курс «Прикладные задачи анализа данных»

Функции ошибки / функционалы качества Часть 3: многоклассовые задачи, ранжирование, кластеризация

Александр Дьяконов

23 октября 2020 года

План на несколько лекций

задача регрессии

задача бинарной классификации
- чёткая классификация
- скоринговые функции

задача классификации с несколькими классами

задачи ранжирования

задачи кластеризации

Weighted kappa

Если есть разумные веса ошибок за конкретные несогласованности

Когда это бывает?

$$\kappa = 1 - \frac{\sum_{i=1}^{l}\sum_{j=1}^{l} w_{ij} m_{ij}}{\sum_{i=1}^{l}\sum_{j=1}^{l} w_{ij} s_{ij}} \in [-1, +1]$$

матрица случайных ответов

$$s_{ij} = m_{i:} m_{:j} = \sum_j m_{ij} \sum_i m_{ij}$$

$$s_{ij} \leftarrow \frac{s_{ij}}{m^2} m = \frac{s_{ij}}{\sum_{tr} s_{tr}} \sum_{tr} m_{tr}$$

квадратичные веса

$$w_{ij} = \frac{(i-j)^2}{(l-1)^2}$$

м.б. любая весовая схема

Вычисление Quadratic Weighted Kappa

ответы

	y	a
0	1	1
1	1	1
2	1	2
3	2	1
4	2	3
5	3	2
6	3	3
7	3	3
8	1	2
9	2	2

матрица ошибок

y \ a	1	2	3
1	2	2	0
2	1	1	1
3	0	1	2

матрица случайных ответов

	0	1	2
0	12	16	12
1	9	12	9
2	9	12	9

после нормировки

	0	1	2
0	1.2	1.6	1.2
1	0.9	1.2	0.9
2	0.9	1.2	0.9

матрица весов

	0	1	2
0	0.00	0.25	1.00
1	0.25	0.00	0.25
2	1.00	0.25	0.00

WK = 0.615

Вычисление Quadratic Weighted Kappa

```
from sklearn.metrics import cohen_kappa_score
cohen_kappa_score(df.y, df.a, weights='quadratic')


n = pd.crosstab(df.y, df.a)
n = n.values
m = np.outer(n.sum(axis=1) , n.sum(axis=0))
m = m / m.sum() * n.sum()
w = (np.arange(1, 4)[:,np.newaxis] -
     np.arange(1, 4)) ** 2 / ((3-1)*(3-1))
1 - (np.sum(n*w) / np.sum(m*w))
```

Потом докажем:

Каппа Коэна частный случай Weighted Kappa для 2х классов

Quadratic Weighted Kappa

Применяется в задачах, где классы упорядочены «ранжирование»

	y	1.0	0.83	0.83	0.33	0.8	0.0	-1.0
0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	1	2
2	0	0	1	0	2	0	2	2
3	1	1	1	1	1	0	0	1
4	1	1	1	1	1	0	1	1
5	1	1	0	2	1	0	2	1
6	2	2	2	2	2	2	0	0
7	2	2	2	2	2	2	1	0
8	2	2	2	1	0	2	2	0

Многоклассовая задача «Multi-label»

матрица классификаций

$$\| y_{ij} \|_{m \times l}$$

	class 1	class 2	class 3
0	1	0	0
1	0	1	0
2	0	0	1
3	1	1	0

матрица ответов

$$\| a_{ij} \|_{m \times l}$$

	class 1	class 2	class 3
0	0.75	0.00	0.25
1	0.00	0.50	0.25
2	0.25	1.00	0.25
3	0.00	0.25	0.75

По сути, надо сравнить матрицы на похожесть

микро-подход	**можно сравнивать матрицы как векторы**
макро-подход	**можно сравнивать столбцы матриц**
по объектам	**можно сравнивать строки матриц и усреднять**

Многоклассовая задача: Hamming Loss

Число (процент) ошибок

$$a_{ij} \in \{0,1\}$$

$$\mathrm{HL} = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l} I[y_{ij} \neq a_{ij}]$$

(1 – точность)

Многоклассовая задача: Log Loss (cross-entropy)

Естественное обобщение логистической ошибки

$$a_{ij} \in [0,1]$$

$$\mathrm{logloss} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij}\log a_{ij}$$

