

*Московский государственный университет имени М.В. Ломоносова*

*Факультет вычислительной математики и кибернетики*

*Кафедра математической статистики*

**Алгоритм Скользящего Разделения Смесей  
 и его применение в анализе реальных данных.**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Выполнил:**

Студент 4 курса, ВВО

Птицын Евгений Генрихович

**Научный руководитель:**

д.ф.-м.н., заслуженный профессор

Королёв В.Ю.

**Рецензент:**

к.ф.-м.н., доцент  
Захарова Т.В.

Оглавление

[1 Введение 2](#_Toc8306086)

[2 Вводные понятия 3](#_Toc8306087)

[2.1 Смесь нормальных законов распределения 3](#_Toc8306088)

[***2.1.1*** Определение 3](#_Toc8306089)

[3 Нахождение оценок максимального правдоподобия 7](#_Toc8306090)

[3.1 Метод максимального правдоподобия 7](#_Toc8306091)

[3.2 ЕМ-алгоритм 9](#_Toc8306092)

[3.3 Описание ЕМ алгоритма 9](#_Toc8306093)

[***3.3.1*** Е- и М- шаги 9](#_Toc8306094)

[1.1. Монотонность ЕМ-алгоритма 10](#_Toc8306095)

[***3.3.2*** ЕМ алгоритм как проксимальный 11](#_Toc8306096)

[***3.3.3*** Скорость сходимости 13](#_Toc8306097)

[***3.3.4*** Выбор начального приближения 14](#_Toc8306098)

[***3.3.5*** Правила остановки 15](#_Toc8306099)

[4 Скользящее разделение смесей 17](#_Toc8306100)

[4.1 Описание метода 17](#_Toc8306101)

[4.2 Влияние выбора ширины окна 17](#_Toc8306102)

[4.3 Декомпозиция волатильности финансовых индексов 17](#_Toc8306103)

[***4.3.1*** Волатильность 17](#_Toc8306104)

[5 Практическая часть 18](#_Toc8306105)

[5.1 Реализация СРС с применением CUDA. 18](#_Toc8306106)

[6 Заключение 19](#_Toc8306107)

[7 Список литературы 19](#_Toc8306108)

[8 Приложение А 19](#_Toc8306109)

# Введение

Анализ данных в современном мире занимает все более решающую роль. Он позволяет получить дополнительную, а зачастую и основную информацию необходимой для принятия важных решений, например покупка акций.

Журнал Экономист выпустил статью *«Самый ценный ресурс больше не нефть, а данные»* (Economist, 2017), так же как нефть совершила индустриальную революцию и полностью изменила жизнь людей так же и анализ данных в XXI веке полностью меняет нашу жизнь. Появившиеся методы анализа данных позволяют производить анализ огромных массивов информации.

В данной работе рассматривается один из методов анализа данных, с помощью разложения волатильности на компоненты, метод скользящего разделения смесей (СРС).

Современные финансовые рынки представляют из себя очень сложные системы, состояние которых зависит от огромного количества факторов и параметров. Изменение цен на акции, фьючерсы и другие инструменты, равно как и составные инструменты индексы подвержено влиянию таких факторов, как например состояние экономики разных стран, политический климат, различные микро- и макроэкономические показатели, а также, не в последнюю очередь состояние самого рынка.

Целью данной работы является обзор применения метода скользящего разделения смесей в анализе данных. А именно получение практических результатов применения метода скользящего разделения смесей на различных данных.

В первой части работы дается теоретическое обоснование метода и его основных составляющих.

Во второй части работы приводится практическая ре.

Основная идея и теоретическое обоснование метода скользящего разделения смесей было представлено в работе [Королёв, 2011].

