**1. КЛАСТЕРИЗАЦИЯ ФИНАНСОВЫХ АКТИВОВ КАК ВРЕМЕННЫХ РЯДОВ**

**1.1. Классическая постановка задачи кластеризации**

Прежде, чем рассматривать задачу кластеризации временных рядов, рассмотрим классическую задачу кластеризации в общем виде.

Кластеризация – задача объединения множества объектов в однородные группы (их называют кластера) так, чтобы объекты по своим признакам из одной группы были более похожи друг на друга, чем на объекты из других кластеров.

Задача кластеризации – относится к ряду классических задач машинного обучения без учителя. Обучение без учителя – такой способ обучения, при котором у каждого объекта нет метки класса, и требуется обнаружить в множестве объектов внутренние зависимости, взаимосвязи и зависимости.

Приведем математическую постановку задачи кластеризации. Дано множество – множество объектов обучающей выборки, – -ый объект обучающей выборки, представляющий собой вектор , где – количество признаков в обучающей выборки, дано множество меток кластеров . Для любой пары объектов обучающей выборки задан критерий схожести (функция расстояния): .

Необходимо разбить обучающую выборку на кластеры таким образом, чтобы объекты внутри одного кластера были близки относительно функции расстояния .

Существует много различных мер близости объектов внутри одного кластера, и, как правило, для каждого алгоритма кластеризации, существует своя классическая мера близости, обозначим такую меру близости , где – вектор меток кластеров для каждого объекта обучающей выборки.

Более формально, необходимо найти такую функцию (алгоритм кластеризации) , которая определяет взаимно-однозначное соответствие между множеством меток кластеров и множеством объектов обучающей выборки, и, которая минимизирует меру близости объектов внутри одного кластера .

Очевидно, что не существует оптимального решения задачи кластеризации, поскольку, во-первых, результат решения задачи кластеризации зависит от функции расстояния между объектами , во-вторых, результат решения задачи кластеризации зависит от количества кластеров , в-третьих, нет объективных критериев качества кластеризации, как это, например, есть в задачах обучения с учителем.

**1.2. Постановка и особенности задачи кластеризации финансовых активов временных рядов**

Кластеризация временных рядов решает часть таких практических задач, как:

1. Нахождение финансовых активов, обладающих схожей динамикой;
2. Объединение пациентов в группы по результатам их кардиограммы;
3. Поиск схожих аудиозаписей на основе звуковой спектрограммы.

В рамках данной выпускной квалификационной работы будет рассмотрена первая практическая задача – выделение финансовых активов, обладающих схожей динамикой. Выделение групп финансовых активов со схожей динамикой позволяет, во-первых, оценить степень диверсификации портфеля, диверсифицировать инвестиционный портфель, а, во-вторых, определять динамику финансовых активов, которые с некоторым лагом зависят от других финансовых активов – другими словами, это позволяет понять, вырастет или упадет стоимость финансового актива, исходя из того, вырос или упал актив из его кластера в последнюю единицу времени.

Как и в кластерном анализе объектов, качество результатов кластеризации временных рядов определяется алгоритмом кластеризации и функцией расстояния двух временных рядов.

Временные ряды задаются векторами , такая многоуровневая индексация объясняется тем, что временные ряды не обязательно имеют одинаковое количество наблюдений.

Что касается алгоритмов кластеризации – все алгоритмы имеют место быть, все зависит от специфики конкретной задачи. Однако, с расстоянием двух временных рядов не все так однозначно.

