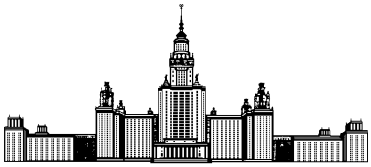
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

КУРСОВАЯ РАБОТА СТУДЕНТА 317 ГРУППЫ

"Математические и технологические проблемы построения графика в параллельных осях"

"Mathematical and technological problems of plotting parallel axes"

Выполнил:
студент 3 курса 317 группы
Тыцкий Владислав Игоревич

Научный руководитель:
к.ф-м.н., доцент
Майсурадзе Арчил Ивериевич

Москва, 2021

Аннотация

В анализе данных важной частью любого исследования является представление данных в наглядной для человека форме. Это необходимо не только для самого исследователя, но и для тех, кто читает исследование. Для представления данных низкой размерности существует множество вариантов визуализации. Однако далеко не все эти методы подходят для высокоразмерных данных. В данной курсовой работе изучается диаграмма для визуализации многомерных данных под названием "график в параллельных осях", рассмотрены его многочисленные модификации, введены новые методы увеличения читаемости, а также представлен обзор собственной библиотеки визуализации данного графика.

Содержание

1 Введение **3**

2 Модификации **4**
2.1 Добавление меток 4
2.2 Гладкие линии 4
2.3 Связывание линий 5
2.4 Иерархические графики 6

3 Проблемы построения **7**
3.1 Основные проблемы 7
3.2 Выбор порядка и направления осей 7

4 Пример использования **11**

5 Библиотека визуализации **12**
5.1 Обзор текущих прикладных средств 12
5.2 Цели и задачи библиотеки 13
5.3 Технические особенности 13
5.4 Возможности библиотеки 14

6 Заключение **14**

1 Введение

В анализе данных важной частью любого исследования является представление данных[1] в наглядной для человека форме. Это необходимо не только для самого исследователя, но и для тех, кто читает исследование.

Для представления данных низкой размерности (до 3-ей) существует множество вариантов визуализации. Однако далеко не все эти методы подходят для высокоразмерных данных. Это является фундаментальной проблемой представления информации на экране компьютера. Её обходят по-разному: через представление координат (одной и более) как параметры и рисование нескольких диаграмм для низкоразмерных данных, через рисование проекций на подпространства или через агрегирование выборки. Но среди всех способов визуализации можно выделить так называемый график в параллельных осях (parallel coordinates).

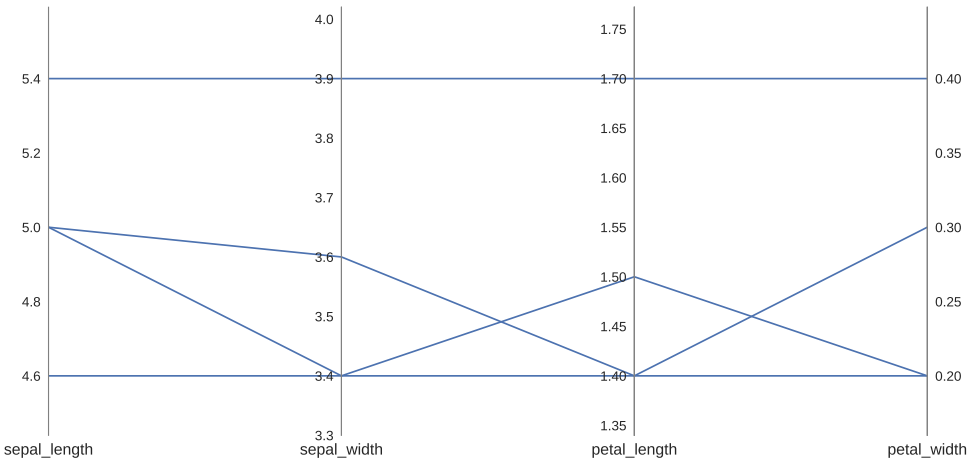


Рис. 1: Классический график в параллельных осях

График в параллельных осях — метод визуализации многомерных данных. Для отображения векторов в n-мерном пространстве рисуется n параллельных линий (осей) на одинаковом расстоянии друг от друга. У каждой оси есть направление, а также положение относительно других осей. Вектор в пространстве представляется в виде ломаной кривой, с вершинами на параллельных осях. Точка пересечения кривой с i-ой осью соответствует i-ой координате объекта. График позволяет "увидеть" не только поведение каждого отдельного объекта, но и зависимости между соседними осями.[1]

На данный момент исследователи редко используют график в параллельных осях в своих работах. Такое положение дел может быть связано с недостаточной читаемостью классических представлений графика, а также с практически полным отсутствием программных реализаций данного графика.