ЕМ алгоритм был описан в работе [Dempster, Laird, Rubin, 1977]

# Вводные понятия

## Смесь нормальных законов распределения

### Определение

Прежде всего нам потребуется ввести понятие смеси нормальных законов распределения. Рассмотрим функцию вида , при этом множество имеет борелевскую σ-алгебру Σ. Более того предположим что для каждого фиксированного функция представляет из себя функцию распределения по , а при любом фиксированном функция является измеримой по , то есть для любого выполняется условие . Пусть Q – вероятностная мера, определенная на измеримом пространстве . Тогда функцию распределения

называют смесью функции распределения по относительно Q. Распределение называется смешиваемым, тогда как мера Q задает смешивающее распределение. Если **Y –** m-мерная тождественная случайная величина (т.е. ), определенная на вероятностном пространстве , тогда функцию распределения можно записать в виде

Если — плотность распределения, соответствующая функции распределения ,

тогда смеси соответствует плотность

Если случайный вектор **Y** имеет дискретное распределение и принимает значения с вероятностями соответственно, то получится смесь вида

называемая дискретной. В этом случае функции распределения называются компонентами смеси , а числа называются весам соответствующих компонент,   
. Ели же в дискретной смеси число ненулевых весов конечно (то есть случайный вектор принимает конечное число значений), то дискретную смесь называют конечной.

Если функции распределения соответствует плотность , то дискретной смеси соответствует плотность

Особо подчеркнем, что при разных значениях параметра ***y*** функции распределения могут относиться к различным типам распределения. К примеру, если смесь дискретна, то есть мера Q приписывает вероятность точкам , причем , где — функции распределения, возможно, соответствующие разным типам при разных *j*, то

Более того, если в таком случае компоненты абсолютно непрерывны и имеют плотности , то смесь так же будет абсолютно непрерывной с плотностью

Особую роль играют сдвиг/масштабные смеси. Формально их можно определить следующим образом. Пусть в определении, сформулированном выше, *m =* 2. Предположим, что вектор ***y*** имеетвид.  
где и , так что функция распределения допускает представление

Тогда — это положительная полуплоскость, то есть , и функция распределения

называется *сдвиг/масштабной смесью* распределения *F* относительно Q. Здесь *u* — это параметр масштаба, а *v* — параметр сдвига (положения). Если функция распределения *F* имеет плотность *f*, то функция распределения соответствует плотность  
так что смеси соответствует плотность (\_\_\_)

Если и *—* случайные величины, заданные на одном и том же достаточно богатом вероятностном пространстве, так что при каждом фиксированном значении пары случайных величин случайная величина имеет функцию распределения , то смесь (\_\_\_) может быть записана в виде

Более того, в таком случае из теоремы Фубини вытекает, что функция распределения соответствует случайной величине , где случайная величина и случайный вектор стохастически независимы. Кстати, легко убедиться, что в таком контексте чисто сдвиговая смесь является не чем иным, как функцией распределения суммы двух независимых случайных величин и , то есть сверткой их функций распределения. В то же время, чисто масштабная смесь  
 является ничем иным, как функцией распределения произведения двух независимых случайных величин и .

Что бы определить дискретную сдвиг/масштабную смесь функции распределения , положим , где , и в качестве специального случая получим

Если при этом , то мы получаем чисто масштабную конечную смесь

Если функция распределения абсолютно непрерывна и имеет плотность , то смеси (3.\_\_\_) функций распределения соответствует смесь плотностей

Физический смысл понятия смеси вероятностных распределений может быть проиллюстрирован на примере дискретной смеси. Рассмотрим некоторую популяцию, которая не является однородной и, в свою очередь, состоит из некоторого числа, скажем, субпопуляций. Предположим, что наблюдаемый признак (или наблюдаемая характеристика) внутри *j*-й субпопуляции распределен в соответствии с функцией распределения , которую можно интерпретировать как условную вероятность того, что значение наблюдаемого признака у случайно выбранного индивидуума будет меньше, чем *x*, при условии, что случайно выбранный индивидуум является представителем *j*-й субпопуляции. Пусть вероятность того, что при случайном выборе индивидуума из всей (генеральной) популяции будет выбран представитель именно *j*-й субпопуляции, равна . Тогда по формуле полной вероятности безусловная вероятность того, что значение наблюдаемого признака у индивидуума, случайно выбранного из всей генеральной популяции, будет меньше, чем , окажется равной

