Отчет о проделанной работе в семестре

Тыцкий Владислав

Декабрь 2020 г.

Введение

В анализе данных важной частью любого исследования является представление данных [1] в наглядной форме для человека. Это необходимо как для самого исследователя, так и для тех, кто будет читать исследование.

Когда дело касается представления низкоразмерных данных(до 3-ей размерности), возможностей для визуализации изобретено довольно много. В случае данных высокой размерности многие методы для низкоразмерных данных не работают. Это фундаментальная проблема представления информации на экране компьютера и устройства человеческого зрения(а быть может и устройства мира??). Эту проблему обходят по-разному: одну или несколько координат можно воспринимать как параметр и рисовать несколько диаграмм для низкоразмерных данных, можно рисовать проекции на подпространства, можно агрегировать выборку. Но среди всех способов визуализации можно выделить так называемый график в параллельных осях(parallel coordinates).

График в параллельных осях

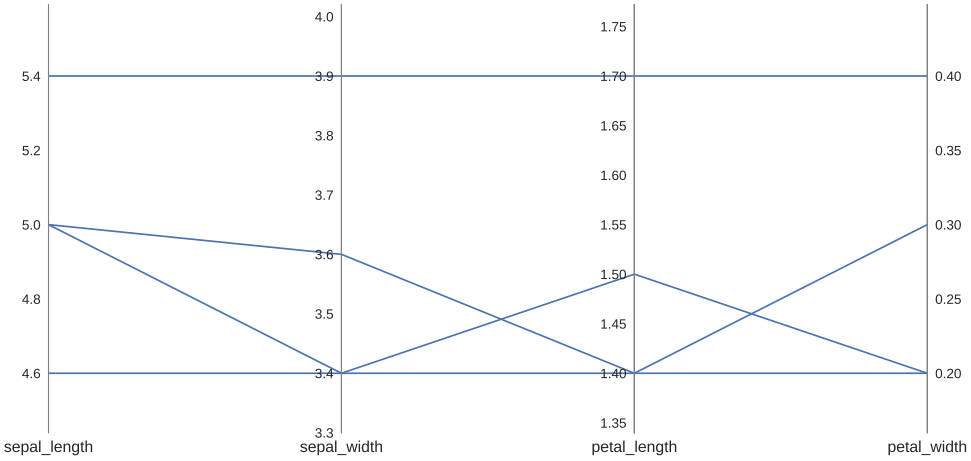

Рис. 1: Классический график в параллельных осях

График в параллельных осях — метод визуализации многомерных данных. Для отображения векторов в n-ом пространстве рисуется n параллельных линий(осей) на равном расстоянии друг от друга. Вектор представляется в виде ломаной кривой, с вершинами на параллельных осях. Точка пересечения линии с i-ой осью соответствует i-ой координате объекта.

[1] Часто будут использоваться синонимы выборка, датасет

Возникают естественные вопросы:

- В каком порядке расположить оси?
- В какую сторону направлять ось?
- Какой масштаб выбрать для каждой оси?

Модификации

1. Добавление кластеров

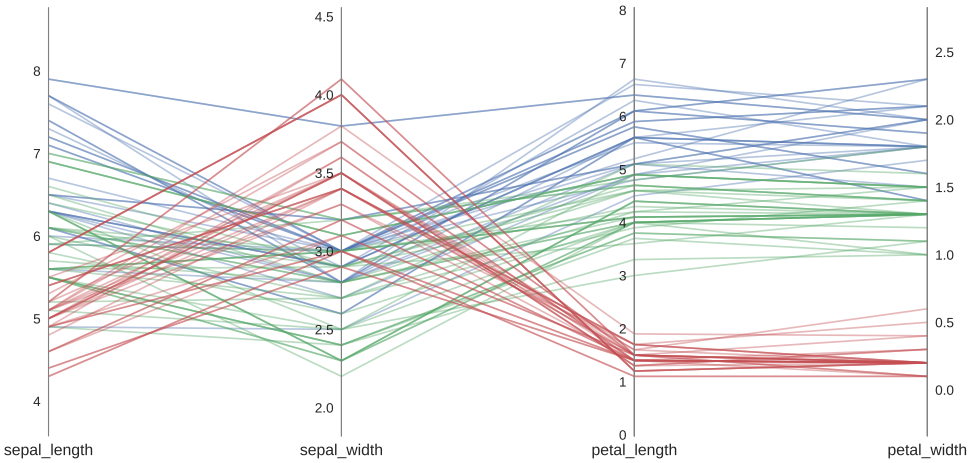

Рис. 2: График в параллельных осях с кластерами

К сожалению, классический график в параллельных осях становится практически нечитаемым при увеличении количества объектов и осей, поэтому чаще всего его применяют в немного 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.