[1] Часто будут использоваться синонимы: выборка, датасет

2 Модификации

2.1 Добавление меток

Обычно классический график в параллельных осях используют в модифицированном виде — каждому объекту из выборки ставят в соответствие некоторую категориальную метку по какому-то правилу[2], а далее линия, соответствующая объекту, окрашивается в некоторый цвет однозначный метке объекта. Таким образом на графике можно проследить как ведут себя "похожие" объекты.

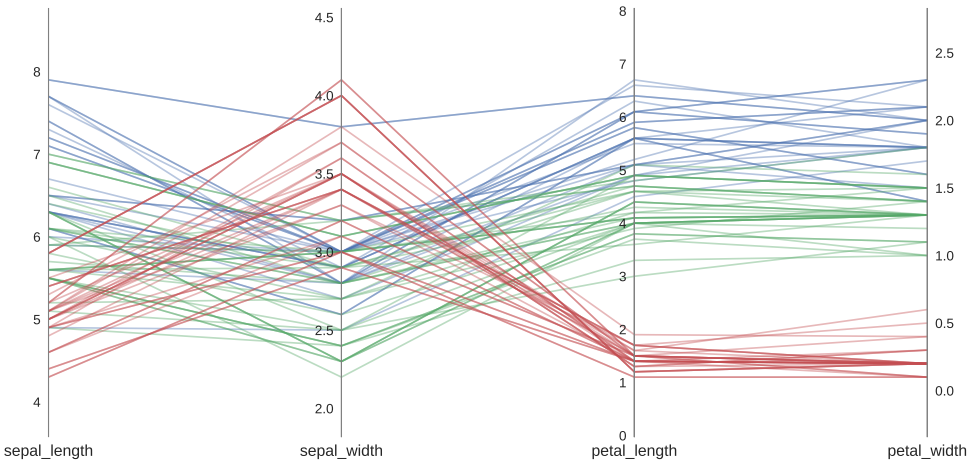


Рис. 2: График в параллельных осях с кластерами

На Рис. 2 можно наблюдать некоторые монотонные зависимости между признаками (осями) в рамках кластеров.

2.2 Гладкие линии

Нет никакой необходимости рисовать именно ломанные линии, можно рисовать гладкие кривые, которые "входят" под некоторым углом к оси (чаще всего перпендикулярно). Человеку легче воспринимать гладкие линии, поэтому читаемость графика может возрасти.[2]

[2]Правило выбирают так, чтобы объекты с одной меткой были "похожими" в некотором смысле. Иногда правило может естественным образом вытекать из самих данных, например, различные виды растений.