(тонкость: лучше для непересекающихся классов)

Многоклассовая задача: Mean Probability Rate

(это функционал качества, $a_{ij} \in [0,1]$ ~ распределение)

Есть макро-версия, см. дальше

$$\mathrm{MPR} = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij}a_{ij}$$

$$\mathrm{MAPR} = \frac{1}{l}\sum_{j=1}^{l}\frac{\sum_{i=1}^{m} y_{ij}a_{ij}}{\sum_{i=1}^{m} y_{ij}}$$

Многоклассовая задача: MSE, MAE

$$\mathrm{MSE} = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}(y_{ij} - a_{ij})^2$$

$$\mathrm{MAE} = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}| y_{ij} - a_{ij} |$$

это всё вариации на тему схожести / различия бинарного и вещественного вектора

Многоклассовый AUCROC: Макро-усреднение

$$\mathrm{AUC} = \frac{1}{l}\sum_{j=1}^{l}\mathrm{AUC}_j$$

AUC_j **– значение функционала в задаче бинарной классификации**

«j-й класс / не j-й класс»

$$\{(x_i, I[y(x_i)_{[j]} = 1])\}_{i=1}^{m}$$

Многоклассовый AUCROC: Весовое макро-усреднение

$$\mathrm{AUC} = \frac{\sum_{j=1}^{l} P_j \,\mathrm{AUC}_j}{\sum_{j=1}^{l} P_j}$$

P_j **– вероятность j-го класса**

(процент «1» в столбце матрицы классификации)

Многоклассовый AUCROC: Микро-усреднение

значение функционала в задаче

$$\{((x_i, j), I[y(x_i)_{[j]} = 1])\}_{i=1, j=1}^{m, l}$$

«вытягиваем матрицу ответов в вектор»

Многоклассовый AUCROC: Усреднение по объектам

$$\mathrm{AUC} = \frac{1}{m}\sum_{i=1}^{m}\mathrm{AUC}_i'$$

AUC_i' **– значение функционала в задаче**

$$\{((x_i, j), I[y(x_i)_{[j]} = 1])\}_{j=1}^{l}$$

«решение задачи по строкам»

Многомерный AUC: минутка кода

```
from sklearn.metrics import roc_auc_score

roc_auc_score(y, a, average='macro')
# эквивалентно:
auc_pclass = [roc_auc_score(y[:,i], a[:,i]) for i in range(l)]
auc_pclass, mean(auc_pclass)

roc_auc_score(y, a, average='micro')
# эквивалентно:
roc_auc_score(y.ravel(), a.ravel())

roc_auc_score(y, a, average='weighted')
# эквивалентно:
w = y.sum(axis=0)
sum(np.array(auc_pclass) * w) / sum(w)

roc_auc_score(y, a, average='samples')
# эквивалентно:
auc_pinstance = [roc_auc_score(y[i,:], a[i,:]) for i in
range(m)]
auc_pinstance, mean(auc_pinstance)
```

Многомерный AUC ROC

матрица классификаций

	class 1	class 2	class 3
0	1	0	0
1	0	1	0
2	0	0	1
3	1	1	0

матрица ответов

	class 1	class 2	class 3
0	0.75	0.00	0.25
1	0.00	0.50	0.25
2	0.25	1.00	0.25
3	0.00	0.25	0.75

macro	micro	weighted	samples
0.49	0.53	0.52	0.56

	class 0	class 1	class 2
AUC_per_class	0.62	0.5	0.33
P_per_class	0.50	0.5	0.25

	class 0	class 1	class 2	class 3
AUC_per_instance	1.0	1.0	0.25	0.0

Точность: сравнение макро- и микро- усреднения

$$\mathrm{P}_j = \frac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j}$$

$$\mathrm{P}_{\text{macro-mean}} = \frac{1}{l}\sum_{j=1}^{l}\mathrm{P}_j$$

$$\mathrm{P}_{\text{micro-mean}} = \frac{\sum_{j=1}^{l}\mathrm{TP}_j}{\sum_{j=1}^{l}\mathrm{TP}_j + \sum_{j=1}^{l}\mathrm{FP}_j}$$

Кстати, макро-усреднение делают по-разному.

Часто: среднее геометрическое.