Но многие проблемы от этого не исчезли(и даже добавились):

- По прежнему теряется читаемость при увеличении количества объектов и признаков.
- Визуально человеку сложнее воспринимать ломанные линии.
- Какое правило выбрать для разметки объектов?

[2] Обычно правило выбирают так, чтобы объекты с одной меткой были "похожими" в некотором смысле – это называют кластеризацией

2. Гладкость линий

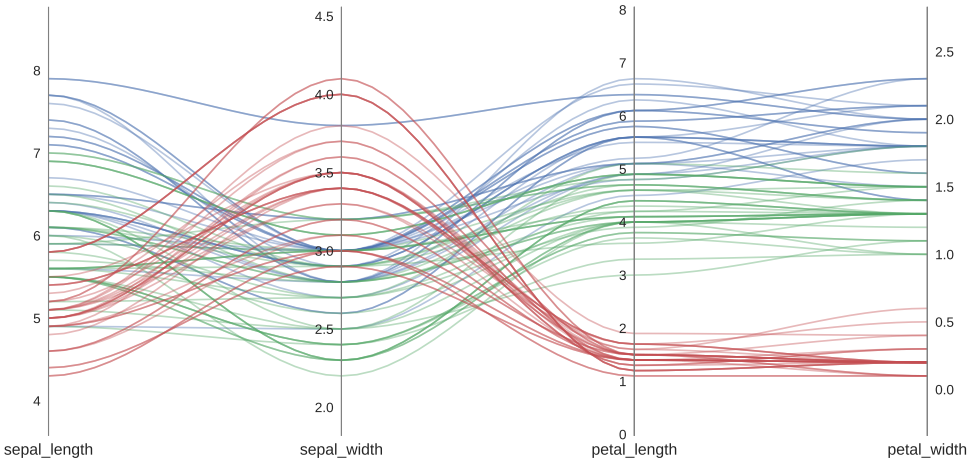

Рис. 3: График в параллельных осях с гладкими линиями

Нет никакой необходимости рисовать именно ломанные линии, поэтому можно рисовать гладкие кривые, которые ”входят” перпендикулярно оси.

3. Связывание линий

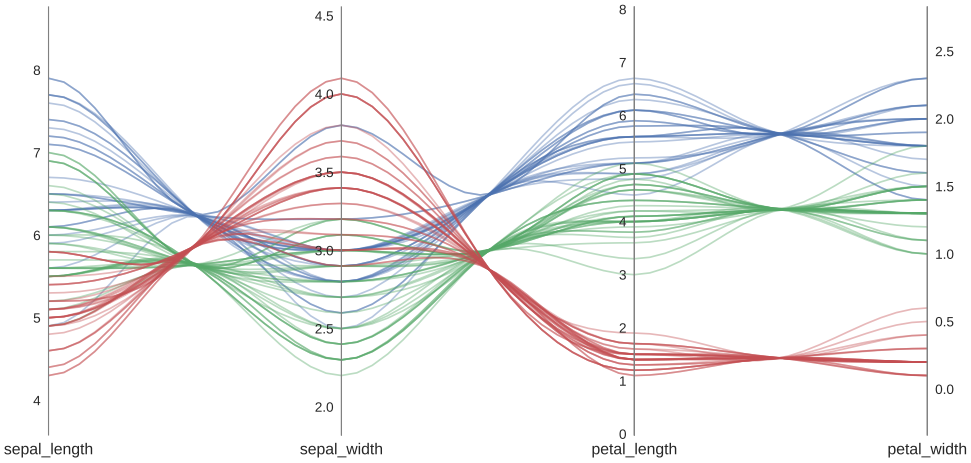

Рис. 4: График в параллельных осях со ”жгутами”

Линии с одинаковыми метками могут связываться в ”жгут” между парой осей, а далее распадаться к соответствующим точкам на оси. Степень связанности можно регулировать. Эту идею можно обобщить – пусть линии связываются не потому что принадлежат одному классу, а потому что имеют близкие значения координат i и

i+1 при построении. Потенциально это избавляет от необходимости кластеризовать объекты перед построением.