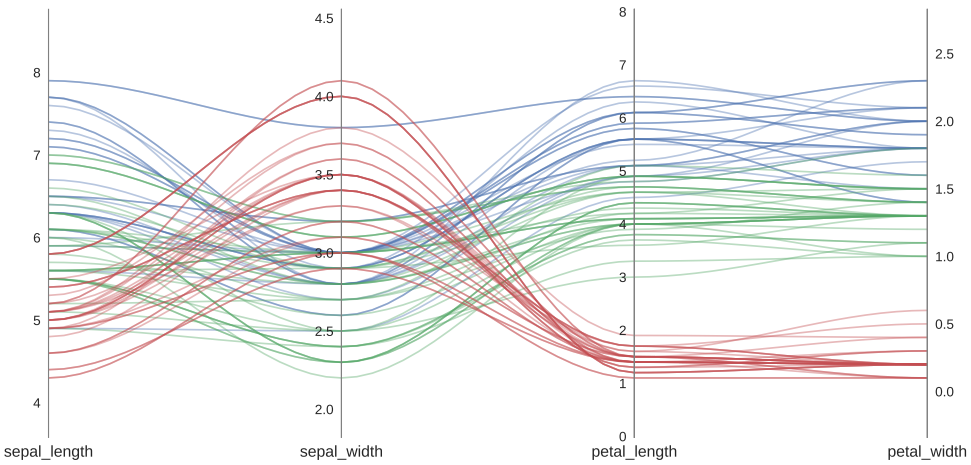


Рис. 3: График в параллельных осях с гладкими линиями

2.3 Связывание линий

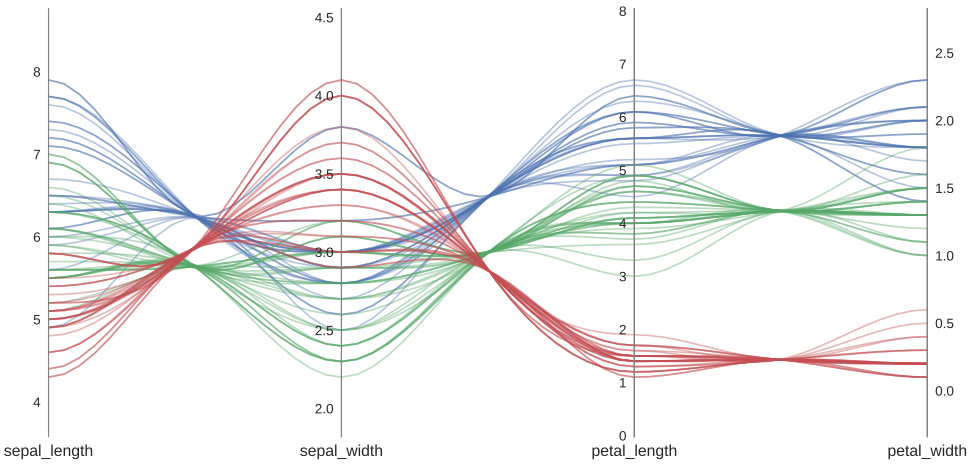


Рис. 4: График в параллельных осях со ”жгутами”

Линии с одинаковыми метками могут связываться в ”жгут” между парой осей, а далее распадаться к соответствующим точкам на оси. Степень связанности регулируема. В таких графиках теряется читаемость в рамках каждого объекта, но проще смотреть на группы объектов в совокупности.[3]

2.4 Иерархические графики

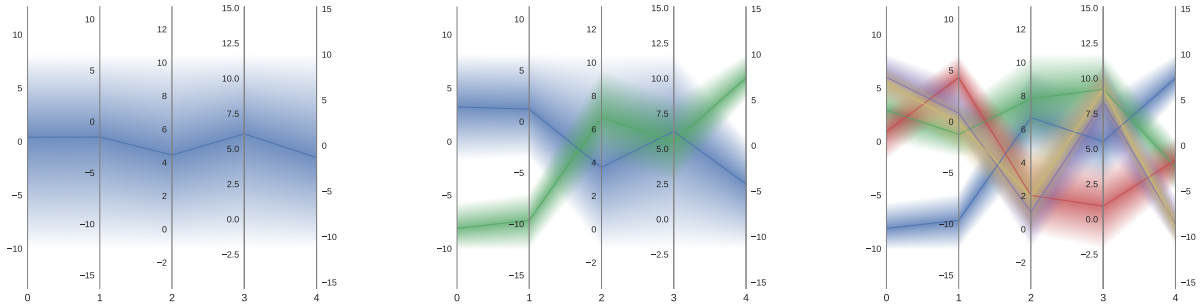


Рис. 5: Иерархические графики начиная с корня и заканчивая большим количеством кластеров

Иерархические графики в параллельных осях представляют собой метод визуализации не объектов, а иерархических кластерных структур — дендрограмм. Вместо визуализации конкретных объектов будем визуализировать сообщества похожих объектов. Чтобы визуализировать сообщества (кластеры) нужно выбрать некоторые статистики, например среднее, стандартное отклонение, максимум, минимум. Среднее нарисуем обычной линией, а стандартное отклонение отобразим полупрозрачным градиентом (Рис.5). Так график становится более читабельным, а детализация регулируется с помощью включения новых кластеров из дендрограммы.[4]

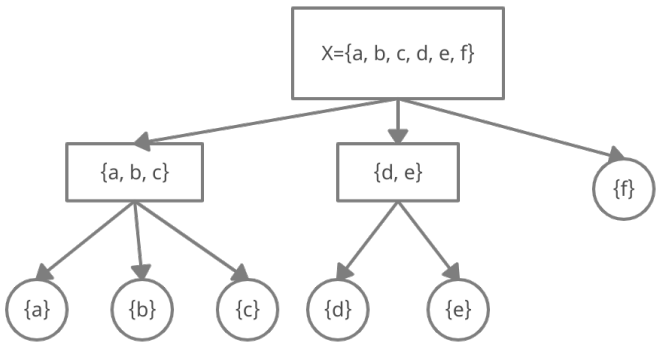


Рис. 6: Дендрограмма

Формализуем определение.

