram"

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

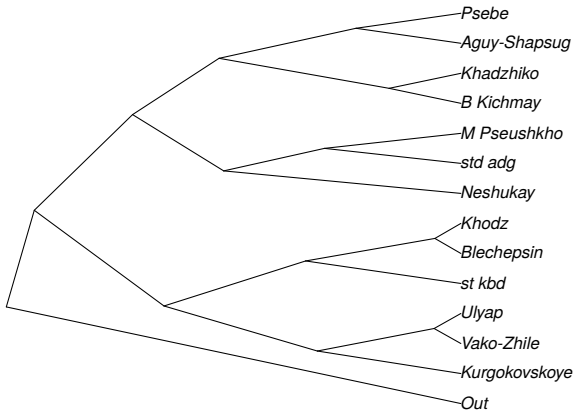
hierarchical

проблемы

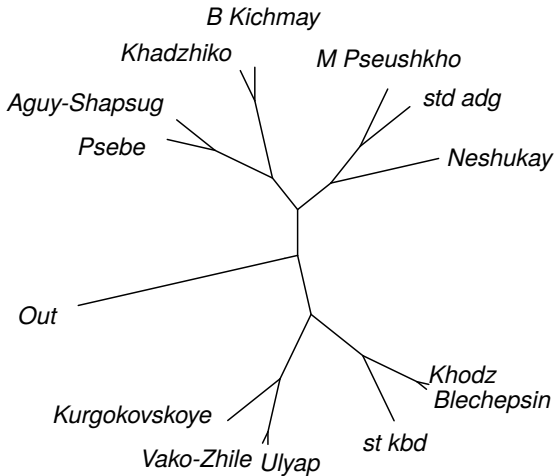
дендрограммы

валидация кластеров

деревья
решений



Дендрограммы: type = "unrooted"



Плохо работает, нужно доводить руками.

Дендрограммы: type = "fan"

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

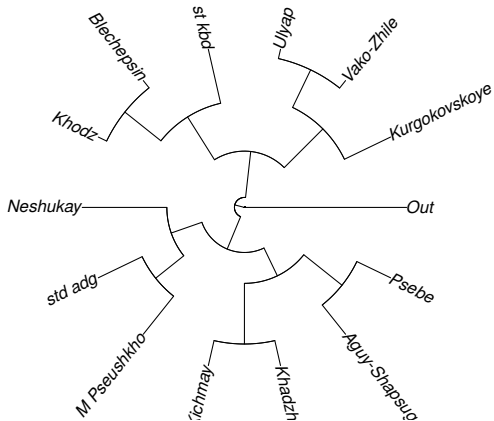
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Дендрограммы: type = "radial"

метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means

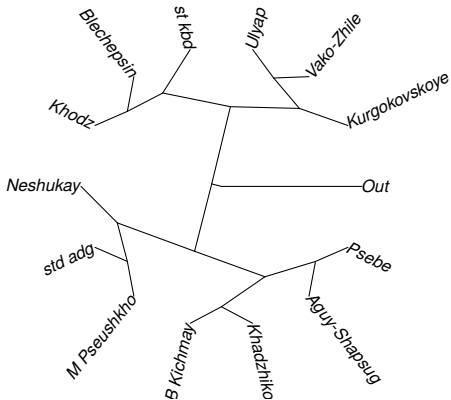
hierarchical

проблемы

дендрограммы

валидация кластеров

деревья
решений



Деревья решений

Достаточно популярным средством построения моделей является дерево решений. В узлах дерева пишутся условия, ограничивающие предикторы, на ребрах записываются значения предикторов, а на листьях дерева записаны значения предсказываемой переменной. Деревья решений позволяют решать как задачи регрессии, так и классификации.

- | | |
|---|---|
| pro ДР легко интерпретировать | contra Даже незначительные изменения в обучающих данных могут привести к значительной перестройке модели |
| pro ДР могут работать с переменными любого типа | |
| pro ДР автоматически подбирают модель, учитывая взаимодействия | contra Невысокая предсказательная точность |

Для преодоления недостатков можно использовать случайные леса (random forest) и другие ансамбли деревьев.

презентация доступна: <http://goo.gl/dqocQt>

Деревья решений

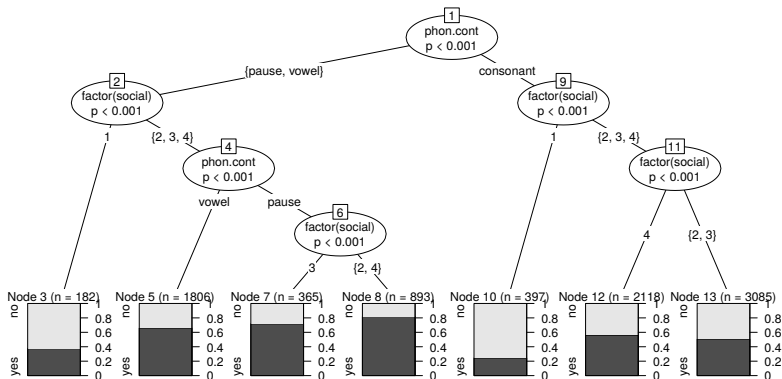
метрики
расстояний

метрики расстояний
heatmap

методы
кластеризации

k-means
hierarchical
проблемы
дендрограммы
валидация кластеров

деревья
решений



```
df <- read.csv("http://goo.gl/NwbKsN")
```

```
df$social <- factor(df$social) # внимание! числовой vs. номинативный
```

```
library(party)
```

```
fit <- alertctree(s.deletion phon.cont+social, data=df)
```

```
plot(fit)
```

Спасибо за внимание

Пишите письма
agricolamz@gmail.com

Список литературы

- Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. ACM SIGKDD explorations newsletter 4(1), 65--75.
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. Biometrics, 857--871.
- Gower, J. C. and P. Legendre (1986). Metric and euclidean properties of dissimilarity coefficients. Journal of classification 3(1), 5--48.
- Hawkins, D. M. (1982). Topics in applied multivariate analysis, Volume 1. Cambridge University Press.
- Lloyd, S. P. (1982). Least squares quantization in pcm. Information Theory, IEEE Transactions on 28(2), 129--137.