Алгоритмы на ординальных данных. Lens Depth Function

Вотинов Антон

Научный руководитель - Панов Максим

19.05.2016

ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Ординальные данные

Что это?

Ординальные данные дают бинарный ответ на сравнение расстояний.
Примеры ординальных данных:

- $d(A, B) < d(C, D)$ - расстояние между A и B меньше, чем между C и D;
- $(d(A, B) < d(B, C)) \wedge (d(A, B) < d(A, C))$ - C является выбросом в тройке (A,B,C);
- k-NN.

Ординальные данные являются естественными для некоторых приложений:

- Краудсорсинг;
- Поисковые системы;
- Снижения влияния ошибок в измерениях.

Ординальные данные

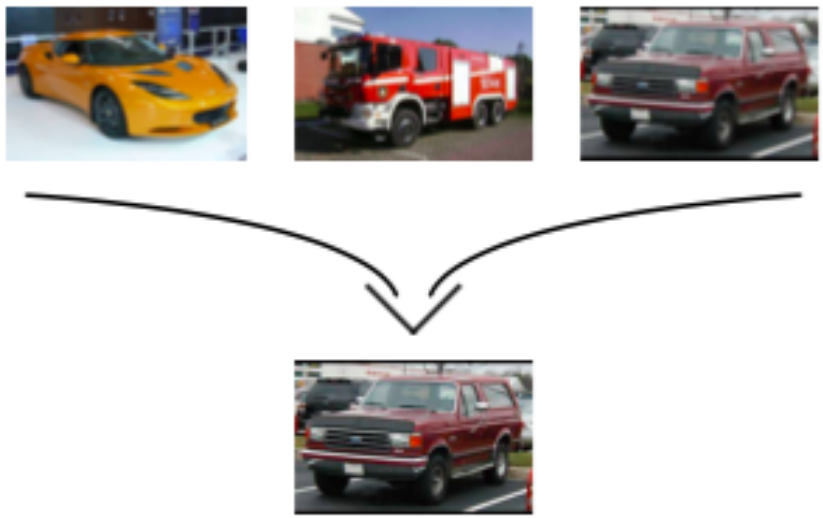
Постановка задачи Luxburg

Lens depth function and k-relative neighborhood graph: versatile tools for ordinal analysis.

Для тройки объектов (A, B, C) имеет формализованное сравнение вида:

$$(d(A, B) < d(B, C)) \wedge (d(A, C) < d(B, C))$$

«Объект A является самым центральным среди тройки (A,B,C) »