Пусть $X = (x_1, \ldots, x_n)$ – выборка, где $x_i \in \mathbb{R}^n$.
Назовем множество P m-разбиением множества X на m-подмножеств $\{P_1, \ldots, P_m\}$ такое, что:

1. $P_i \cap P_j = \varnothing, \quad \forall i, j = \overline{1, m}$
2. $\bigcup_{i=1}^{m} P_i = X$

Организуем иерархическую структуру (дендрограмму) в виде дерева, где корню соответствует X, а каждая вершина сопоставлена элементу разбиения родительской вершины (Рис.6) После построения дендрограммы необходимо выбрать выбрать какие разбиения будем отображать. Часто в качестве критерия отбора используется глубина дерева.

3 Проблемы построения

3.1 Основные проблемы

Как и любые средства визуализации график в параллельных осях обладает достоинствами и недостатками.

К достоинствами можно отнести то, что мы можем визуализировать пространства практически любой размерности. Также график обладает высокой вариативностью и простой интерпретацией, но за вариативность мы платим большим количеством гиперпараметров, которые нужно подбирать. Главный недостаток – потеря читаемости на больших и "грязных" выборках. Некоторые модификации графика частично решают эту проблему, но так мы можем потерять важную информацию о конкретных объектах. Помимо прочего у исследователя могут возникать естественные вопросы при построении:

- В каком порядке расположить оси?
- В какую сторону направлять оси?
- Как много объектов отобразить?
- Какой масштаб выбрать для каждой оси?

Обычно ответы на них ложатся на плечи самого исследователя и не всегда подбор этих гиперпараметров эффективен и объективен. Далее мы введем формальные критерии качества для ответов на первые два вопроса.

3.2 Выбор порядка и направления осей

Чтобы формализовать "правильный" порядок и направление осей необходимо понять, когда и почему человек лучше воспринимает зави-

симости на графике. Выделим основные причины:

- Линии редко пересекаются между собой.
- Монотонная зависимость между двумя соседними координатами.
- Направление осей вверх (от меньшего к большему).
- Слишком "шумные" зависимости где-нибудь на последних (справа) осях[3]

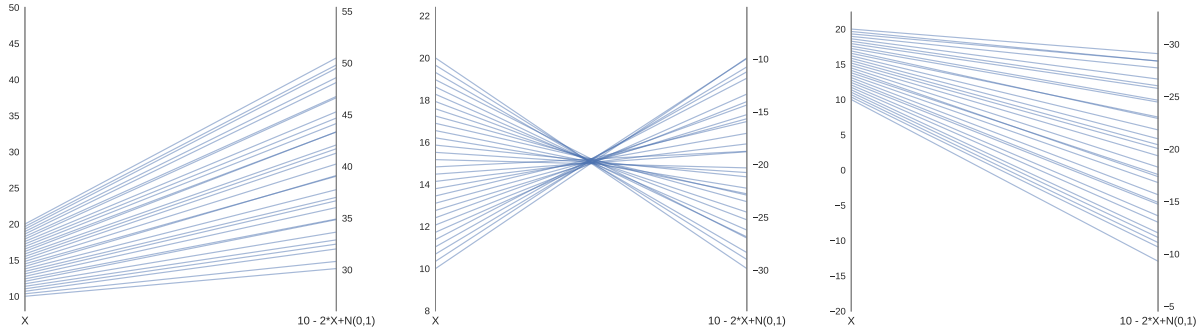


Рис. 7: Хорошо читаемая зависимость. Плохо читаемая зависимость. Предыдущий пример с перевернутой осью

Приведем некоторые примеры плохого и хорошего поведения графика. На Рис. 7 в первом случае мы видим хорошо читаемую зависимость между осями – она монотонна и возрастает. Во втором случае зависимость тоже монотонна, но убывающая. Из-за этого мы получаем большое количество пересечение линий, что сильно сказывается на читаемости т.к. сложнее проследить за каждой отдельной линией. В случае более "шумных" зависимостей ситуация может усугубляться еще сильнее. Третий случай дублирует второй, но вторая ось направлена в другую сторону, что позволяет увидеть монотонную зависимость.