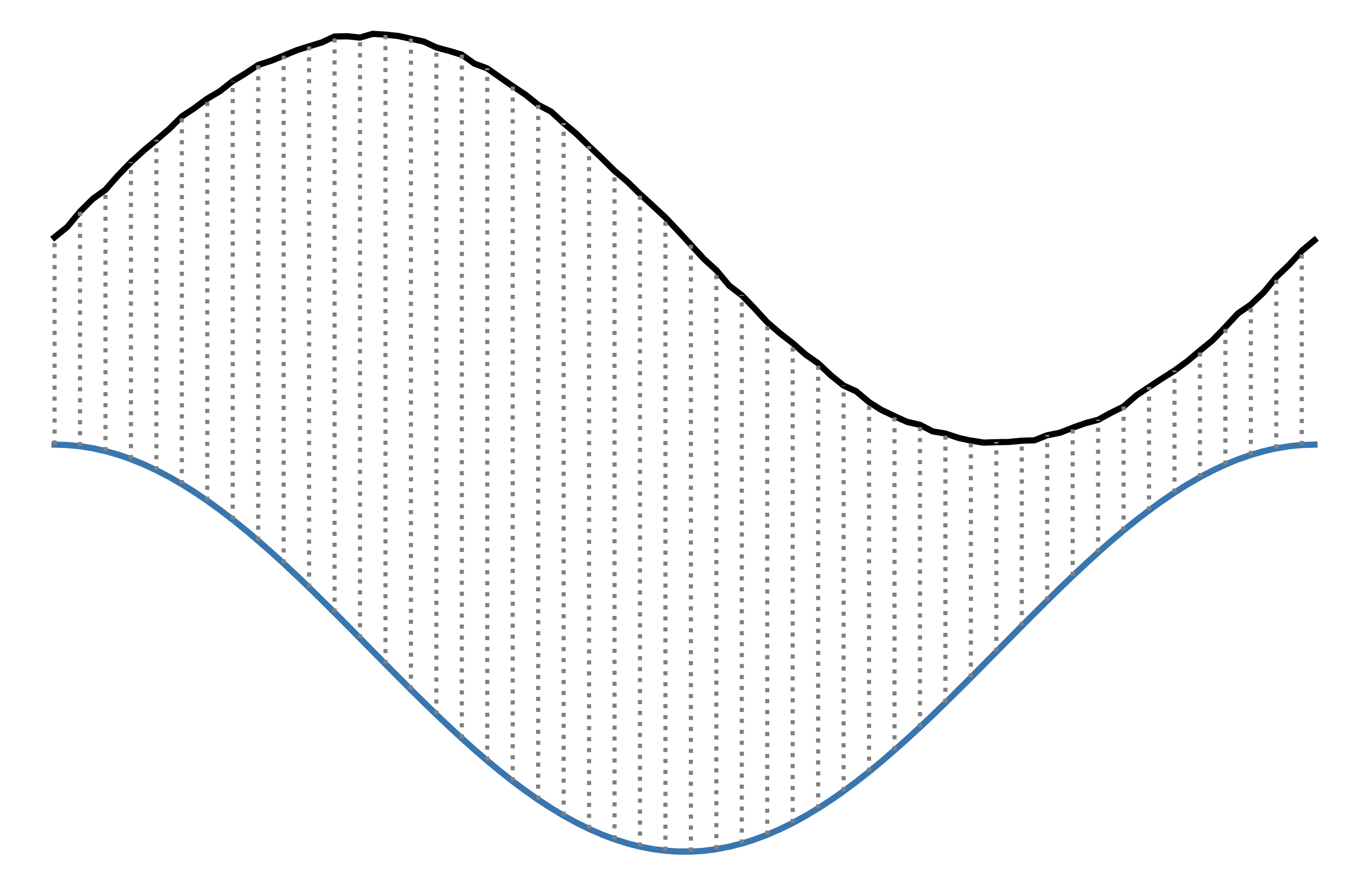
В случае, если временные ряды и имеют одинаковое количество наблюдений, то для них можно вычислить такие классические расстояния, как:

1. Евклидово расстояние
2. Манхэттенское расстояние
3. Расстояние Чебышева

В случае, если временные ряды и имеют различное количество наблюдений, то можно предпринять некоторые меры, чтобы привести ряды к одинаковому количеству наблюдений, например:

1. Обрезать слева либо справа временной ряд, в котором больше наблюдений – это называется каттинг (англ. cutting)
2. Заполнить слева либо справа временной ряд, в котором меньше наблюдений, нулями – это называется паддинг (англ. padding)

После этого, для полученных модификаций временных рядов можно вычислить рассмотренные выше классические расстояния. Фактически, такой подход вычисления расстояний между временными рядами можно визуализировать образом, показанным на рисунке ниже.