Точность: сравнение макро- и микро- усреднения

При вычислении каких-то функционалов, например точности

МАКРО	• вычислить точность для каждого класса • усреднить полученные точности `(H.TP / (H.TP + H.FP)).mean()`	0.344

МИКРО	• вычислить точность сразу для всех классов
	
	`TP = H.TP.sum() + H.TP.sum()`
	`FP = H.FP.sum() + H.FP.sum()`
	`P = TP / (TP + FP)`
		0.246

	TP	FP
class 1	2	2
class 2	5	10
class 3	10	40

	P
class 1	0.50
class 2	0.33
class 3	0.20

Точность: сравнение макро- и микро- усреднения

	TP	FP
class 1	2	2
class 2	5	10
class 3	100	400

не изменились точности по классам $\Rightarrow$ не изменилась макро-точность 0.344

изменились TP, FP по классам $\Rightarrow$ микро-точность смещается в сторону «большего» класса: 0.206 (вместо 0.246)

где какое усреднение лучше использовать?

Совет: смотреть дисперсию показателей по классам

F-мера – ещё больше вариантов усреднения

Макро F-мера

$$\mathrm{F}_{\text{macro-mean}} = \frac{1}{l}\sum_{j=1}^{l} \mathrm{F}_j$$

на основе других макро-параметров

$$\mathrm{F} = \frac{2}{\dfrac{1}{\mathrm{P}_{\text{macro-mean}}} + \dfrac{1}{\mathrm{R}_{\text{macro-mean}}}}$$

ДЗ провести сравнение!

Сбалансированная точность «Balanced accuracy» – макро-усреднение полноты

Сбалансированная точность (accuracy) не есть усреднение точностей (precision)

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} R_j = \frac{1}{l}\sum_{j=1}^{l} \frac{\sum_{t=1}^{m} I[y(x_t)_{[j]}=1] I[a(x_t)_{[j]}=1]}{\sum_{t=1}^{m} I[y(x_t)_{[j]}=1]}$$

```
from sklearn.metrics import balanced_accuracy_score
```

Другие (неэквивалентные) определения:

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} \min[P_j, R_j]$$

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} \min[\mathrm{sens}_j, \mathrm{spec}_j]$$

ДЗ Сравните разные подходы

Пример: соревнование LSHTC

```
Id,Predicted
1,12 35 200
2,54 55
3,11
4,1 7 101
...
```

$$\tilde{F} = \frac{2\tilde{P}\tilde{R}}{\tilde{P} + \tilde{R}}$$

$$\tilde{P} = \frac{1}{l}\sum_{j=1}^{l}\frac{TP_j}{TP_j + FP_j}$$

$$\tilde{R} = \frac{1}{l}\sum_{j=1}^{l}\frac{TP_j}{TP_j + FN_j}$$

Решающее правило с отсечкой:

$$\alpha_{ij} = \begin{cases} 1, & \gamma_{ij} \geq \min(c, \max\{\gamma_{ij}\}_{j=1}^{l}), \\ 0, & \text{иначе.} \end{cases}$$

Решать задачу по вертикали / по горизонтали

Оценка результатов поиска/рекомендаций

Задача с бинарной релевантностью

$$x_1 \prec x_2 \prec \ldots \prec x_m$$

$y_i = 1$ – **релевантный объект**

$y_i = 0$ – **нерелевантный объект**

Задача ранжирования

Целевой признак может быть бинарным, но это не задача классификации

Precision at n

Точность на первых n элементах

$$p@n = \frac{y_1 + \ldots + y_n}{n}$$

Average Precision at n

Средняя точность на первых n элементах

$$ap@n=\sum_{k=1}^{n}\frac{P(k)}{\min(n,r)}$$

r **– мощность множества релевантных объектов**

(товаров, документов)

n **– сколько рекомендаций будет учитываться**

$$P(k)=\begin{cases} p@k, & y_k=1,\\ 0, & y_k=0,\end{cases}$$

y_i **– бинарное значение релевантности**

Mean Average Precision

– усреднение ap@n по всем пользователям

Average Precision at n

Примеры (три релевантных объекта, $r=3$):