В определенном смысле операция смешивания вероятностных распределений обеспечивает возможность формально интерпретировать популяции, реально являющиеся неоднородными, как однородные. Очень часто нельзя непосредственно определить тип субпопуляции, к которой принадлежит очередное наблюдение. Вследствие этого вся (генеральная) популяция вынужденно считается однородной, хотя на самом деле она таковой не является и содержит индивидуумов, принадлежащих к существенно различным типам. Именно такая ситуация типична для анализа финансовых временных рядов. Поэтому чрезвычайно важно иметь возможность осуществить операцию, в некотором смысле обратную операции смешивания, а именно, операцию разделения (расщепления) смесей. Статистические процедуры, реализующие эту операцию, описываются ниже. Эти процедуры в значительной степени зависят от свойства идентифицируемости смесей вероятностных распределений.

# Нахождение оценок максимального правдоподобия

## Метод максимального правдоподобия

Оценки максимального правдоподобия и основанные на функции правдоподобия выводы являются важнейшими в статистической теории и анализе данных. Метод максимального правдоподобия является базовым методом с хорошими свойствами. Это наиболее часто используемая методика в частотном анализе, и она может быть в равной степени применима к поиску апостериорного распределения в Байесовском анализе. (C. P. Robert, Bayesian Computational Methods)

Зачастую Байесовские решения оправдывают с помощью функции правдоподобия и оценок максимального правдоподобия (MLE), и Байесовские решения похожи на оценки правдоподобия со штрафом. Оценка максимального правдоподобия является универсальным методом и широко используется в любой области, где используются статистические методы.

Предположим, что наблюдаемые данные имеет плотность , где вектор, содержащий неизвестные параметры в постулированной форме для плотности **.** Наша цель максимизировать функцию правдоподобия как функцию , над параметрическим пространством . Так, что,

Или эквивалентно для логарифма правдоподобия,

Цель ОМП - оценить так, чтобы определить последовательность корней (2.1) согласованную и асимптотически эффективную. Подобная последовательность существует при подходящих регулярных условиях (Cramér 1946). Почти наверно, эти корни соответствуют локальному максимуму на . Для оценочных моделей в общем, функция правдоподобия имеет глобальный максимум на . Типичная последовательность корней, (2.1) с желаемыми асимптотическими свойствами, получается взяв как корень, который глобально максимизирует ; в этом случае, является Оценкой Максимального Правдоподобия. Далее будем называть – ОМП даже в ситуациях, когда она может не давать глобальный максимум правдоподобия. Несомненно, в некоторых примерах на смесях (McLachlan and Peel 2000, Chap. 3), функция правдоподобия не ограниченна. Однако, для этих моделей, при обычных условиях регулярности, могут существовать последовательности корней (2.1) со свойствами согласованности, эффективности и асимптотический нормальности. (McLachlan and Basford 1988, Chap. 12).

Когда функция правдоподобия или логарифмическая функция правдоподобия являются квадратичными в параметрах, как в случае независимых нормально распределенных наблюдений, ее максимум может быть получен решением системы линейных уравнений с параметрами. Однако, часто на практике функция правдоподобия не является квадратичной, что приводит к нелинейным проблемам в оценке максимального правдоподобия. Например: (а) модели приводят к математическим ожиданиям нелинейным в параметрах; (б) несмотря на возможную линейную структуру, функция правдоподобия не квадратичная по параметрам по причине не «нормальных» ошибок, отсутствующих данных или зависимостях.