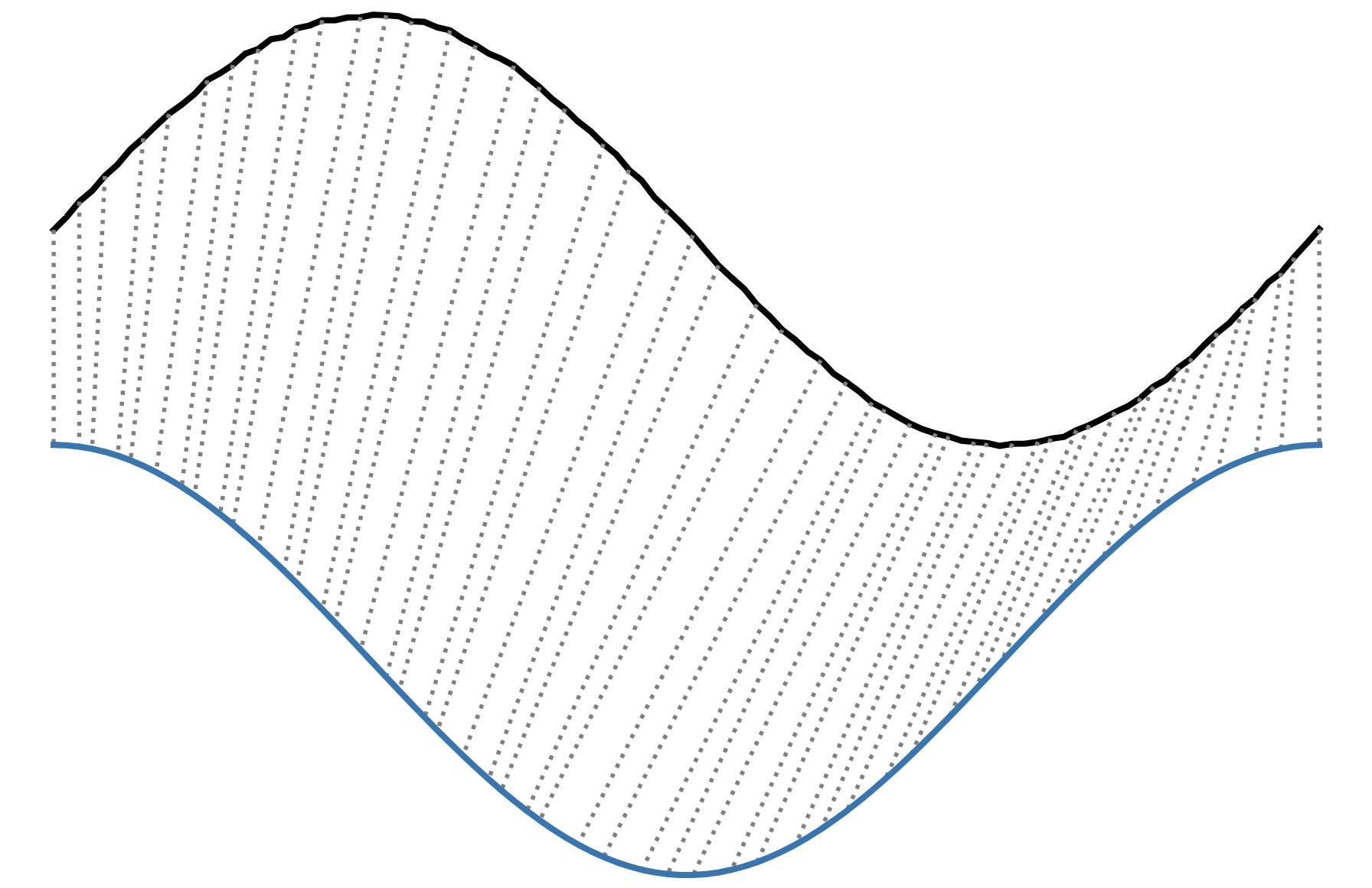
Однако, к несчастью, такой подход не совсем в полной мере отражает фактическую дистанцию, фактическое расстояние между временными рядами. Для примера, рассмотрим конкретный временной ряд и его копию, которая сдвинута по оси абсцисс влево – в таком случае, рассчитав любое классическое расстояние между рядами расстояние будет существенно отличаться от нуля несмотря на то, что временные ряды, фактически, идентичные.

Для решения данной проблемы существуют специальные меры расстояния между временными рядами, однако принято выделять, как правило, одно конкретное расстояние – DTW-расстояние, которое вычисляется по алгоритму динамической трансформации временной шкалы.

**1.3. Алгоритм динамической трансформации временной шкалы**

DTW-расстояние впервые было применено в задачи распознавания речи – для двух звуковых дорожек необходимо было определить, произнесены ли на них одинаковые смысловые фразы, учитывая разную скорость произнесения одинаковых фраз разными людьми.

Смысл алгоритма динамической трансформации временной шкалы, исходя из названия, состоит в том, что алгоритм трансформирует временной ряд (то есть, сжимает либо растягивает) для того, чтобы достичь оптимального сопоставления элементов двух последовательностей. Визуализация DTW-расстояния представлена на рисунке ниже.