Корреляция

Для введения метрики качества нам понадобится величина, характеризующая меру монотонной зависимости между объектами.

[3]Человеку привычнее воспринимать информацию слева направо.

Пусть даны две выборки $X = (x_1, \ldots, x_n), Y = (y_1, \ldots, y_n)$.
Корреляция Пирсона вычисляется по следующей формуле:

$$\rho_{XY} = \frac{\sum\limits_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum\limits_{i=1}^{n}(x_i - \overline{x})^2 \sum\limits_{i=1}^{n}(y_i - \overline{y})^2}}, \quad |\rho_{XY}| \leq 1$$

где $\overline{x}, \overline{y}$ *– выборочные средние.*

Заметим, что корреляция Пирсона не всегда хорошо проявляет при поиске монотонной зависимости между выборками.[4] Введем меру, лишенную этого недостатка:

Корреляция Спирмена вычисляется по следующей формуле:

$$r_{XY} = \frac{\sum\limits_{i=1}^{n}(R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum\limits_{i=1}^{n}(R_i - \overline{R})^2 \sum\limits_{i=1}^{n}(S_i - \overline{S})^2}}, \quad |r_{XY}| \leq 1$$

где R_i *– ранг наблюдения* x_i, S_i *– ранг наблюдения* y_i

Функционал качества

Наконец, мы готовы ввести функционал, характеризующий степень качества размещения осей.

Пусть $\pi = (\pi_1, \ldots, \pi_n)$ *перестановка множества* $\{1, \ldots, n\}$, *где* n *размерность пространства.*
Мы хотим максимизировать такой функционал:

$$\mathcal{R}(X, \pi) = \sum_{i=1}^{n-1} |r_{X^{\pi_i} X^{\pi_{i+1}}}| \to \max_{\pi}$$

где $r_{X^i X^j}$ *– Корреляция Спирмена между* i *координатой и* j *координатой.*

Заметим, что функционал может поощрять монотонно убывающие зависимости, что, вообще говоря, приведет к большому количеству пе-

[4]Корреляцию Пирсона также можно использовать в дальнейших выкладках.

ресечений линий между осями. Но нетрудно показать, что после нахождения порядка линий можно выбрать направления осей так, чтобы пересечения между линиями были минимальны.

Смысл данной формулы состоит в том, что мы хотим найти расположение осей максимизирующие сумму корреляций между соседними парами осей на графике. Так мы можем найти максимально "полезные" зависимости между признаками.

Оптимизация функционала качества

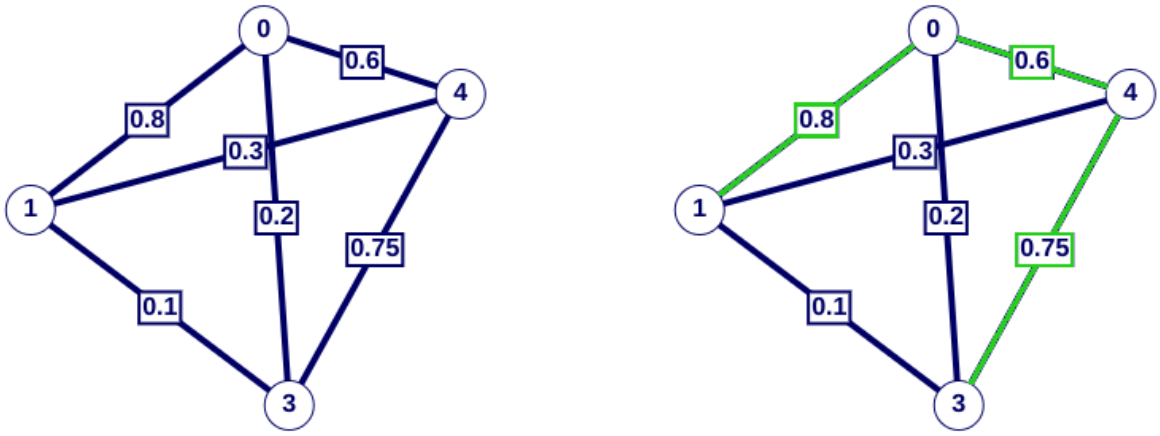


Рис. 8: Вершины соответствуют координатам, а ребра попарным корреляциям

Максимизация этого функционала тесно связна с задачей "о самом длинном пути" в теории графов.