4. Иерархические графики в параллельных осях

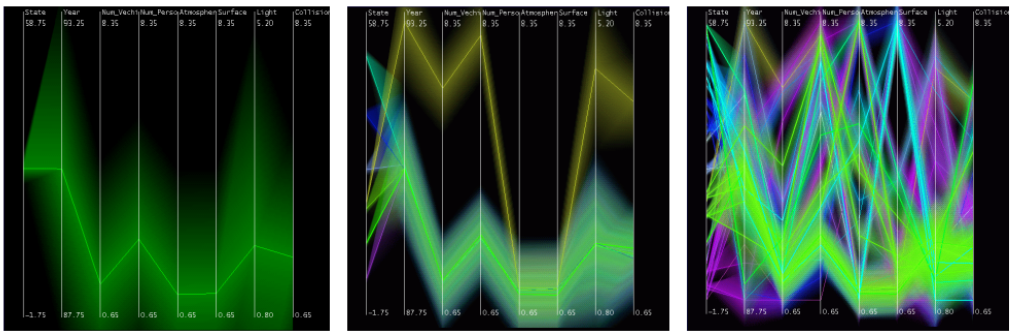

Рис. 5: Иерархические графики начиная с корня и заканчивая большим количеством кластеров

Иерархические графики в параллельных осях представляют собой метод визуализации не объектов, а некоторых иерархических кластерных структур — дендрограмм. Вместо визуализации конкретных объектов будем визуализировать сообщества похожих объектов. Чтобы визуализировать сообщества(кластеры) нужно выбрать некоторые статистики, например среднее и стандартное отклонение. 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.

Библиотека визуализации

Обзор текущих прикладных средств

В то время как написано большое количество работ о графиках в параллельных осях, существует лишь несколько заметных 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 базовая версия графика. В других же популярных библиотеках визуализации нет реализации графика в параллельных осях.

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

Цели и задачи библиотеки

В первую очередь необходимо заметить, что библиотека написана(пишется) с на базе matplotlib. Это очень мощная низкоуровневая библиотека, умеющая рисовать всевозможные статические диаграммы. Статичность диаграммы можно считать как недостатком, так и достоинством. [3] 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 спектр возможностей существенно уменьшается. Библиотека на базе matplotilb позволит пользователю не только тончайшим образом настраивать вид графика, но и быстро получить красивый и информативный график "из коробки". [4]

Возможности библиотеки:

- Построение классических графиков в параллельных осях
  - Возможность рисовать гладкие линии. Должен быть непрерывный параметр, задаюший вид кривой.
  - Возможность "связывания" линий кластеров. Должен быть непрерывный параметр задающий степень связывания.
  - Возможность "связывания" линий на основе близости. Также должен быть непрерывный параметр задающий степень связывания.
- Построение иерархических графиков
  - Отрисовка полупрозрачного градиента.
  - Работа с иерархическими кластерами(дендрограммами).
  - Изображение распределения с помощью градиента.(boxplot, histogram).
- Дополнительно
  - выделение подмножества линий в диапазоне значений одной из осей
  - нахождение оптимального расположения осей
  - создание иерархических кластеров на основе входящей выборки

[3] 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.

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

Технические особенности:

- Простой высокоуровневый интерфейс. Как и в библиотеке seaborn методы могут принимать pandas.DataFrame, обычные numpy массивы или списки – для всего единый интерфейс.
- Эстетичные графики "из коробки".

Итоги(после первого семестра)

По итогам семестра удалось реализовать большую часть возможностей библиотеки, касающихся классического графика в параллельных осях

- Возможность рисовать гладкие линии. **Пока что не добавлен параметр задающий вид кривой**.
- Возможность "связывания" линий кластеров. Добавлен непрерывный параметр задающий степень связывания.
- **Возможность связывания линий на основе близости не реализована**

Интерфейс для пользователя практически полностью повторяет реализацию seaborn. Пока что в качестве параметра можно использовать только DataFrame, но уже сейчас можно получать красивые графики "из коробки". [5]

[5] Большинство графиков в отчете нарисованы с помощью данной библиотеки.