$0 \prec 0 \prec 0$
$$ap@3=\frac{1}{3}[0+0+0]$$

$0 \prec 0 \prec 1$
$$ap@3=\frac{1}{3}\left[0+0+\frac{1}{3}\right]$$

$0 \prec 1 \prec 1$
$$ap@3=\frac{1}{3}\left[0+\frac{1}{2}+\frac{2}{3}\right]$$

$1 \prec 0 \prec 0$
$$ap@3=\frac{1}{3}\left[\frac{1}{1}+0+0\right]$$

$0 \prec 0 \prec 1 \prec 1 \prec 1$
$$ap@5=\frac{1}{3}\left[0+0+\frac{1}{3}+\frac{2}{4}+\frac{3}{5}\right]$$

$1 \prec 1 \prec 1 \prec 0 \prec 0$
$$ap@5=\frac{1}{3}\left[\frac{1}{1}+\frac{2}{2}+\frac{3}{3}+0+0\right]$$

Concordant – Discordant ratio

$$\frac{|\{(i,j)\,|\,y_i > y_j, 1 \le i < j \le m\}|}{|\{i\,|\,y_i = 1\}| \cdot |\{j\,|\,y_j = 0\}|}$$

Упорядочили: E, D, C, B, A (по убыванию релевантности)

На самом деле: B, E – релевантные

Пары «релевантный» – «нерелевантный»:

BA	**EA**
BC	**EC**
BD	**ED**

Качество упорядочивания: 4 / (2 + 4)

~ AUC ROC

Что ещё может встретиться… в задачах рекомендации

$$\frac{1}{|Z|}\sum_{z\in Z}\frac{|\{x_1,\ldots,x_z\}\cap\{x'_1,\ldots,x'_z\}|}{z}$$