Задача о самом длинном пути – это задача поиска простого пути максимальной длины в заданном графе. Является NP-трудной и не может быть решена за полиномиальное время для произвольных графов. Покажем связь нашей задачей:

Пусть $X = (x_1, \ldots, x_n)$ *– выборка, где* $x_i \in \mathbb{R}^n$.

Построим связный граф $G(V, E)$, *где каждая вершина* $u^i \in V$ *соответствует* i*-ой координате (*i*-ой оси на графике), а каждому ребру* $\{u^i, u^j\} \in E$ *сопоставим вес равный* $|r_{X^i X^j}|$. *(Рис. 8)*

Если в таком графе мы найдем самый длинный простой путь, то задача максимизации функционала будет решена. Простейший перебор

имеет асимптотику $O(n!)$, но помощью методов динамического программирования можно ее можно улучшить. Заметим, что график в параллельных осях стоит использовать при $n < 15$, иначе график перестанет быть читаемым.

4 Пример использования

Для примера был выбран датасет "The Boston Housing Dataset". Предварительно убраны категориальные признаки, а также проведена кластеризация методом K-means для большей наглядности.

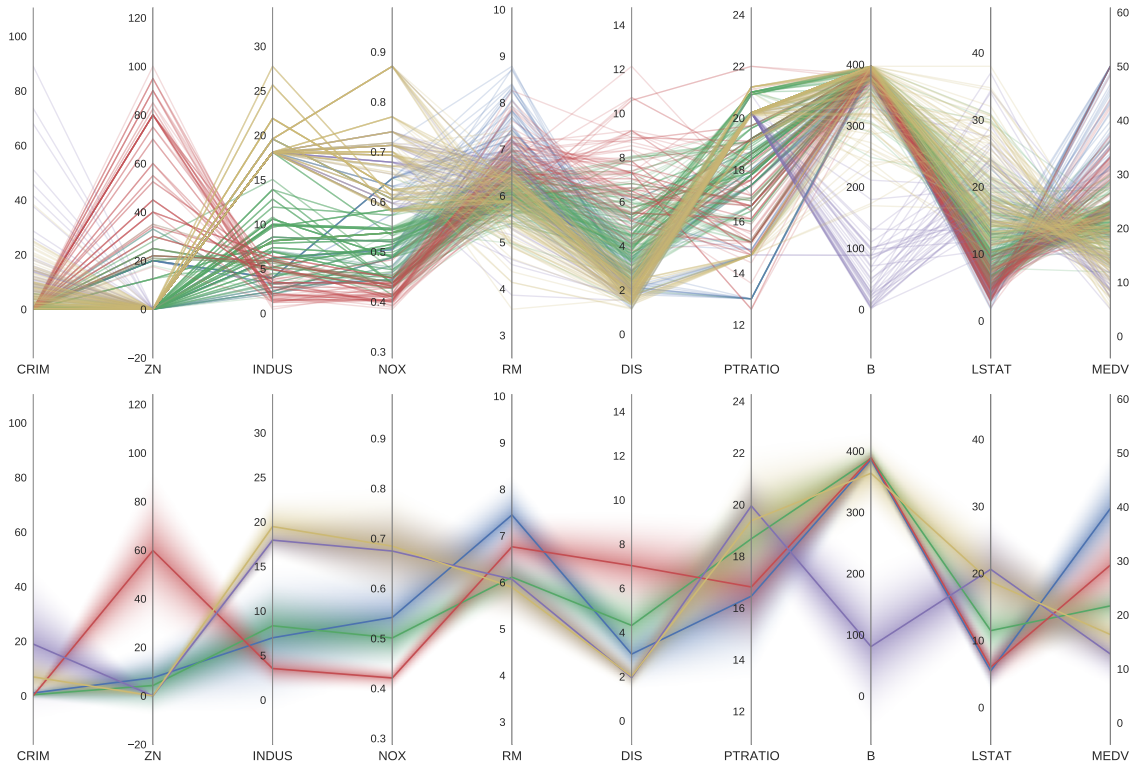


Рис. 9: Графики в параллельных осях без выбора порядка

На Рис.9 представлены графики без выбора порядка признаков. График получается довольно "шумным" – много пересечений линий, сложно заметить зависимости между признаками.