Традиционно ОМП в этих ситуациях производится численно итерационным методом, решая уравнения, например методом Ньютона-Рафсона (NR) и его вариантами, такими как метод Фишера. При разумных предположениях на и достаточно точных начальные значениях, последовательность полученная методом Ньютона-Рафсона имеет локальную квадратичную сходимость к решению для (2.1). Квадратичная сходимость рассматривается как основное преимущество метода Ньютона-Рафсона. Но в применении эти методы могут быть утомительны аналитически и численно даже в сравнительно простых случаях. Смотри McLachlan and Krishnan (2008, Sect. 1.3) и Meng and van Dyk (1997). ЕМ-алгоритм предлагает лучшую альтернативу по многим критериям. Сейчас это популярный инструмент для итерационного нахождения ОМП в различных проблемах с отсутствующими или неполными данными.

## ЕМ-алгоритм

## Описание ЕМ алгоритма

### Е- и М- шаги

В результате действия ЕМ-алгоритма, представляющего собой итерационную процедуру, вычисляется последовательность значений параметра . Если задано некоторое значение , то вычисление следующего значения можно условно подразделить на два этапа, аббревиатура наименования которых и дала название всей процедуре. Опишем эти этапы.

**E-шаг.** Вычисление математического ожидания (Expectations). Вычисляется значение вектора скрытых переменных по текущему приближению вектора.

Определим функцию как условное математическое ожидание логарифма полной функции правдоподобия при известном значении наблюдаемой компоненты **X*:***

В этом определении *θ* являетсяаргументом функции , **X** и – параметры, так что соотношении (3.1) символ означает усреднение по **Y** относительно меры .

При известном значении **X = *x*** функцию можно вычислить по формуле

**М-шаг**. Производится максимизация функции правдоподобия и находится следующее приближение вектора θ.

Итерационный процесс останавливается в соответствии с заранее согласованным критерием остановки. Например, заранее выбирается какая-нибудь метрика и фиксируется малое положительное число ε. Процесс останавливается на m-ом шаге, если .

Заметим, что иногда название «ЕМ-алгоритм», данное в бумаге Dempster et al. (1977), объясняют не упомянутой выше аббревиатурой английских слов *Expectation-Maximization,* но возводят к термину *Estimation-Maximization.* (например (Айвазян и др., 1989)). По всей вероятности, первый термин все же лучше отражает суть ЕМ-алгоритма.

## Монотонность ЕМ-алгоритма

Свойство монотонности ЕМ-алгоритма было впервые установлено в работе (Шлезингер, 1965). Впоследствии это свойство обобщенных и модифицированных версий ЕМ-алгоритма систематически исследовалось в работах (Dempster, Laird and Rubin, 1977), (Everitt and Hand, 1981), (Boyles,1983), (Wu, 1983), (Redner and Walker, 1984), (Jordan and Xu, 1996), (Xuand Jordan, 1996).

Недавно в работах (Neal and Hinton, 1998) и (Chretien and Hero, 2000) было замечено, что ЕМ-алгоритм принадлежит к классу так называемых проксимальных алгоритмов (или PP-алгоритмов, от английского термина Proximal Point algorithms) (в статье (Neal and Hinton, 1998) отмечено соответствующее ключевое свойство ЕМ-алгоритма, но при этом ЕМ-алгоритм формально не идентифицирован как PP-алгоритм). Это замечание существенно упрощает исследование свойства монотонности ЕМ-алгоритма. При этом главную роль играет следующее представление функции .

Из соотношения вытекает, что

По этому

Согласно принятой выше терминологии, обычная статистическая процедура поиска оценок максимального правдоподобия направлена на максимизацию по логарифма неполной функции правдоподобия, который при известном значении равен первому слагаемому в правой части (3.2.1)

### ЕМ алгоритм как проксимальный

Рассмотрим расстояние Кульбака–Лейблера (Kullback and Leibler,1951) (см. также, например, (Кульбак, 1967), (Cover and Thomas, 1991)) между условными плотностями и , определяемое как условное математическое ожидание логарифма отношения правдоподобия при известном значении наблюдаемой компоненты относительно меры :

Расстояние Кульбака–Лейблера можно вычислить по формуле

При этом, применяя неравенство Иенсена, в силу вогнутости логарифмической функции легко убедиться, что

Таким образом, расстояние Кульбака–Лейблера удовлетворяет условиям:

1.   
2.