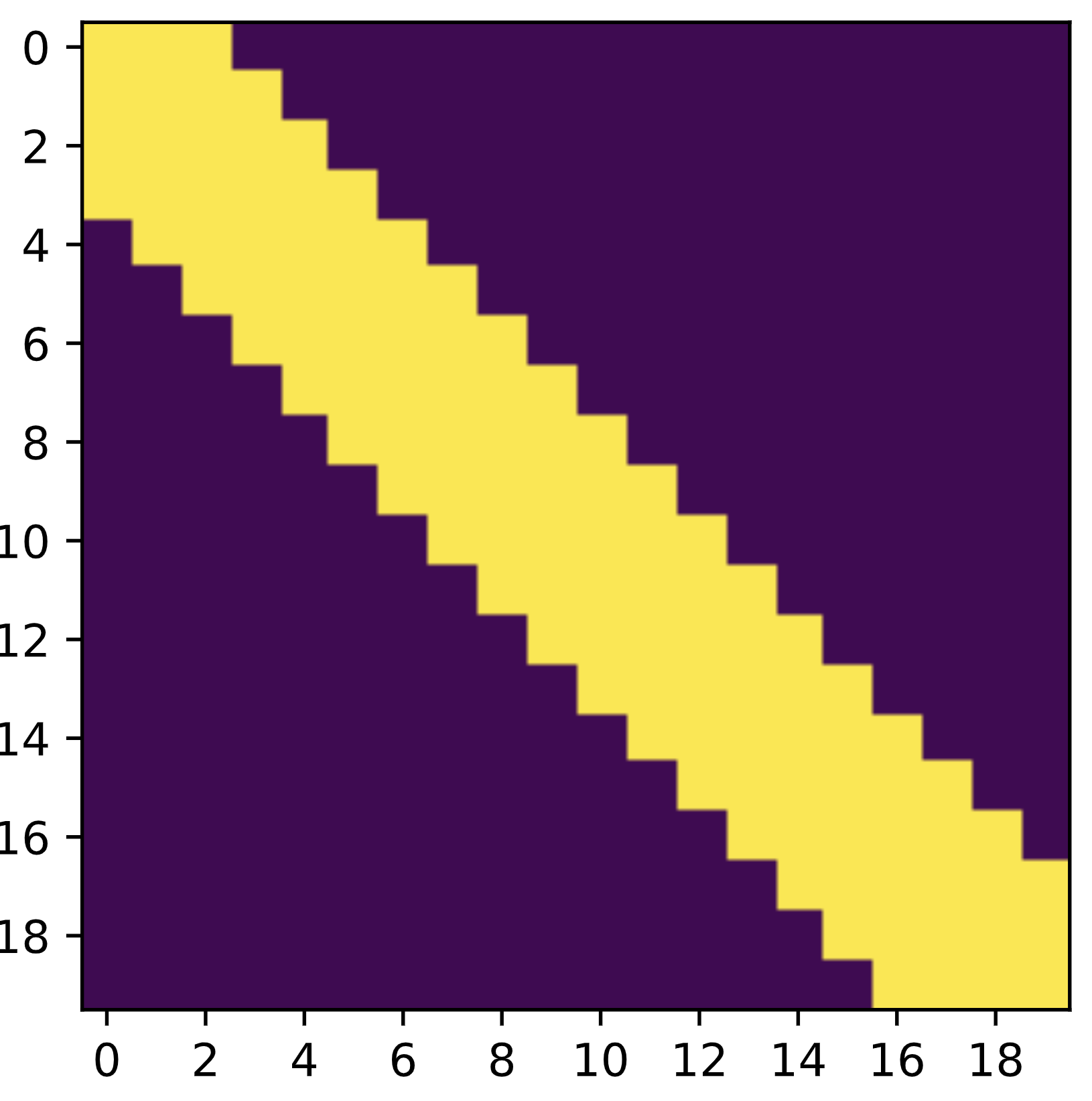


Рассмотрим математическую постановку DTW-алгоритма: даны два временных ряда и . Далее, считается прямоугольная матрица локальных потерь порядка , элемент которой соответствует расстоянию между -ым элементом временного ряда и -ым элементом временного ряда , в качестве такого расстояния, как правило, берется либо , либо . После чего находится такой путь в матрице из левого верхнего угла в правый нижний, который, во-первых, минимизирует расстояние между рядами и . Тогда DTW-расстояние вычисляется по следующей формуле:

Однако, количество всевозможных путей в матрице экспоненциально возрастает с ростом входных данных. Поэтому, чтобы гарантировать нахождение пути за полиномиальное время (в нашем случае за ), вводят локальное ограничение на соседние пары индексов в пути :

Фактически, это ограничение означает, что путь должен быть монотонным, другими словами, в рамках пути по матрице можно двигаться либо вправо, либо вниз, либо вправо вниз по диагонали.