$x_1,\ldots,x_n$ – **упорядоченный** список ответов

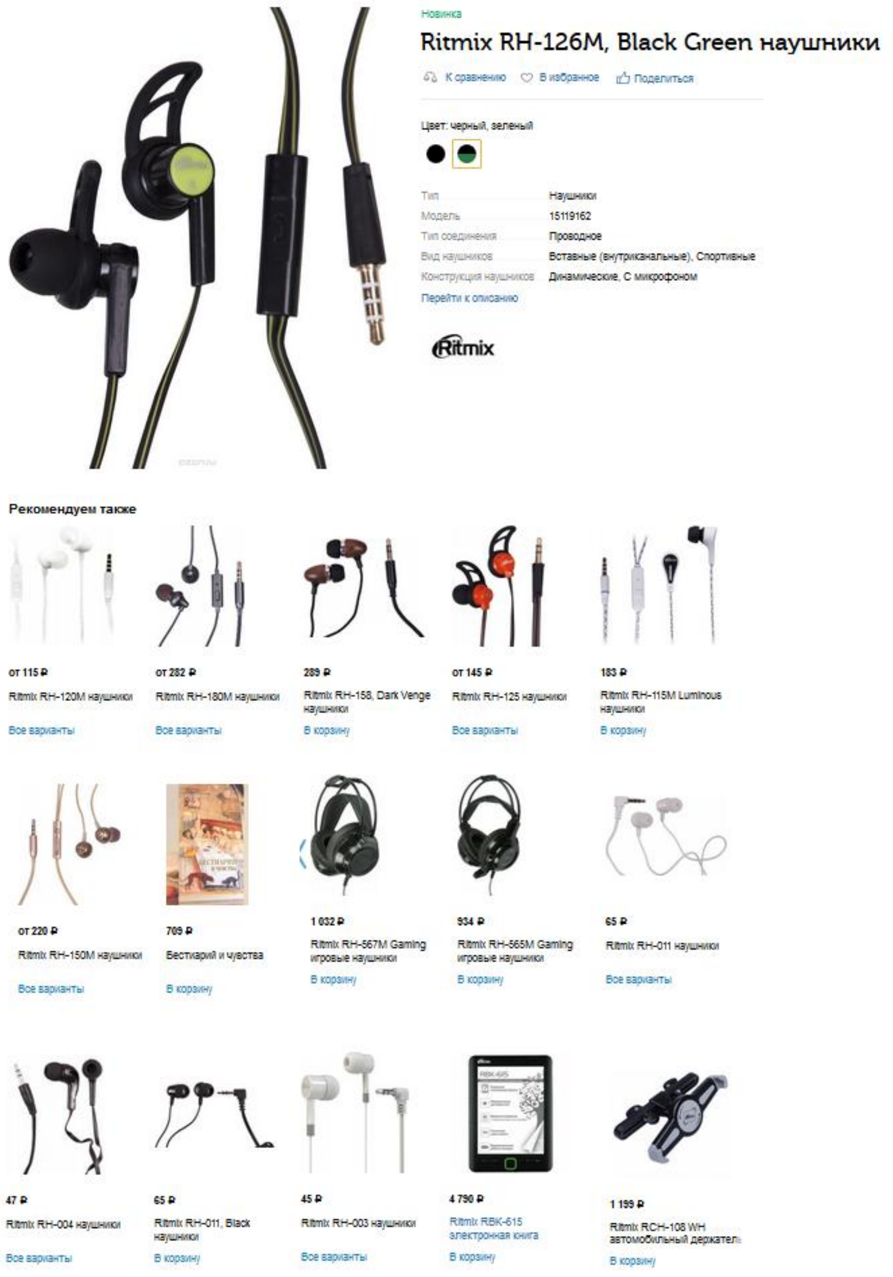
$x'_1,\ldots,x'_m$ – все релевантные

$$Z\subseteq\{1,2,\ldots,n\}$$

$$Z=\{5,10,15,20,25,30\}$$

когда логично применить?

Рекомендации



Mean Reciprocal Rank (MRR)

– это усреднение Reciprocal rank (RR) по всем ранжированиям, который сделал алгоритм.

$$RR = \frac{1}{\min\{i : y_i = 1\}}$$

Часто оптимизируют именно его!

Классические функционалы в поиске

Случай небинарной релевантности

Выдали id документов/товаров/..., а их ценность (релевантность):

$$y_1, \ldots, y_m$$

Cumulative Gain at n

$$\mathrm{CG@}n = y_1 + \ldots + y_n$$

Discounted Cumulative Gain at n

$$\mathrm{DCG@}n = \sum_{i=1}^{n} \frac{2^{y_i} - 1}{\log_2(i+1)}$$

Ещё вариант:

$$DCG@n = y_1 + \sum_{i=2}^{n} \frac{y_i}{\log_2(i)} = y_1 + y_2 + \frac{y_3}{\log_2 3} + \ldots + \frac{y_n}{\log_2 n}$$

Цена ошибок за неправильное ранжирование

$$\frac{1}{\log_2(1+1)} - \frac{1}{\log_2(1+2)} \approx 0.37$$

$$\frac{1}{\log_2(1+10)} - \frac{1}{\log_2(1+11)} \approx 0.01$$

$$\frac{1}{\log_2(1+10)} - \frac{1}{\log_2(1+20)} \approx 0.06$$

Normalized DCG

$$nDCG = \frac{DCG}{IDCG}$$

IDCG = ideal DCG

для того, чтобы не было зависимости от длины выдачи

Ещё подход к сравнению порядков:

Пусть алгоритм выдал

$$x_1 \prec x_2 \prec \ldots \prec x_m$$

Правильный порядок

$$x_{i_1} \prec x_{i_2} \prec \ldots \prec x_{i_m}$$

Надо сравнить:

$$(1, 2, \ldots, m)$$

$$(i_1, i_2, \ldots, i_m)$$

Ранговые корреляции…

Ещё подход к оценке ранжирования

Известны вероятности того, что объект является релевантным

$$p_i = p(x_i)$$

~ пользователь выберет ссылку

Expected reciprocal rank (ERR)

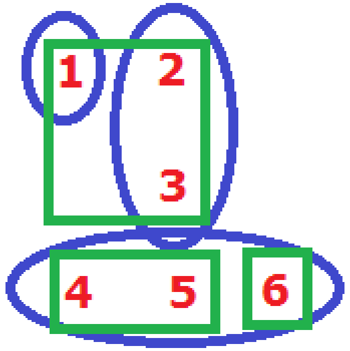
$$ERR@n = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{k}p_k\prod_{i<k}(1-p_i)$$

Как интерпретировать?