Положив

мы таким образом замечаем, что свойства 1 и 2 расстояния Кульбака–Лейблера соответствуют условиям, которые определяют штрафную функцию в определении проксимального алгоритма.

Осталось убедиться, что с и расстоянием Кульбака–Лейблера в качестве штрафной функции соотношение (5.3.11), определяющее проксимальный алгоритм, трансформируется в соотношение, определяющее М-этап ЕМ-алгоритма. Действительно, при таких и имеем

Заметим, что второе слагаемое в правой части последнего соотношения (равное дифференциальной энтропии условного распределения ) не зависит от . Поэтому

Но выражение в фигурных скобках в правой части оказывается в точности равным функции , откуда вытекает требуемое соотношение

Другими словами, рекуррентные соотношения  
и

задают одну и ту же последовательность, то есть ЕМ-алгоритм является специальным проксимальным алгоритмом. А так как любой проксимальный алгоритм обладает свойством монотонности, то свойство монотонности оказывается присущим и ЕМ-алгоритму в том смысле, что, если последовательность вычисляется в соответствии с правилами, определяющими ЕМ-алгоритм, то  
то есть

[1]

### Скорость сходимости

Скорость сходимости ЕМ-алгоритма обычно меньше, чем квадратичная, доступная с Ньютоновскими методами. В работе (Dempster et al., 1977) показано, что скорость сходимости ЕМ алгоритма линейна и скорость зависит от доли информации в наблюдаемых данных. Следовательно, в сравнении с задачей с полными данными, если большая часть данных отсутствует, сходимость может быть достаточно медленной.

Определим отображение из параметрического пространства ***Ω*** в само себя, таким, что . Функция ***М*** называется ЕМ отображением.

Если сходится к какой-то точке и непрерывна, тогда стационарная точка алгоритма; такая что, должна удовлетворять условиям . Разложив в ряд Тейлора в точке , в окрестности получим что

Где есть матрица Якоби для , имеющая (i,j)-й элемент равный

Где и *d* размерность **.** Таким образом, в окрестности , ЕМ алгоритм представляет собой, по существу, линейную итерацию с матрицей отношений поскольку обычно не равен нулю. По этой причине часто называют матрицей сходимости. Для вектора , мера фактически наблюдаемой скорости сходимости есть глобальная скорость сходимости определенная как

где это любая норма на *d*-мерном Евклидовом пространстве . Отмечено, что наблюдаемая скорость сходимости равна наибольшему собственному значению при определённых условиях регулярности (Meng and van Dyk 1997). Поскольку большое значение r подразумевает медленную сходимость, глобальная скорость сходимости определена как (Meng, 1994); см. так же McLachlan and Krishnan (2008, Sect. 3.9).

### Выбор начального приближения

ЕМ алгоритм будет сходиться медленно если выбрано плохое начальное значение . Действительно в некоторых случаях, когда функция правдоподобия стремится в бесконечность, на краю параметрического пространства, последовательность оценок генерированная ЕМ алгоритмом может расходиться если выбран слишком близко к границе. Так же с приложением, где уравнение правдоподобия имеет несколько корней, соответствующих локальным максимумам, ЕМ алгоритм следует применять из широкого набора начальных значений в поиске всех локальных максимумов. Вариант ЕМ алгоритма (Wright and Kennedy, 2000) использует метод интервального анализа, для отыскания стационарных точек логарифма функции правдоподобия внутри любого обозначенного региона параметрического пространства; см. также McLachlan and Krishnan (2008, Sect. 7.9).