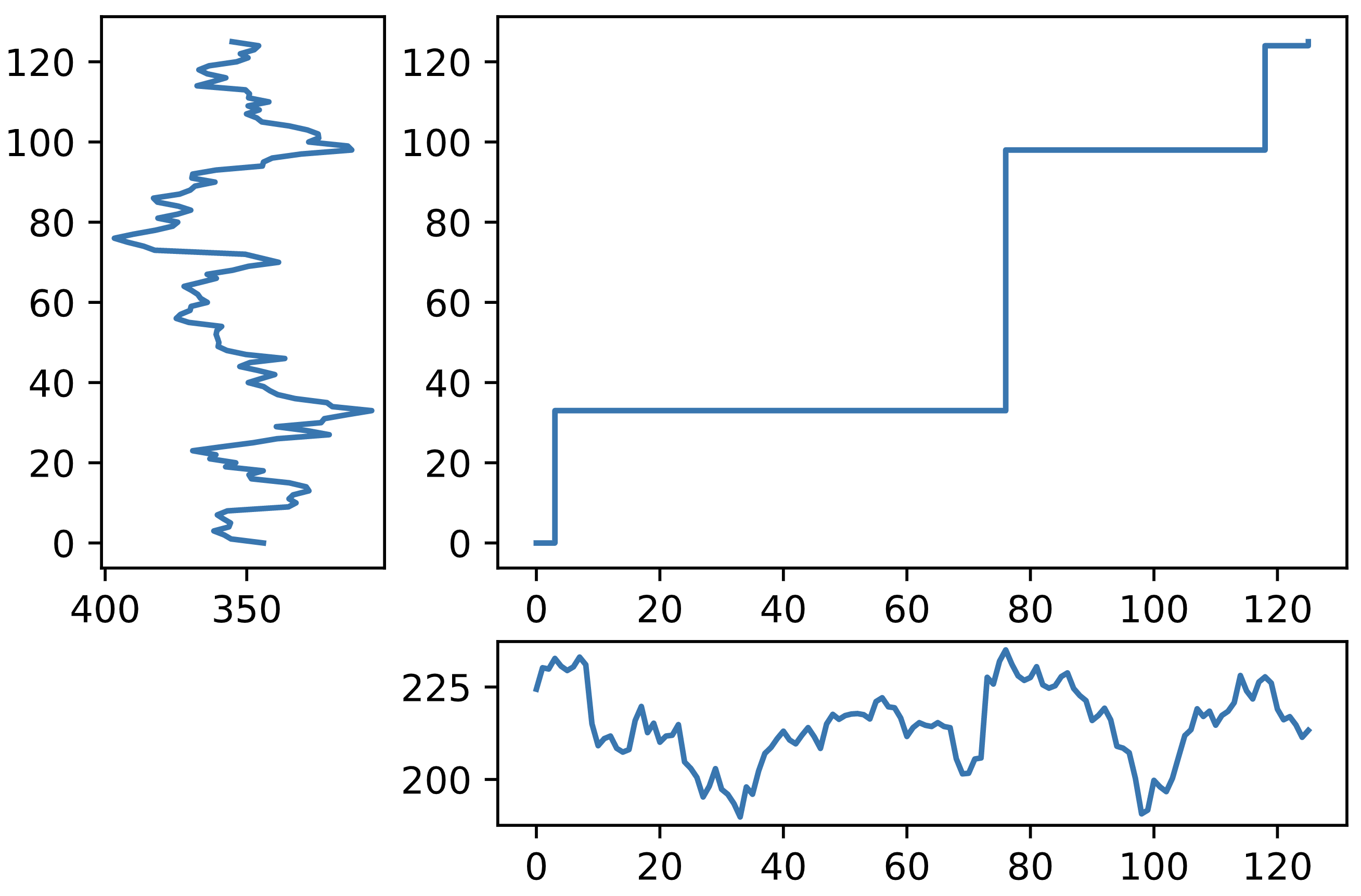
Стоит добавить, что иногда, помимо локального ограничения на соседние пары индексов, также вводятся глобальные ограничения, например, ограничение на ширину окна пути по матрице локальных потерь, в пределах которого разрешается прокладывать путь. Существуют различные ограничения на ширину окна, однако, классическим считается ограничение Сакэ-Чиба (англ. Sakoe-Chiba) – это ограничение на движение в рамках симметричной области вдоль главной диагонали матрицы локальных потерь. Пример такого глобального ограничения Саке-Чиба представлен на рисунке ниже.



Таким образом, нахождение DTW-расстояния сводится к задаче динамического программирования, с состоянием динамики, которое выражается в следующей формуле:

Где – матрица состояний динамики, элемент равен следующему выражению:

Существует классический вариант визуализации пути в матрице локальных потерь, который представлен на рисунке ниже. На рисунке ниже представлена визуализация результата DTW-алгоритма для двух финансовых активов компаний Visa и MasterCard.

**

**1.3. Кластеризация финансовых активов по описательным статистикам**

В предыдущем подразделе был рассмотрен подход кластеризации финансовых активов в признаковом пространстве, сформированном на значениях стоимости финансового актива в единицу времени, то есть, на значениях временного ряда.

Однако, есть альтернативные подходы к формированию признакового пространства финансовых активов. Один из таких альтернативных подходов – формирование признаков на основе описательных статистик временных рядов.

Идея состоит в следующем: если временные ряды имеют схожую динамику, то и их описательные статистики временных рядов не должны значительно различаться.

Тогда, способ кластеризации финансовых активов по описательным активам, состоит в том, чтобы для каждого ряда посчитать большое количество описательных признаков, а затем в полученном признаковом пространстве кластеризовать финансовые активы.

Как и для обычных признаков, можно считать такие стандартные описательные статистики, как среднее, медиана, дисперсия, стандартное отклонение, выборочные квантили. Однако, поскольку работа идет конкретно с временными рядами, то гарантируется, что наблюдения упорядочены во времени, тогда можно считать и статистики временных рядов – автокорреляция различных порядков, стационарность, сезонные компоненты, а также, обычные описательные статистики в окне – например, среднедневное значение, среднемесячное значение и другие всевозможные варианты.