Редакторское расстояние

Операции

добавление к кластеру
создание кластера с одним объектом
удаление из кластера
удаление кластера с одним объектом



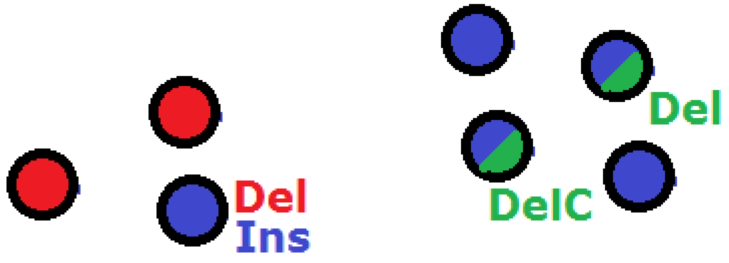

```
1 2 3;4 5;6
1 2 3; 4 5  [delC]
2 3; 4 5  [del]
2 3; 4 5; 1  [insC]
2 3; 4 5 6; 1  [ins]
```

	2 3	4 5 6	1
1 2 3	1	6	2
4 5	4	1	3
6	3	2	2

Редакторское расстояние

- Плохо заносить не в тот кластер (целых две операции на перенос)
- Плохо создавать неправильный кластер

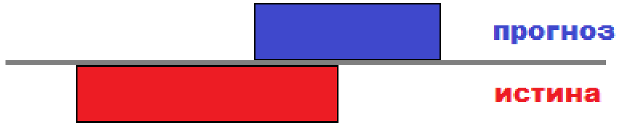
$\Rightarrow$ осторожный алгоритм



- Многое зависит от операций…

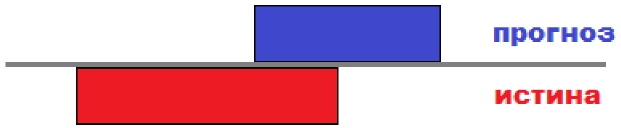
Задача с «неклассическим целевым вектором»

Надо предсказывать не значение, а интервал [a, b]



Как измерить качество?

Задача с интервальным целевым вектором



Интервал – это множество!

Коэффициент Жаккара (Jaccard)

$$\frac{|A \cap B|}{|A \cup B|}$$

коэффициент Шимкевича-Симпсона (Szymkiewicz, Simpson)

$$\frac{|A \cap B|}{\min(|A|,|B|)}$$

коэффициент Браун-Бланке (Braun-Blanquet)

$$\frac{|A \cap B|}{\max(|A|,|B|)}$$

См. википедию «Коэффициент сходства» для переноса идеи Колмогорова об обобщённом среднем...

Вариации на тему усреднения…

коэффициент Сёренсена (Sörensen)

$$\frac{2|A\cap B|}{|A|+|B|}$$

коэффициент Кульчинского (Kulczinsky)

$$\frac{|A\cap B|}{2}(1/|A|+1/|B|)$$

коэффициент Отиаи (Ochiai)

$$\frac{|A\cap B|}{\sqrt{|A|\cdot|B|}}$$

Меры включения

$$\frac{|A\cap B|}{|A|} \qquad \frac{|A\cap B|}{|B|}$$

$$\frac{|A\cap B|}{2|A|-|A\cap B|} \qquad \frac{|A\cap B|}{2|B|-|A\cap B|}$$

Как решать задачи с интервалами? Потом вернёмся…

Оценка результатов кластеризации

Если знаем верную кластеризацию… внешняя оценка (External evaluation)

Вопрос: когда?

ничего не знаем $\Rightarrow$ согласованность с данными

внутренняя оценка (Internal evaluation)

Оценка результатов кластеризации: «Internal evaluation»

Пусть чёткая (нет пересечений) кластеризация $U = u_1 \cup \ldots \cup u_{|U|}$

множества $X = \{x_1, \ldots, x_m\}$

Davies–Bouldin index

Использует центроиды и дисперсии

$$DB = \frac{1}{|U|} \sum_{i=1}^{|U|} \max_{j \ne i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

Dunn index =

min между кластерами / max внутри кластерами

$$D = \frac{\min\limits_{1 \le i < j \le |U|} d(u_i, u_j)}{\max d_{\text{in}}(u_i)}$$