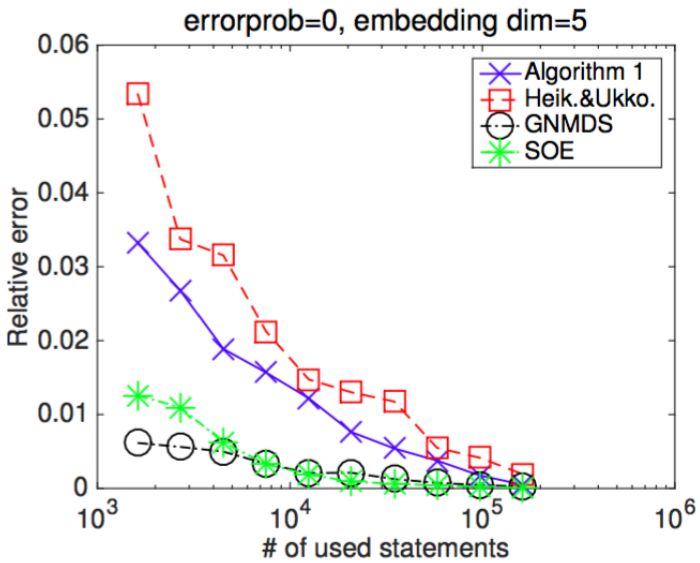


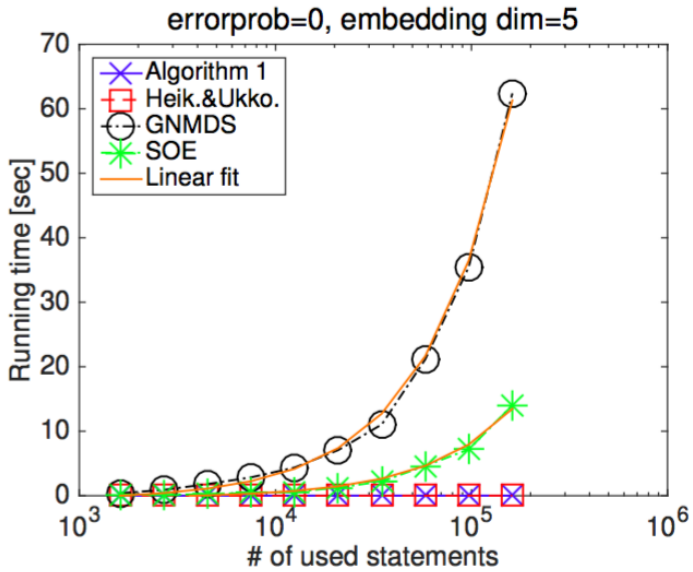
Ординальные данные

Алгоритмы на ординальных данных

Такое представление ординальных данных позволяет решать некоторые задачи:

1. Поиск медоида;
2. Поиск выбросов;
3. Классификация;
4. Кластеризация.





Ординальные данные

Поиск медоида

Медоид - наиболее «центральный» элемент для набора наблюдений D.

Алгоритм

Input: a collection S of statements of the kind (*) from some data set D
Output: as estimate of a medoid of D

1. For every object O in D compute

$$LD(O) := \frac{n_O^C}{n_O}$$

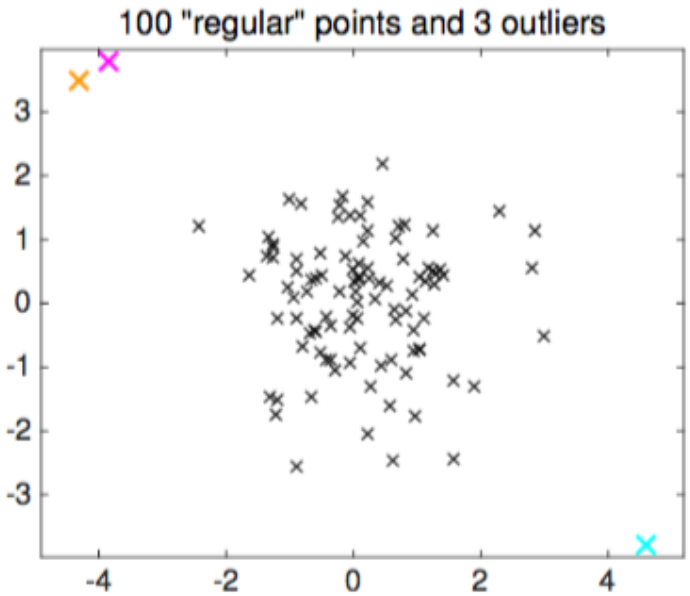
   n_O^C - number of statements comprising O as most central object, n_O - number of statements comprising O. If n_O zero, set $LD(O) = 0$;

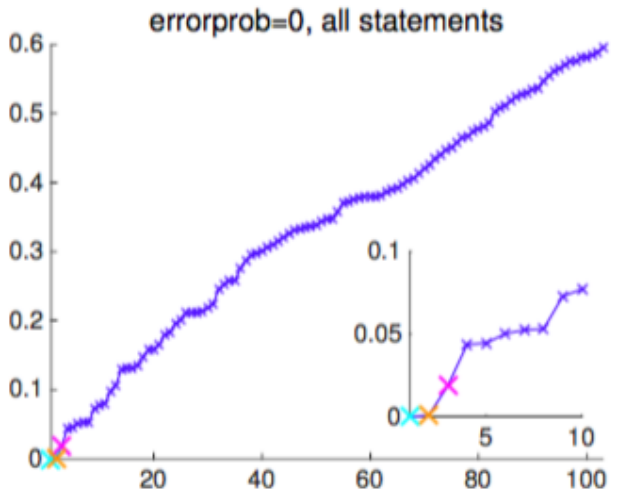
2. Retrun an object O for which $LD(O)$ is maximal.

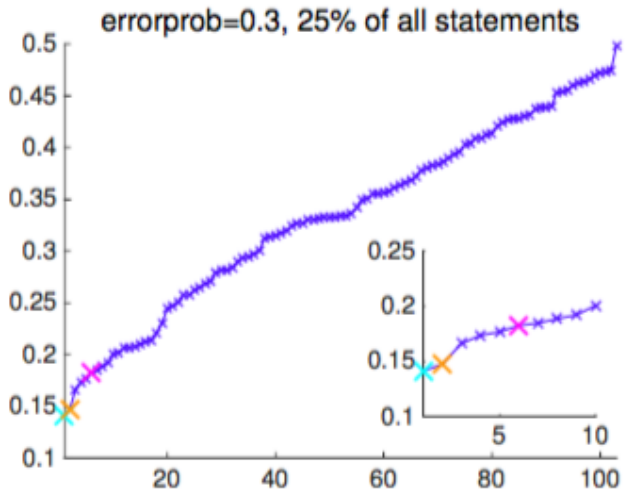
Ординальные данные

Пример поиска выброса

Поиск выбросов - противоположная задача для поиска медоида.







Ординальные данные

Классификация

Алгоритм

Input: a collection S of statements of the kind (*) from some data set D somprising a set L of labeled objects and a set U of unlabeled objects; there are K classes

Output: an inferred class label for every unlabeled object in U

1. For every object O in D and $i \in \{1, ..., K\}$ compute
   $N_{C_i}(O)$ - number of statements comprising O and two labeled objects from $Class_i$ with O as most central
   $D_{C_i}(O)$ - number of statements comprising O and two labeled objects from $Class_i$
   $LD_{C_i}(O) := \frac{N_{C_i}(O)}{D_{C_i}(O)}$;
2. train an arbitrarry classifier with training data:
   $\{(LD_{C_1}(O), ..., LD_{C_K}(O)) : O \subseteq L\} \in R^K$,
   where the label of $\{(LD_{C_1}(O), ..., LD_{C_K}(O))\}$ equals the label of O;
3. predict class label of every unlabeled object $O_u \in U$ by the classifier.

Ординальные данные

Пример классификации

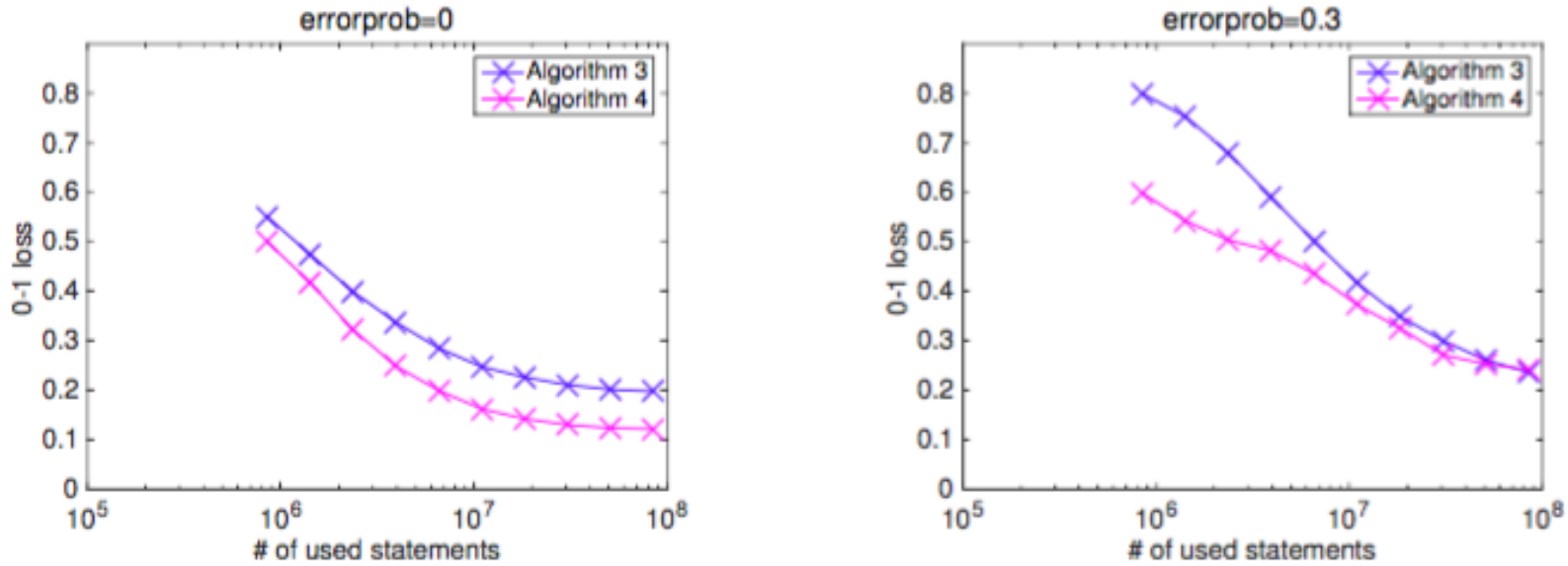


Figure 16: **Classification — 300 labeled and 500 unlabeled USPS digits with Euclidean metric. k-NN classifier on top of Algorithm 3.** 0-1 loss (22) for Algorithm 3 (in blue) and for Algorithm 4 (in pink) as a function of the number of provided statements of the kind ($\star$). The set of all possible statements contains $\binom{800}{3} \approx 8.5 \cdot 10^7$ statements corresponding to the rightmost measurement.

Ординальные данные

Кластеризация

Алгоритм

Input: a collection S of statements of the kind (*) from some data set D; k - the size of the neighborhood; l - number of clusters; σ - for weighted version
Output: a clustering $C_1, ..., C_l \subseteq D$

1. For every pair (O_i, O_j) of objects in D compute:
   $N(O_i, O_j)$ - number of statements comprising both O_i and O_j and another object as most central;
   $D(O_i, O_j)$ - number of statements comprising both O_i and O_j;
   $V(O_i, O_j) = \frac{N(O_i, O_j)}{D(O_i, O_j)}$;
2. Let $W = (w)_{ij}$ be a $(n \times n)$ matrix with:
   $$w_{ij} = \begin{cases} e^{-\frac{V(O_i, O_j)}{\sigma^2}} & \text{if } V(O_i, O_j) < \frac{k}{|D|-2} \\ 0 & \text{else} \end{cases}$$
3. Apply spectral clustering to W with l as input parameter for the number of clusters;
4. return clusters $C_1, ..., C_l$

Ординальные данные

Пример кластеризации

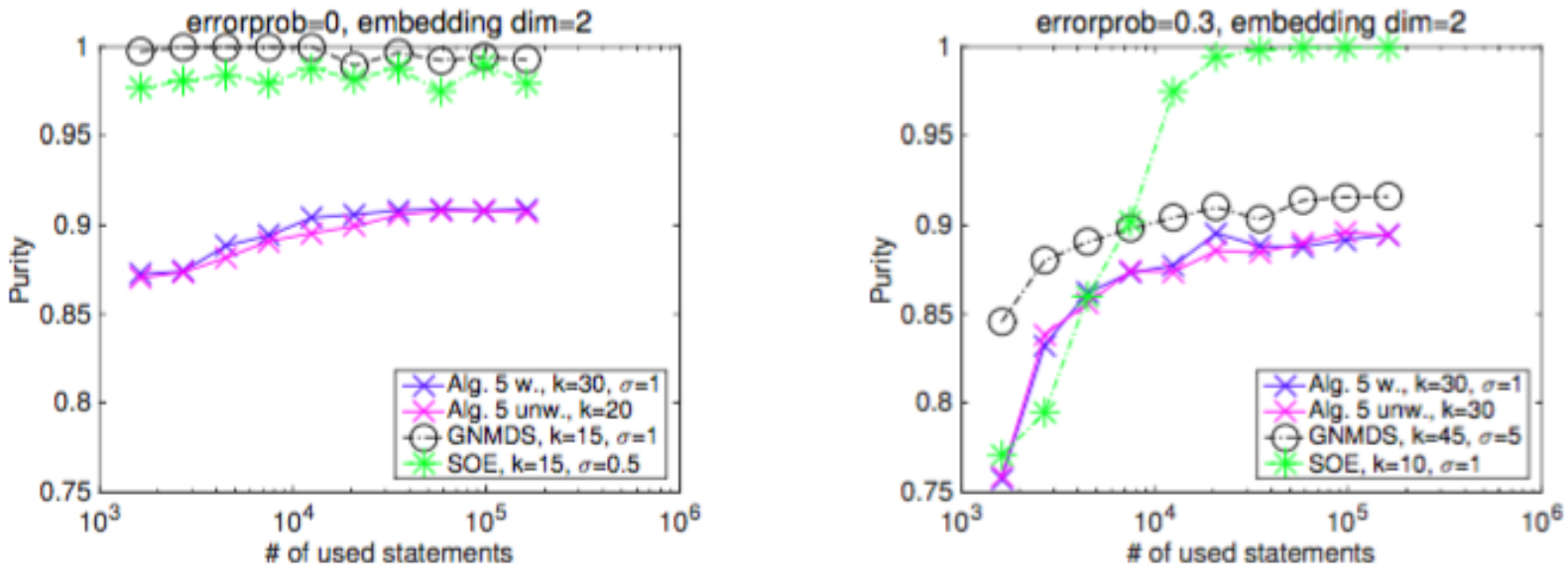


Figure 17: **Clustering — 100 points from a 2-dim uniform distribution on two equally sized moons with Euclidean metric.** Purity (23) for Algorithm 5 in its weighted (in blue) and unweighted version (in pink) and for an embedding approach using GNMDS (in black) or SOE (in green) as a function of the number of provided statements of the kind (⋆). The set of all possible statements contains $\binom{100}{3} = 161700$ statements corresponding to the rightmost measurement.

Lens Depth Function

Определение

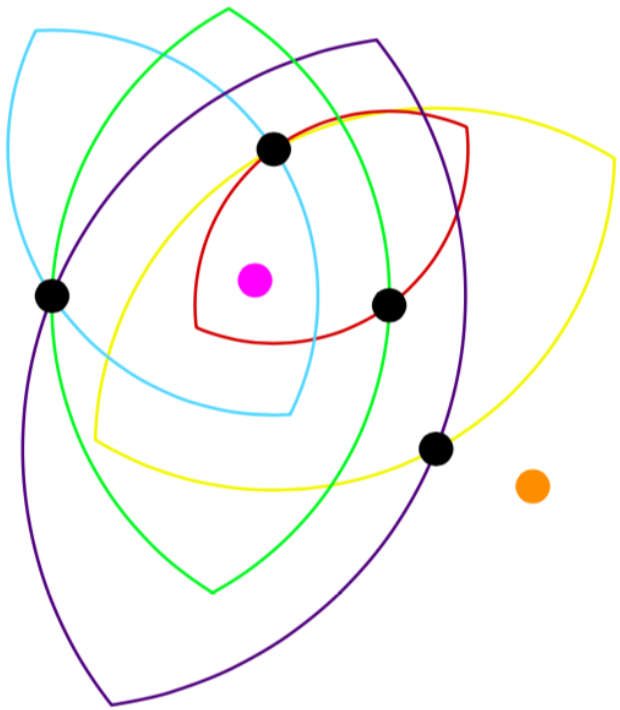
Определим линзу:

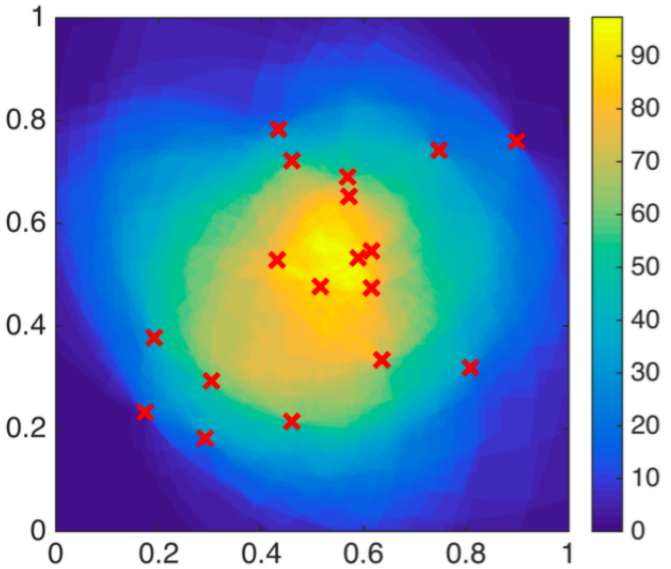
$$Lens(x_i, x_j) = \{x \in \chi : max\{d(x, x_i), d(x, x_j)\} < d(x_i, x_j)\}$$

Определим Lens Depth Function:

$$LD(x; D) = Prob(x \in Lens(x_i, x_j))$$

Заметим, что $LD(O)$ из алгоритма оценивает $LD(x; D)$.

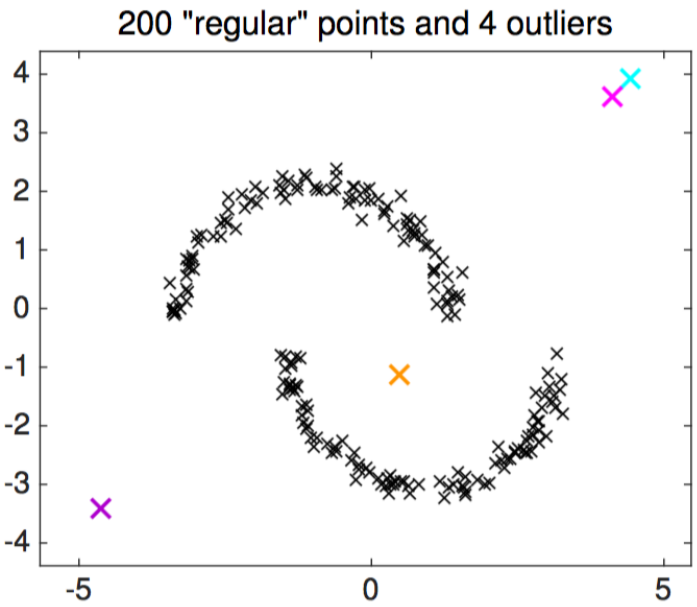


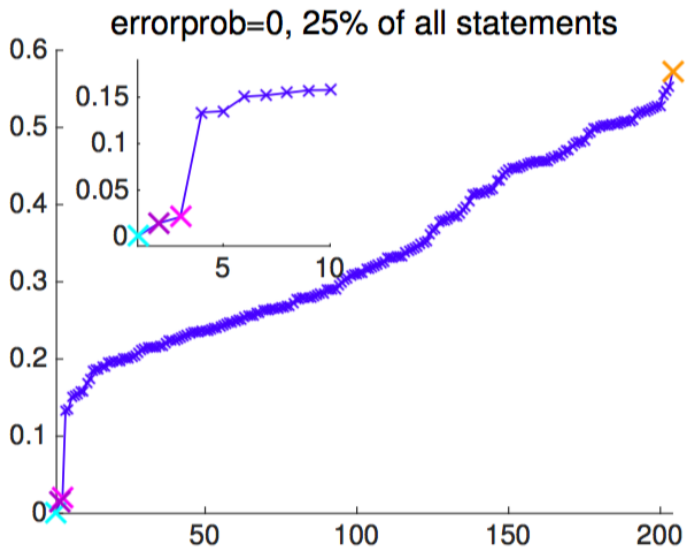


Lens Depth Function

Пример, когда это не работает

Очевидные выбросы могут быть восприняты как медоид.