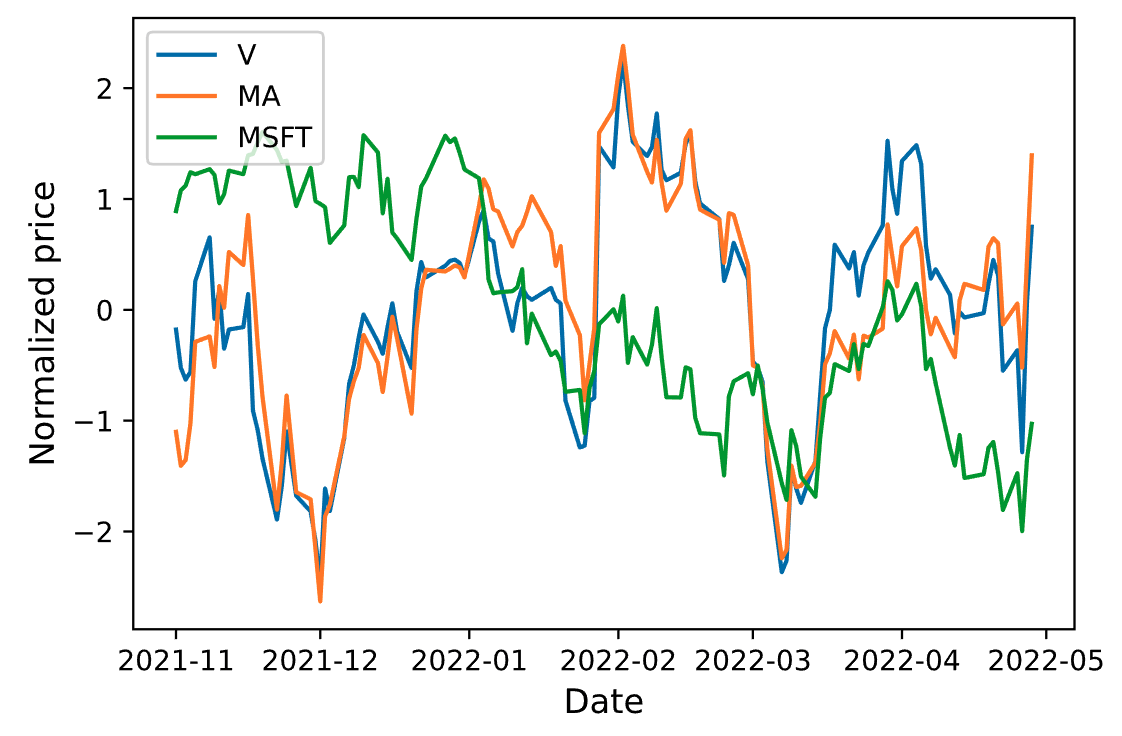
Стоит отметить, что в данном случае, учитывание в признаковом пространстве автокорреляции решает проблему, зависимости пары временных рядов с некоторым лагом, которая в предыдущих подпунктах решалась с использованием алгоритма динамической трансформации временной шкалы.

В качестве доказательства схожести финансовых активов, у

***ТУТ ДОПИСАТЬ РАЗУМНОЕ ЧТОТО***

Изображение выглядит как стол

Автоматически созданное описание



**1.4. Кластеризация финансовых активов по скрытому векторному представлению**

Другой, более современный подход к формированию признакового пространства – обучение искусственных нейросетей, имеющих архитектуру автокодировщика (англ. autoencoder), и затем, взятие эмбеддинга (векторного представления) на основе скрытого латентного представления для описания свойств финансового актива как временного ряда.

Рассмотрим подробнее архитектуру автокодировщика – это такая специальная архитектура искусственных нейронных сетей, при которой структура нейросети состоит из двух частей: первая – это энкодер (англ. encoder), а вторая – декодер (англ. decoder).

Автокодировщики используются для решения таких задач, как генерация новых объектов (из того же распределения, что и выборка), понижения размерности, формирования эмбеддингов на основе скрытого латентного представления автокодировщика. Ключевой особенностью автокодировщиков является то, что такой алгоритм обучения является обучением без учителя.

Алгоритм обучения автокодировщика следующий: во-первых, энкодер преобразует входной вектор в его скрытое представление преобразованием , затем декодер восстанавливает сигнал по его скрытому представлению преобразованием . Затем, по результату работы автокодировщика для входного вектора считается значение функции ошибки , функционал которой минимизируется по весам автокодировщика методом обратного распространения ошибки и оптимизации весов автокодировщика одним из вариаций градиентного спуска.

Поскольку и – это векторы в , то функция ошибки в случае автокодировщика – это одна из классических функций ошибок в задаче регрессии, например MSE (минимизация суммы квадратов ошибок) или MAE (минимизация суммы модулей ошибок).

Стоит отметить, что одной из отличительных особенностей автокодировщиков является то, что в случае большой выборки, автокодировщик будет очень долго обучаться, чтобы начать показывать хорошие результаты, потому что в таком случае автокодировщик будет выучивать распределение входных данных, которых оказалось много.

**1.5. Алгоритмы решения задачи кластеризации**

Рассмотрим несколько классических алгоритмов решения задачи кластеризации, которые будут использованы далее, и результаты которых будут сравнены далее.

Первый алгоритм кластеризации – KMeans. Алгоритм KMeans – один из наиболее популярных методов кластеризации, ввиду его простоты и сходимости.

Исходя из названия, в алгоритме KMeans заранее задается число кластеров , после чего алгоритм стремится минимизировать сумму квадратов отклонений объектов от центра кластера этого объекта. В качестве центра тяжести кластера, как правило, берут центр масс объектов в кластере.