Silhouette

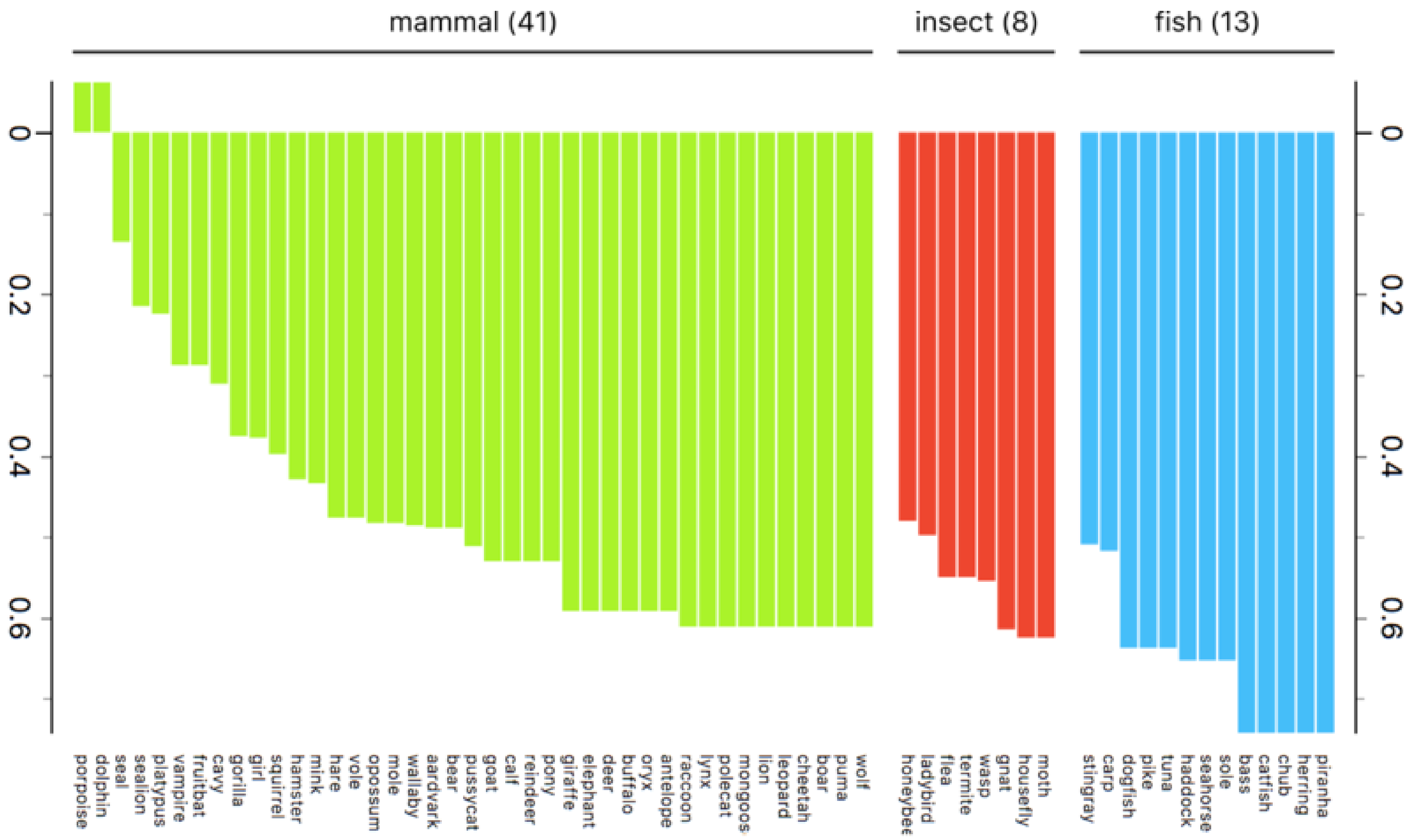
$x_i \in u_1, \; d(x_i, u_2) \le d(x_i, u_3) \le \ldots$

Расстояние считается как среднее до всех точек кластера

$$\text{silhouette}(x_i) = \frac{d(x_i, u_2) - d(x_i, u_1)}{\max(d(x_i, u_2), d(x_i, u_1))}$$

Можно усреднять по точкам

Оценка результатов кластеризации: «Internal evaluation»



https://en.wikipedia.org/wiki/Silhouette_(clustering)