Различные способы определения начального значения были рассмотрены специально в рамках модели смесей. С EMMIX программой (McLachlan and Peel 2000, pp. 343–344), значение начального параметра может быть получено, автоматически используя случайные части данных, или алгоритм кластеризации k-средних, или иерархический метод кластеризации. С случайными начальными значениями, эффект центральной предельной теоремы делает параметры изначально похожими, по крайней мере в больших выборках. С программой EMMIX, есть дополнительная опция для случайного старта, чтобы уменьшить этот эффект путем случайного выбора подвыборки из данных, которая потом случайно приписывается *g* компонентам. Как описано в McLachlan and Peel (2000, Sect. 2.12), подвыборка должна быть достаточно большой, чтобы убедиться, что первый M-шаг может произвести недегенеративную оценку вектора параметров .

Ueda and Nakano (1998) предложили алгоритм детерминированного отжига EM (DAEM) для того, чтобы итерационный процесс EM мог восстановиться после плохого выбора начальных значений. Они предложили использовать принцип максимальной энтропии и аналог статистической механики, когда, параметр, скажем θ, вводится с соответствующей «температуре» в смысле отжига. С их алгоритмом DAEM. E-шаг осуществляется путем усреднения по распределению взятому пропорционально к текущей оценке условной плотности полных данных (имея наблюдаемые данные) в степени θ; см пример McLachlan and Peel (2000, pp. 58–60). В 2005 Pernkopf and Bouchaffra (2005) соединили генетический алгоритм (GA) и EM алгоритм для подгонки смесей нормальных распределений, где предложенный алгоритм менее чувствителен к начальному приближению и позволяет выбираться из локально оптимальных решений.

### Правила остановки

Необходимо иметь разумные критерии для остановки итерационного ЕМ-алгоритма. Существует несколько подходов:

Выберем малое число .

1. Условимся останавливать ЕМ-алгоритм, если расстояние между эмпирической функцией распределения и теоретической смесью функций распределения, в которую подставлены параметры, вычисленные с помощью ЕМ-алгоритма, становится меньше .

Где

- теоретическая смесь нормальных функций распределения, в которую подставлены параметры, вычисленные с помощью ЕМ-алгоритма на m-итерации,

-эмпирическая функция распределения (, если , и в противном случае).

1. ЕМ-алгоритм можно останавливать, когда разность между значениями функции правдоподобия, вычисленными на соседних итерациях, мала, то есть дальнейшая работа ЕМ-алгоритма практически не увеличит текущее правдоподобие:
2. ЕМ-алгоритм можно останавливать, когда расстояние между оценками параметров, вычисленными на соседних итерациях мало, то есть дальнейшая работа ЕМ-алгоритма практически не изменит уже полученные оценки:

В соотношениях (3.3) и (3.5) нормы можно определять по-разному.

У всех перечисленных критериев остановки есть и достоинства, и недостатки.

Критерий (3.3) основанный на близости эмпирической функции распределения и теоретической смеси функции распределения, в которую подставлены параметры, вычисленные с помощью ЕМ-алгоритма, дает хорошую возможность судить о том насколько хороша полученная аппроксимация. Однако ЕМ-алгоритм может никогда не достичь таких значений параметров теоретической функции распределения, которые обеспечивают хорошее согласие с имеющимися наблюдениями.

Критерий (3.4), ориентирующийся на функцию правдоподобия естественно согласуется с методом, используемым в ЕМ-алгоритме, но как бы сводит все имеющиеся возможности, так сказать, степени свободы алгоритма, к единственному значению. Согласно этому критерию, алгоритм может быть остановлен если последовательно вычисленные значения *L* близки, но параметры далеки от точки максимума (функция правдоподобия может принимать близкие значения на 2х достаточно отдаленных друг от друга множествах значений параметров).