Математическая постановка алгоритма состоит в следующем: имеется набор наблюдений , каждое наблюдение – это -мерный вектор, тогда результат работы алгоритма разбивает наблюдений на подмножеств так, чтобы минимизировать внутрикластерную дисперсию (сумму квадратов отклонений). Задача алгоритма KMeans – найти такое разбиение , что:

Алгоритм KMeans является итерационным: на каждой итерации алгоритма пересчитывается центр масс для каждого кластера, после чего для каждого наблюдения находится ближайший центр масс и данное наблюдение помечается номером кластера, который имеет данный центр масс. Алгоритм завершается, когда на какой-либо итерации не произошло изменения суммарного внутрикластерного расстояния, однако программно прекращение работы алгоритма происходит, когда на какой-либо итерации изменение суммарного внутрикластерного расстояние оказалось меньше, чем на .

Стоит отметить, что существует несколько вариаций алгоритма в зависимости от того, как инициализируются центры кластеров по умолчанию. Есть несколько вариантов инициализации центра кластеров:

* Во-первых, инициализация центров кластеров случайными объектов из выборки, то есть, выбирается случайных объектов из выборки и эти объекты назначаются центрами кластеров;
* Во-вторых, инициализация центров кластеров происходит случайно по каждой компоненте всех векторов, другими словами, для каждого центра кластеров каждая компонента вектора инициализируется случайно из какого-либо распределения, например, из равномерного;
* В-третьих, инициализация центров кластеров происходит в соответствии с алгоритмом kmeans++, который заключается в нахождении «оптимальных» начальных значений центров кластеров, интуитивно смысл алгоритма состоит в том, чтобы распределить центры кластеров равномерно (в смысле плотности соседних объектов) по всему пространству признаков.

К преимуществам KMeans можно отнести гарантированную сходимость алгоритма, простоту использования и быстроту реализации, а к недостаткам – необходимость выбора количества кластеров , чувствительность к выбросам, зависимость результатов работы алгоритма от изначальной инициализации центров кластеров, а также тот факт, что не гарантируется нахождение глобального оптимума суммарного внутрикластерного расстояния, как правило, находится лишь один из локальных минимумов.

Второй алгоритм кластеризации – DBSCAN (англ. density-based spatial clustering of applications with noise). В современном сообществе машинного обучения, данный алгоритм многими считается лучшим алгоритмом кластеризации, поскольку в данном алгоритме не требуется заранее указывать количество кластеров, поскольку алгоритм хорошо умеет работать с выбросами, а также, так как данный алгоритм основан на плотности объектов в признаковом пространстве, то алгоритм группирует объекты, которые имеют высокую плотность и расположены тесно, а объекты, имеющие низкую плотность, алгоритм помечает как выбросы.

У алгоритма DBSCAN есть два гиперпараметра – это минимальное число точек в кластере и радиус эпсилон-окрестности . Работа алгоритма DBSCAN начинается с произвольной точки выборки, после чего находится число точек в -окрестности данной точки – в случае, если это число не меньше , то точка и все ее соседи в -окрестности начинают образовывать кластер, в противном случае, точка помечается выбросом, шумом, однако, заметим, что эта точка может быть позже найдена в -окрестности какой-либо точки из ее соседей и включена в кластер.

Другими словами, если найдена точка с высокой плотностью, то она и все ее соседи в -окрестности добавляются в один кластер. Данный процесс продолжается, пока не будет найден полный состав данного кластера, после чего выбирается новая случайная точка и процесс повторяется для нее и ее соседей, и так до тех пор, пока все точки выборки не будут помечены шумом либо конкретным номером кластера.

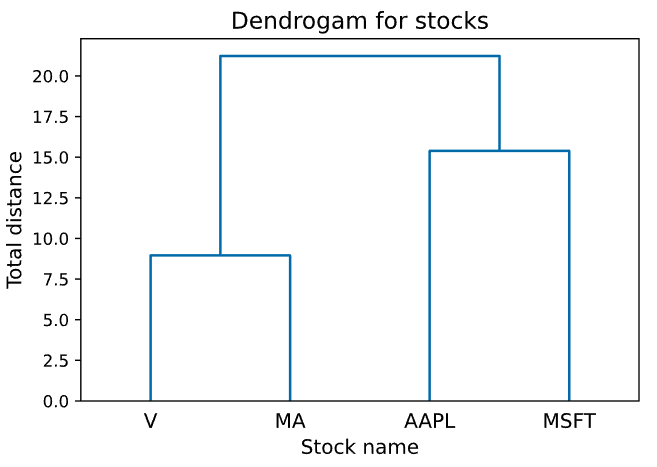
Стоит отметить, что DBSCAN может использоваться с любой функцией расстояния, а также DBSCAN может работать на основе предподсчитанной матрицы расстояний.

К преимуществам DBSCAN можно отнести то, что DBSCAN устойчив к выбросам, не требует априорного задания числа кластеров, имеет всего два интерпретируемых гиперпараметра, которые могут гибко настраиваться, а также, что DBSCAN находит плотные кластеры произвольной формы. К недостаткам же, как правило, относят лишь то, что для большого числа признаков и большого количества элементов в выборке DBSCAN алгоритм будет работать очень долго.