Calinski-Harabasz Index (Variance Ratio Criterion)

$$\frac{\operatorname{trace}\left(\frac{1}{|U|-1}\sum_{i=1}^{|U|}|U_i|(x-c_i)(x-c_i)^{\mathrm{T}}\right)}{\operatorname{trace}\left(\frac{1}{m-|U|}\sum_{i=1}^{|U|}\sum_{x\in U_i}(x-c_i)(x-c_i)^{\mathrm{T}}\right)}$$

след матрицы межклассовой ковариации / след матрицы внутриклассовой ковариации

лучше подходит для выпуклых кластеров и евклидовой метрики

External evaluation: взаимная информация

Пусть чёткие (нет пересечений) кластеризации

$$U = u_1 \cup \ldots \cup u_{|U|}$$

$$V = v_1 \cup \ldots \cup v_{|V|}$$

множества $X = \{x_1, \ldots, x_m\}$

$$p_i = \frac{|u_i|}{m}$$

$$H(U) = -\sum_{i=1}^{|U|} p_i \log p_i$$

Аналогично $H(V)$

$$p_{ij} = \frac{|u_i \cap v_j|}{m}$$

$$MI = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} p_{ij} \log \frac{p_{ij}}{p_i p_j}$$

потом MI ~ насколько более чётко определена U при знании V

уже её можно использовать…

External evaluation: скорректированная взаимная информация

Adjusted mutual information

$$AMI(U,V)=\frac{MI(U,V)-\mathbf{E}(MI(U,V))}{\max(H(U),H(V))-\mathbf{E}(MI(U,V))}$$

1 – если кластеризации равны

~0 – если кластеризации случайны

матожидание можно вычислить аналитически

нужно калибровать, т.к. чем больше кластеров в кластеризациях, тем больше значение MI

```
from sklearn.metrics import mutual_info_score # MI
from sklearn.metrics import normalized_mutual_info_score # [0, 1]
from sklearn.metrics.cluster import adjusted_mutual_info_score
adjusted_mutual_info_score([0, 0, 1, 1], [0, 0, 1, 1])
```

https://en.wikipedia.org/wiki/Adjusted_mutual_information

ДЗ параметрический пример, в котором AMI меняется в своих пределах

External evaluation: V-мера

V – среднее гармоническое homogeneity и completeness

homogeneity ~ каждый кластер содержит только объекты отдельного класса
completeness ~ все объекты конкретного класса отнесены в один кластер

```
from sklearn.metrics.cluster import homogeneity_score

from sklearn.metrics.cluster import completeness_score

from sklearn.metrics.cluster import v_measure_score

v_measure_score([0, 0, 1, 1], [0, 0, 1, 1])
```

External evaluation: Adjusted Rand index

Аналогичная «Adjusted» идея, но проще…

поскольку кластеризация задаёт отношение эквивалентности

Rand index

$$R=\frac{|\{i,j:(i\sim_U j)\,\&\,(i\sim_V j)\}|+|\{i,j:(i\nsim_U j)\,\&\,(i\nsim_V j)\}|}{C_m^2}$$

теперь калибровка под случайную кластеризацию:

$X \backslash Y$	Y_1	Y_2	$\ldots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\ldots$	b_s	

$$\overbrace{ARI}^{\text{Adjusted Index}}=\frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}}-\overbrace{\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right]}_{\text{Max Index}}-\underbrace{\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{n}{2}}_{\text{Expected Index}}}$$

```python
from sklearn.metrics.cluster import adjusted_rand_score
adjusted_rand_score([0, 0, 1, 1], [0, 0, 1, 1])
```

https://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index

External evaluation: общий подход

Кластеризация ~ классификация пар

$$\{x_1,\ldots,x_m\} \to \{(1,1),\ldots,(i,j),\ldots,(m,m)\}$$

$$a_U(i,j)=1 \Leftrightarrow i \sim_U j$$

Можно сравнивать классификации a_U **и** a_V

Пример, Rand index: $\mathrm{RI} = \dfrac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$

Fowlkes-Mallows index (FMI)

– среднее геометрическое точности и полноты

$$\mathrm{FMI} = \frac{\mathrm{TP}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})}}$$

```
from sklearn.metrics.cluster import fowlkes_mallows_score
fowlkes_mallows_score([0, 0, 1, 1], [0, 0, 1, 1])
```

есть и много других…

Литература

К.Д. Маннинг, П. Рагхаван, Х. Шютце «Введение в информационный поиск». — Вильямс, 2011.