Критерий (3.5) использующий расстояние между последовательно полученными оценками вектора параметров, конечно же, остановит ЕМ-алгоритм, если последний достигнет точки максимума функции правдоподобия. К сожалению, этот критерий так же остановит алгоритм в случае, когда алгоритм достигнет области, далекой от точки максимума, но в которой скорость сходимости очень низка. [1]

# Скользящее разделение смесей

## Описание метода

## Влияние выбора ширины окна

## Декомпозиция волатильности финансовых индексов

### Волатильность

[Королёв, 2011] [Dempster, Laird, Rubin, 1977][Nvidia, 2019]

# Реализация метода СРС с применением CUDA.

Для получения экспериментальных результатов, в рамках этой работы был реализован метод скользящего разделения смесей на языке Python с использованием библиотек NumPy, SciPy, Matplotlib.

Однако скорость вычислений оказалась слишком мала. Для вычисления отрезка в 1300 выборок, с размером окна порядка 1000 потребовалось чуть более 2х суток с применением процессора Intel core i7-3770(В режиме Boost до 4,5 ГГц). Поэтому была так же написана реализация алгоритма с применением вычислений с использованием графического адаптера, используя библиотеку CUDA, компании Nvidia.

CUDA это платформа для параллельных вычислений с применением графических карт (graphical process units, GPU) созданная компанией Nvidia. Оптимизация платформы позволяет запустить вычисления параллельно. Последовательную часть программы, оптимизированную для выполнения в один поток, на CPU, а основную часть, с «тяжелыми» вычислениями, на ядрах GPU (которых более тысячи) параллельно. При разработке программы для CUDA может использоваться язык C, C++, Fortran, MATLAB.

Большая разница в вычислительных мощностях при вычислении с плавающей запятой у CPU и GPU вызваны тем, что GPU специально предназначены для выполнения «тяжелых», хорошо распараллеливаемых вычислений, а именно рендеринг графики, по этой причине при разработке GPU большая часть процессора отдается под обработку данных нежели под их кеширование или управление. Точнее сказать GPU особенно хорошо приспособлены для решения задач, которые могут быть представлены в виде задач для параллельного вычисления. Одинаковая программа выполняется на множестве вычислительных блоков с большой интенсивностью вычислений (отношении операций вычисления к операциям обращения к памяти). Поскольку одна и та же программа выполняется для каждого элемента из набора данных не требуются сложные системы управления потоком команд, и поэтому, поскольку программа выполняется на многих элементах и имеет высокую интенсивность вычислений, задержки доступа к памяти могут быть спрятаны за вычислениями вместо использования больших блоков кеширования.

Благодаря тому, что

Благодаря тому что в EM алгоритме основные вычислительные затраты приходятся на вычисление матрицы размером весов Q на Е- шаге, где каждый элемент матрицы вычисляется по одной и той же формуле. Этот процесс отлично распараллеливается.

Так же и вычисления на М- шаге, хотя и являются менее затратными, тоже хорошо поддаются распараллеливанию. Все вместе позволяет практически полностью осуществить выполнение алгоритма на GPU, оставив CPU лишь контроль последовательности выполнения. Что позволяет производить вычисления используя лишь GPU, его графический процессор и встроенную графическую память.

Были произведены тесты с использованием видеокарты Nvidia GTX 660 и CPU Intel i7-3770.

Сравнительные результаты представлены на графике.

.\*Вставить сравнительный график скорости расчета одного цикла EM алгоритма на CPU и GPU\*

# Заключение

В данной работе было показано бла бла бла.

# Список литературы

1. Dempster A.P., Laird N.M., Rubin D.B. Maximum Likelihood from Incomplete Data Via the EM Algorithm // J. R. Stat. Soc. Ser. B. 1977. Т. 39. № 1. С. 1–38.

2. Nvidia. CUDA C PROGRAMMING GUIDE v10.1 // 2019. С. 301.

3. Королёв В.Ю. Вероятностно-статистические методы декомпозиции волатильности хаотических процессов. Москва: Издательство Московского университета, 2011. 512 с.

# Приложение А

Пока не знаю. Может листинги программ. Рисунки схемы таблицы буду вставлять в работу