Третий алгоритм кластеризации – иерархическая кластеризация. Иерархическая кластеризация – такие алгоритмы кластеризации, которые направлены на создание иерархии, которую обычно визуализируют в виде дерева вложенных кластеров. Иерархическая кластеризация бывает агломеративная, когда новые кластеры создаются путем объединения более мелких кластеров, и дивизионная, когда новые кластеры создаются путем разбиения больших кластеров на более мелкие. Здесь и далее будут рассматриваться только алгоритмы агломеративной кластеризации, поскольку они более популярны, чем алгоритмы дивизионной кластеризации.

Для визуализации результатов иерархической кластеризации используется дендрограмма – такое дерево, построенное по матрице расстояний объектов выборки, в котором в листьях дерева находятся объекты выборки, а в узлах дерева находятся подмножества объектов выборки, то есть кластера, при этом объединение двух узлов создает новый узел, в котором находится объединение подмножеств объектов его узлов-потомков.

Пример дендрограммы для выборки из четырех объектов финансового рынка, четыре компании – Visa, Mastercard, Apple, Microsoft, представлен на рисунке ниже



В данном случае видно, что в дереве 4 листа – это названия компаний, и 3 узла, первый узел – объединение Visa и Mastercard в один кластер, второй узел – объединение Apple и Microsoft в один кластер, третий узел – объединение всей выборки в один кластер. Каждый узел находится на какой-либо высоте по оси ординат – это значит, что суммарное внутрикластерное расстояние объектов данного кластера равно отметке по высоте, на которой находится данный узел в дендрограмме.

Понятно, что для того, чтобы получить кластера, необходимо обрезать дерево по определенной высоте, тогда после обрезания дерева по определенной высоте, останутся поддеревья, и в корне каждого полученного поддерева (поддеревом может быть и дерево из одной вершины, из одного объекта) будет состав кластера.

Стоит отметить, что дендрогромма – это ни в коем случае не алгоритм кластеризации, это лишь визуализация интуиции, стоящей за алгоритмами агломеративной кластеризации.

Более формально, алгоритм агломеративной кластеризации состоит в следующем:

* Для каждой пары объектов выборки считаются парные расстояния по заданной метрике расстояний, которая, на самом деле, математически может быть абсолютно любой, и получается матрица расстояний
* Каждый объект данных рассматривается как отдельный кластер;
* Два кластера, которые являются ближайшими по какому-либо межкластерному расстоянию, объединяются в один кластер, после чего считается (либо обновляется) матрица расстояний между кластерами;
* Третий пункт повторяется, пока не останется один кластер

В случае, когда расстояние считается между объектами, все довольно понятно – это, как правило, какое-либо классическое расстояние, например – Eвклидово расстояние, расстояние Чебышева или манхэттенское расстояние.

Однако, не совсем очевидный момент – это то, как считается расстояние между кластерами, ведь кластер – совокупность объектов, и не совсем понятно, как можно считать расстояние между двумя совокупностями объектов. Эта проблема решается какой-либо агрегацией объектов в кластерах для подсчитывания расстояния между кластерами.

Существует несколько классических расстояний между кластерами:

* Метод одиночной связи – расстоянием между кластерами и является минимальное расстояние между всеми парами объектов из этих кластеров:
* Метод полной связи – расстоянием между кластерами и является максимальное расстояние между всеми парами объектов из этих кластеров:
* Метод средней связи – расстоянием между кластерами и является среднее расстояние между всеми парами объектов из этих кластеров:
* Метод центроидной связи – расстоянием между кластерами и является квадрат расстояния между центрами кластеров:
* Метод связи Уорда – расстоянием между кластерами и является минимизации суммы квадратов расстояний объектов внутри кластеров, такой подход, с точки зрения статистики, фактически, приводит к минимизации дисперсии каждого отдельного кластера, тогда итоговая формула выглядит следующим образом:

Преимуществами агломеративной иерархической кластеризации являются интуитивная интерпретируемость алгоритма, гибкая настраиваемость ввиду большого количества различных возможных расстояний между кластерами и объектами выборки, интерпретируемая визуализация результатов работы алгоритма.

К недостаткам агломеративной иерархической кластеризации же можно отнести вычислительную стоимость работы алгоритма, потому что для большого количества элементов выборки расчет матрицы попарных расстояний – это очень дорогая операция, чувствительность к выбросам и шуму, а также необходимость определения числа кластеров.

**1.6. Кластеризация финансовых активов индекса S&P 500**

fasd

**1.7. Сравнение результатов кластеризации финансовых активов индекса S&P 500**

adsf

**2. ФОРМИРОВАНИЕ ОПТИМАЛЬНОЙ СТРУКТУРЫ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ**

**3. ПРОГНОЗИРОВАНИЕ ДОХОДНОСТИ ФИНАНОСОВЫХ АКТИВОВ**