На Рис. 10 проиллюстрированы графики после нахождения оптимального порядка осей. После оптимизации нам проще разглядеть структуру кластеров не только в совокупности, но и между кластерами. Для примера рассмотрим ось NOX(количество азота в воздухе) и DIS(сумма расстояний от центров занятости). До оптимизации мы не могли наблю-

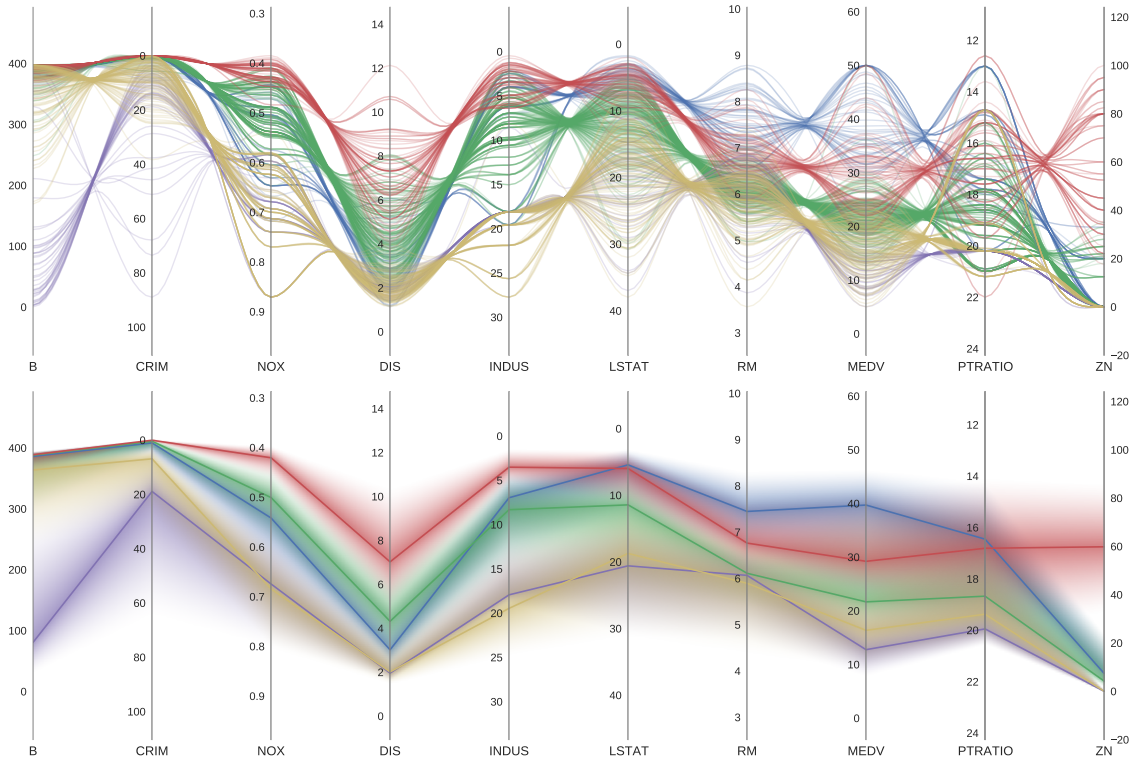


Рис. 10: Графики в параллельных осях с выбором порядка

дать монотонно убывающую зависимость (чем больше NOX, тем меньше DIS), но после она четко видна. Таким образом нахождение оптимального порядка с точки зрения введенного функционала максимизирует число интерпретируемых зависимостей между осями. Но нужно заметить, что нам не гарантируется, что мы обнаружим все "хорошие" зависимости так как каждый признак может "взаимодействовать" максимум с двумя другими признаками на графике.