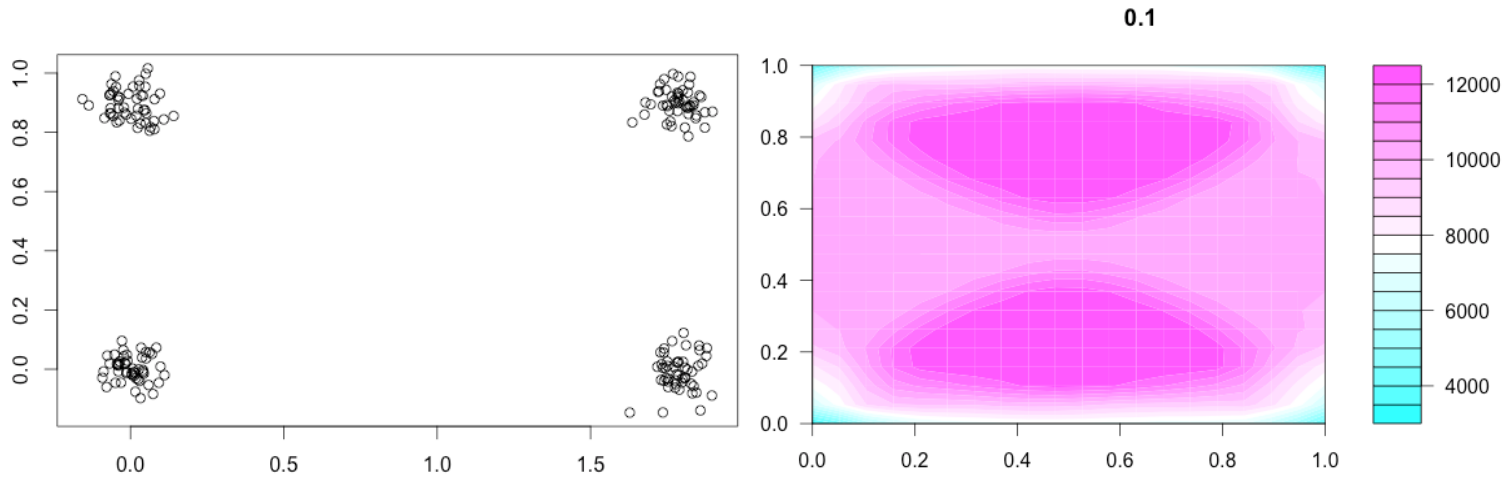




Lens Depth Function

Пример, когда это не работает

Lens Depth Function не имеет максимума в центре симметричного распределения (свойства Depth Functions).



Local Lens Depth Function

Определение

Проблема Lens Function в том, что она не учитывает локальные свойства распределения!
Вводим Local Lens Depth Function, которая учитвает не только тройки сравнений, но и расстояние между объектами:

$$LD_{local}(x; \tau, P) = Prob(x \in Lens(X, Y) \wedge d(X, Y) < \tau)$$

Local Lens Depth Function

Модель графона

Для оценки Local Lens Function мы предлагаем использовать модель графона. Для этого предположим, что мы имеем данные вида «близки ли объекты X и Y». Пусть человек отвечает на этот вопрос в соответствии со следующим алгоритмом:

1. Рассчитываем расстояние $d(X, Y)$;
2. Мэппируем это расстояние в $[0; 1]$ с помощью некоторой функции $g(.) : g(d(X, Y))$
3. Подбрасывает монетку с вероятностью орла, равной $g(X, Y)$.

По массиву данных вида «близки ли объекты X и Y» можно оценить функцию $g(.)$, что позволит учесть локальные свойства распределения исходных данных.
Заметим, что нам не нужно как-то задавать τ, так как он автоматически рассчитывается человеком.

Local Lens Depth Function

Модель графона

Модель графона соотевстует непараметрической регрессии с неизвестным дизайном:

- Мы не знаем значение расстояния $d(X, Y)$ (латентная переменная), не знаем функцию $f(.) = g(d(X, Y))$;
- Мы пытаемся восстановить фунцию $f(.)$ по наблюдениям вида «близки ли объекты X и Y»;
- Таким образом, задача оценки графона $f(.)$ сводится непараметрической регрессии с неизвестным дизайном.

При локально константном приближении (Stochastic Block Model) мы можем восстановить графон $f(.)$.