5 Библиотека визуализации

5.1 Обзор текущих прикладных средств

Несмотря на то, что написано большое количество работ о графиках в параллельных осях, существует лишь несколько программ, общедоступных для работы с ними. Например: ELKI, GGobi, Mondrian, Orange и ROOT. Отдельно выделяется D3.Parcoords.js – мощная библиотека на языке JavaScript, посвященная только графикам в параллельных осях. В python в библиотеке pandas есть лишь его базовая версия. В других

же популярных python библиотеках нет даже этого.

Удивительно, что такое мощное средство визуализации обходят стороной разработчики библиотек. Возникает желание создать собственный продукт со всеми возможными подходами к рисованию графика в параллельных осях.

5.2 Цели и задачи библиотеки

В первую очередь необходимо заметить, что библиотека пишется на базе matplotlib. Это мощная низкоуровневая библиотека, умеющая рисовать всевозможные статические диаграммы. Статичность диаграммы можно считать как недостатком, так и достоинством[5]. С одной стороны интерактивность в случае графиков в параллельных осях существенно ускоряет построение эстетичного графика, но с другой это может излишне перегружать и усложнять взаимодействие пользователя с библиотекой, а также существенно уменьшить спектр возможностей. Библиотека на базе matplotilb позволит пользователю не только тончайшим образом настраивать вид графика, но и быстро получить красивый и информативный график "из коробки". [6]

5.3 Технические особенности

- Простой высокоуровневый интерфейс. Как и в библиотеке seaborn методы могут принимать pandas.DataFrame, обычные numpy массивы или списки – для всего единый интерфейс.[7]

- Возможность сохранения графика в любой формат, поддерживаемый matplotlib

- Нативно встраивается в любые программные оболочки, поддерживающие matplotlib.

- Возможность кастомизации уже готового графика.

[5] В сфере визуализации монополия на использование интерактивных графиков отдана JavaScript и его библиотекам. Статичные графики обычно рисуют с помощью matplotilb или библиотек, созданных на его базе.

[6] Примером может служить библиотека seaborn, написанная на базе matplotlib, но использующая высокоуровневые функции, позволяющие избегать утомляющей настройки.

[7] Пока что в качестве входных данных доступен только pandas.DataFrame

5.4 Возможности библиотеки

- Построение классических графиков в параллельных осях
  - Возможность рисовать гладкие линии.
  - Возможность "связывания" линий кластеров.
- Построение иерархических графиков
  - Отрисовка полупрозрачного градиента.
  - Работа с иерархическими кластерами(дендрограммами).
- Дополнительно *(пока не реализовано)*
  - выделение подмножества линий в диапазоне значений одной из осей.
  - нахождение оптимального расположения осей.
  - создание иерархических кластеров на основе входящей выборки.

6 Заключение

В работе были рассмотрены основные виды графиков в параллельных осях, их достоинства и недостатки, предложен метод повышения читаемости и информативности за счет выбора порядка и направления. Эксперименты показали, что дискретная оптимизация введенного функционала качества улучшает восприятие графика и позволяет найти нетривиальные зависимости в данных. Удалось связать оптимизацию данного функционала с NP-полной задачей *"О максимальном пути"* в полном графе с неотрицательными ребрами. Улучшение асимптотики решения данной задачи открытая проблема. Вообще говоря, данный функционал не единственный вариант оптимизации порядка осей и это требует дальнейших исследований. Также была разработана open-source библиотека на языке Python для построения данных графиков.

Список литературы

[1] Inselberg A., Dimsdale B. Parallel coordinates: a tool for visualizing multi-dimensional geometry //Proceedings of the First IEEE Conference on Visualization: Visualization90. – IEEE, 1990. – С. 361-378.

[2] Heinrich J., Weiskopf D. State of the Art of Parallel Coordinates //Eurographics (STARs). – 2013. – С. 95-116.

[3] Heinrich J. et al. Evaluation of a bundling technique for parallel coordinates //arXiv preprint arXiv:1109.6073. – 2011.

[4] Fua Y. H., Ward M. O., Rundensteiner E. A. Hierarchical parallel coordinates for exploration of large datasets. – IEEE, 1999. – С. 43-508.

[5] Wegman E. J. Hyperdimensional data analysis using parallel coordinates //Journal of the American Statistical Association. – 1990. – Т. 85. – №. 411. – С. 664-675.

[6] Inselberg A. (2009) Parallel Coordinates. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA