

Санкт-Петербургский государственный университет

Системное программирование

Дулетов Дмитрий Евгеньевич

Анализ эффективности применения
сплайновых словарей в задачах
разреженной аппроксимации

Отчёт о прохождении производственной практики

Научный руководитель:
ст. преп. Е.К. Куликов

Рецензент:

Санкт-Петербург
2021

Оглавление

1. Введение	3
2. Постановка задачи	5
3. Обзор	6
3.1. Алгоритмы разреженной аппроксимации	6
3.2. Словари	6
3.3. Приближаемые сигналы	7
4. Реализация	8
4.1. Архитектура	8
4.2. AudioReader	9
4.3. Dictionaries	9
4.4. Algorithms	10
5. Эксперименты	13
5.1. Синтетические функции	13
5.2. Аудиосигналы	15
6. Заключение	18
Список литературы	20

1. Введение

Математические модели сигналов, точно описывающие определенные физические процессы, могут быть сложными и малопригодными для использования в прикладных задачах, основанных на математическом моделировании. Кроме того, практическая регистрация сигналов выполняется, как правило, с определенной погрешностью или с определенным уровнем шумов, которые по своим значениям могут быть выше теоретической погрешности прогнозирования сигналов при расчетах даже по очень точным формулам. Поэтому возникает задача аппроксимации — представления произвольных сложных функций простыми и удобными для практического использования таким образом, чтобы отклонение исходной функции от её приближения в области ее задания было наименьшим по определенному критерию приближения.

Стандартный метод построения приближения связан с выбором ряда базисных сплайн-функций и представления исходной функции как линейной комбинации базисных с некоторыми коэффициентами, вычисляемыми, как правило в соответствии с некоторой локальной схемой. В последнее десятилетие начинает активно развиваться новое направление – разреженная аппроксимация (см., например, [1]- [5] и библиографию в этих работах). Сигнал представляется в виде конечной линейной комбинации элементарных функций, выбранных из некоторого большого, в общем случае линейно зависимого набора функций. В контексте аппроксимации сплайнами такой набор называют сплайновым словарём. Отличие от простой аппроксимации состоит в том, что в разложении участвуют не все функции из набора, а лишь некоторые. При этом выбор наиболее важных функций осуществляется по некоторому, как правило жадному, алгоритму [6]. Разреженная аппроксимация является одним из наиболее динамично развивающихся и перспективных методов представления сигналов, поскольку не зависит от его частоты.

Разреженная аппроксимация получила множество применений в са-

мых разных областях, таких как обработка аудио-, видео-, графической информации, задачах сжатия и восстановления сигнала, очистка от шумов [1, 5, 7]. Статья [3] описывающая методы обработки гравитационных сигналов напрямую использует алгоритмы разреженной аппроксимации. Также нельзя не упомянуть публикации по обработке импульсных сигналов геоакустической эмиссии [1]. Помимо этого, в упомянутой обзорной статье [7] приведены примеры библиотек, основанных на разреженной аппроксимации, таких как `inpaint`. Интерес представляет библиотека `GlobalBioIm` [8] в связи с открытым кодом и обилием использования методов разреженной аппроксимации. В качестве примера использования алгоритма ОМР его создателями можно привести улучшение качества сжатия изображений [9].

Необходимо отметить, что существует целый ряд алгоритмов разреженной аппроксимации, среди которых отметим целые методологические классы — `Matching Pursuit` и `Basis Pursuit`. Однако проведённые ранее исследования по сравнению этих алгоритмов зачастую используют довольно старые наработки, возможно, не отражающие современное положение [1, 7]. За последнее время появилось множество потенциальных улучшений алгоритмов, к сожалению, изучением их производительности занимаются разве что их авторы, зачастую проводя сравнение лишь со старой версией того же алгоритма [6], либо с одним прямым конкурентом [10]. Таким образом, полноценного сравнительного анализа алгоритмов разреженной аппроксимации до сих пор не существует. Кроме того, не вполне исследованы вопросы зависимости качества работы алгоритма в зависимости от выбранного словаря в сравнении с другими алгоритмами. Такие исследования почти не проводились для словарей, основанных на сплайн-функциях, в особенности на активно разрабатываемых в последние годы аналогах B -сплайнов.

2. Постановка задачи

Целью работы является подготовка и проведение ряда экспериментов по сравнению точности разреженной аппроксимации, основанной на различных словарях, в том числе основанных на сплайн-функциях. Построенные приближения при этом анализируются как для синтетических функций, так и для реальных сигналов, рассмотренных в недавних публикациях. В результате исследования необходимо сделать вывод об эффективности применения разреженной аппроксимации, в особенности основанной на словарях из минимальных сплайнов.

Для достижения поставленной цели были поставлены следующие задачи:

1. Изучение существующих обзорных работ по сравнению алгоритмов разреженной аппроксимации
2. Выбор алгоритмов разреженной аппроксимации для проведения сравнительного анализа
3. Выбор ряда словарей для оценки эффективности алгоритмов разреженной аппроксимации
4. Реализация программного модуля, состоящего из средства обработки и дискретизации входных данных, реализаций алгоритмов разреженной аппроксимации и средств генерации словарей.
5. Проведение численных экспериментов на ряде синтетических функций, а также реальных сигналов
6. Оценка перспективности использования разреженной аппроксимации, основанной на словарях из минимальных сплайнов, в практических приложениях.

3. Обзор

3.1. Алгоритмы разреженной аппроксимации

Существует два основных подхода к построению разреженной аппроксимации — Basis Pursuit и Matching Pursuit (устойчивых эквивалентных терминов на русском языке, к сожалению, автор и его научный руководитель не знают). Основополагающими алгоритмами обоих семейств являются Lasso и OMP (Orthogonal Matching Pursuit) соответственно. Главной идеей первого алгоритма является выбор наилучших атомов на каждой итерации, при этом суммарное количество выбранных атомов в итоге минимально, в тоже время OMP выбирает по одному атому на каждом этапе и склонен использовать почти все атомы словаря в аппроксимации (подробнее см. в [7], также далее в тексте приведены псевдокоды алгоритмов).

Также интерес представляет алгоритм Stagewise Orthogonal Matching Pursuit (StOMP) [11], призванный объединить преимущества обоих рассмотренных выше подходов. Подобно OMP, приближение сигнала строится итеративно, но на каждом шаге выбирает сразу несколько векторов, как в алгоритме Lasso.

В последние годы был предложен целый ряд новых алгоритмов MP-семейства, развивающих и оптимизирующих предложенный подход. Отметим среди них, например, алгоритмы из работ [12]- [15].

3.2. Словари

Выбор словаря наряду с выбором метода разреженной аппроксимации является вторым ключевым вопросом на пути построения качественного приближения. Но, в отличие от не вполне исследованного вопроса об эффективности алгоритмов, по свойствам и особенностям применения многих известных словарей существует обзорная статья [2], имеющая более тысячи цитирований в Scopus. На основании приведённых в ней сведений было решение выбрать в качестве словарей для проведения экспериментов словарь Габора, как один из классических и весьма про-

стных в реализации, в том числе как средство дополнительного тестирования и отладки алгоритмов, как метод прототипирования. Также был использован словарь DST, основанный на дискретном косинусном преобразовании, поскольку это один из самых распространённых в прикладном использовании словарей [7].

Также были использованы ранее почти не описанные в литературе словари, основанные на сплайн-функциях. В работе рассматриваются изучавшиеся ранее словари из B-сплайнов [16, 17], а также из минимальных сплайнов двух видов: тригонометрических [22] и гиперболических [23]. Минимальные сплайны предоставляют прекрасный аппарат аппроксимации, поскольку они получаются из аппроксимационных соотношений. Подробную информацию о схеме построения минимальных сплайнов и их свойствах можно найти, например, в монографии [21].

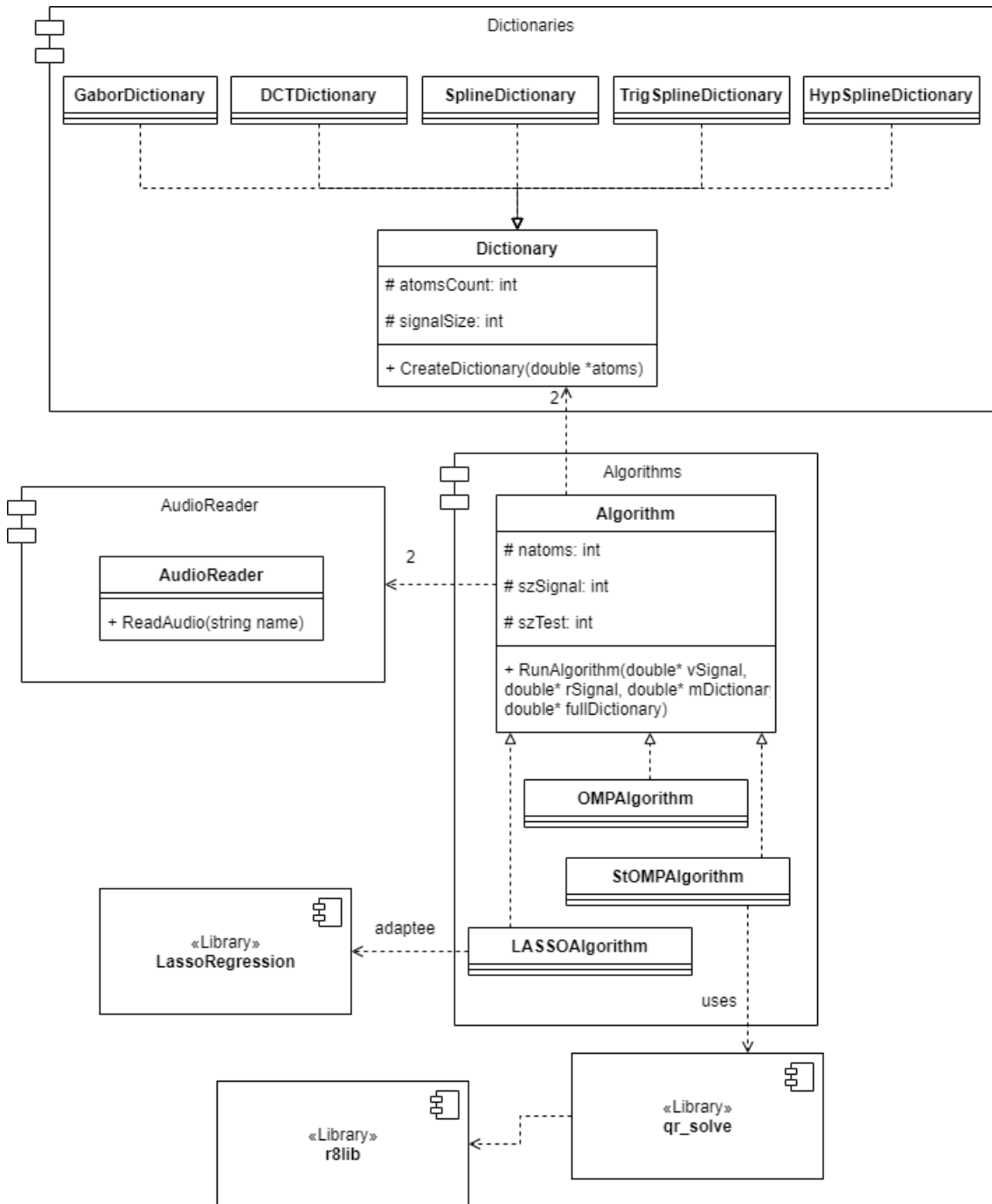
3.3. Приближаемые сигналы

В большей части статей алгоритмы сравниваются либо на случайных сигналах, либо на данных, характерных для узкой предметной области (как, например, характеристики сопротивления материалов в задачах устойчивости мостов в статье [20]).

Мы будем использовать в численных экспериментах данные двух типов: сначала будут рассмотрены синтетические данные — элементарные функции, которые, как правило, используются в качестве модельных в задачах аппроксимации. Далее будут проведены эксперименты на реальных данных. В таком качестве на практике рассматривались изображения [5, 7], аудио [3], видео [1] и любые другие сигналы. Нами был выбран набор данных из статьи [3] про аппроксимацию поведения гравитационных волн, представленных в виде wav файлов, поскольку такой вид сигналов предположительно (данная гипотеза основывается на анализе существующих публикаций) может отразить преимущества использования сплайновых словарей.

4. Реализация

4.1. Архитектура



Приложение, разработанное для постановки численных экспериментов, было написано на языке программирования C++. Несмотря на то, что большая часть программ в данной области использует MATLAB, указанный язык программирования был выбран из соображений кроссплатформенности, а также возможно использования open-source сооб-

ществом без необходимости установки проприетарных программ.

Разработанное программное решение состоит из трёх модулей. Модуль `AudioReader` считывает данные из аудиофайла и передаёт в модуль `Algorithms`. Модуль `Dictionary` создаёт два словаря выбранного типа и также передаёт их в `Algorithms`. В нём полученные данные обрабатываются и затем передаются выбранному алгоритму вместе с требуемой точностью аппроксимации сигнала. В результате работы алгоритма мы получаем массив коэффициентов при атомах словаря. Подробное описание модулей содержится в следующих разделах. С исходным кодом решения, опубликованным на Github, можно ознакомиться по ссылке <https://github.com/Duletov/Coursework>.

4.2. `AudioReader`

Поскольку аудиофайл не требует никакой специальной обработки, была использована библиотека `AudioFile`, предоставляющая протой способ считывания аудиофайлов формата `wav`. Считывание идёт последовательно, из всех аудиоканалов, для входного сигнала берётся каждая десятая точка, что отражает дискретность считываемых данных. После считывания аудиофайла данные упаковываются в массивы типа `double` и отправляются в модуль `Algorithm`.

4.3. `Dictionaries`

Данный модуль по количеству атомов - k , длине сигнала - N и выбранному виду словаря генерирует непосредственно сам словарь.

Элемент словаря Габора формируется по следующей формуле

$$\frac{\cos(2 \cdot \pi \cdot f \cdot (j - i) + p) \cdot \exp\left(\frac{-\pi \cdot (j - i)^2}{s}\right)}{s}$$

где f - frequency, s - scale, p - phase, а j и i итерлируемые параметры, i - номер текущего атома, j - позиция внутри атома. Генерация происходит итеративным перебором j от 1 до k и i от 1 до N . Подробное описание назначения каждого из параметров, а также мотивация к выбору

возможных значений приведены в статье [18].

Элемент словаря DCT имеет следующий вид

$$\cos\left(\frac{\pi \cdot (j + 0.5) \cdot i}{N}\right)$$

Генерация происходит итеративным перебором j и i , где i это номер атома, принимает значения от 1 до N , а j - позиция внутри атома, принимает значения от 1 до k .

Построение словаря на основе минимальных сплайнов происходит следующим образом: единичный отрезок делится на $n-1$ равных отрезков, где n - количество атомов, также по два фиктивных отрезка добавляется слева и справа для корректного построения сплайн-функций на концах отрезка. Далее составляется массив концов этих отрезков, будем обозначать их x_i и называть его сеткой. На каждой итерации строим атом, для этого проходим по всем точкам сетки и проверяем, попадают ли они на отрезок $[x_i, x_{i+3}]$ (именно там располагается носитель сплайна, соответствующий i -ому атому). Если точка лежит в указанном отрезке, вычисляем значение в соответствии со значением сплайн-функции (ссылки на статьи с формулами для вычисления их значений приведены выше), в противном случае она считается равной нулю.

4.4. Algorithms

В этом модуле обрабатываются словари и сигналы, полученные на предыдущих этапах, после чего запускается один из алгоритмов разреженной аппроксимации. Далее подробнее рассмотрим схему работы и псевдокод каждого из них.

4.4.1. OMP

Данный алгоритм на каждой итерации выбирает атом из словаря, вклад которого в уточнение точности аппроксимации будет наибольшим. Далее мы добавляем этот атом к словарю уже выбранных атомов, и ре-

ортагонализируем его. Далее мы нормализуем только что добавленный атом и вычисляем биортагональную функцию от получившегося словаря. Потом мы вычисляем остаток и на основании величины его нормы решаем, необходимо ли проводить ещё одну итерацию алгоритма или же остановить его.

Algorithm 1 OMP

```

for  $k < nAtoms$  do
     $iChosenAtom \leftarrow chooseAtom(residue)$ 
     $newDictionary[k] \leftarrow oldDictionary[iChosenAtom]$ 
    reorthogonalize(newDictionary)
    normalize(newDictionary[k])
    calcBiorthogonal(mBiortogonal, newDictionary[k])
    calcResidue(vSignal, newDictionary)
     $normresidue \leftarrow \sqrt{\langle residue, residue \rangle}$ 
    if  $normresidue < tolerance$  then
        break
    end if
end for

```

4.4.2. Lasso

Сам по себе является скорее принципом оптимизации, нежели алгоритмом. В данной работе использована взята библиотека с открытым исходным кодом LASSO-Regression, написанная на C++. Сам процесс работы алгоритма можно описать как итеративное улучшение коэффициентов регрессии при помощи алгоритма координатного спуска, учитывающий штрафы абсолютных значений коэффициентов. Таким образом алгоритм минимизирует следующее выражение

$$L(\omega) = \sum_{i=1}^N (y_i - \sum_{j=0}^N w_j x_{ij})^2 + \lambda \sum_{j=0}^M |w_j|$$

Мною был написан интерфейс, который из сгенерированных словарей и сигналов генерирует понятный библиотеке датасет. К сожалению, библиотека плохо оптимизирована по памяти, поэтому полноценно включить её в эксперименты не удалось, однако Лассо не был приоритетным

алгоритмом для нас, поэтому оптимизация библиотеки или замена её полноценным модулем остаётся в планах на будущее.

4.4.3. StOMP

Алгоритм StOMP представляет собой цикл, который заканчивается, если вторая норма остатка будет меньше необходимой точности. На каждом этапе алгоритма выбирается нефиксированное количество атомов, превосходящих лямбду, параметр, зависящий от нормы остатка. Эти атомы добавляются в набор уже выбранных ранее атомов и для этого набора находится псевдообратная матрица Мура-Пенроуза. Для выполнения данной операции мною была подключена библиотека `qr_solve`, которая поддерживает данную операцию без лишних преобразований типов, однако она потребовала также включения в проект библиотеки по работе с числами с плавающей точкой `r8lib`. При помощи этой матрицы мы находим коэффициенты для соответствующих набору атомов. После этого пересчитывается остаток и цикл повторяется.

Algorithm 2 StOMP

```
while  $\|residue\|_{L_2} > tolerance$  do  
   $M \leftarrow dictionary \cdot residue$   
   $lambda \leftarrow random(2, 3) \cdot \|residue\|_{L_2} \cdot \sqrt{szSignal}$   
   $I \cup find(M \geq lambda)$   
   $StOMP \leftarrow svd\_solve(I)$   
   $residue \leftarrow vSignal - dictionary \cdot StOMP$   
end while
```

5. Эксперименты

Все эксперименты проходили в два этапа. Первым этапом было тестирование алгоритмов и словарей на синтетических математических функциях, а затем на аудиофайле `Black_Hole_Billiards.wav` из статьи [3].

5.1. Синтетические функции

На данном этапе для каждого алгоритма и словаря были взяты следующие функции:

- x^2
- $\arctg(10 \cdot x)$
- $0.2 \cdot \cosh(5 \cdot x)$

Для каждой функции были взяты следующие пары параметров:

- 10 атомов, 100 тестовых точек
- 20 атомов, 100 тестовых точек
- 100 атомов, 1000 тестовых точек
- 1000 атомов, 10000 тестовых точек

Далее идут результаты тестов, в таблице указана ошибка, вычислявшаяся по формуле $\frac{R_S - R_A}{R_S}$, где R_S — значение входного сигнала в точке с максимальным его отклонением от значения потсроенной аппроксимации R_A в этой же точке.

Словарь	Функция	Алгоритм	10 10 100	20 20 100	100 100 1000	1000 1000 10 ⁵
gabor	x^2	OMP	0,0037	0,0020	0,0016	0,0014
DCT	x^2	OMP	0,0421	0,0384	0,0112	0,0011
B-Spline	x^2	OMP	0,1633	0,0817	0,0178	0,0825
Гип Spline	x^2	OMP	0,1395	0,0817	0,0668	0,0825
Trig Spline	x^2	OMP	0,1396	0,0817	0,0668	0,0825
gabor	x^2	StOMP	0,0040	0,0021	0,0045	0,0027
DCT	x^2	StOMP	0,0421	0,0384	0,0112	0,0001
B-Spline	x^2	StOMP	0,1633	0,0817	0,0178	0,0825
Гип Spline	x^2	StOMP	0,1395	0,0817	0,0668	0,0825
Trig Spline	x^2	StOMP	0,1396	0,0817	0,0668	0,0825
gabor	x^2	Lasso	0,0070	0,0042		
DCT	x^2	Lasso	0,0437	0,0374		
B-Spline	x^2	Lasso	0,0972	0,0683		
Гип Spline	x^2	Lasso	0,0978	0,0683		
Trig Spline	x^2	Lasso	0,0972	0,0683		
gabor	$\arctg(10*x)$	OMP	0,0572	0,0640	0,0413	0,1322
DCT	$\arctg(10*x)$	OMP	0,2883	0,1415	0,0332	0,0033
B-Spline	$\arctg(10*x)$	OMP	0,2064	0,0807	0,0197	0,0826
Гип Spline	$\arctg(10*x)$	OMP	0,0944	0,0806	0,0823	0,0826
Trig Spline	$\arctg(10*x)$	OMP	0,0943	0,0807	0,0823	0,0826
gabor	$\arctg(10*x)$	StOMP	0,0572	0,0640	0,0413	0,1327
DCT	$\arctg(10*x)$	StOMP	0,2883	0,1415	0,0332	0,0033
B-Spline	$\arctg(10*x)$	StOMP	0,2064	0,0807	0,0197	0,0826
Гип Spline	$\arctg(10*x)$	StOMP	0,0947	0,0806	0,0823	0,0826
Trig Spline	$\arctg(10*x)$	StOMP	0,0943	0,0807	0,0823	0,0826
gabor	$\arctg(10*x)$	Lasso	0,2458	0,0755		
DCT	$\arctg(10*x)$	Lasso	0,2645	0,1566		
B-Spline	$\arctg(10*x)$	Lasso	0,2270	0,2184		
Гип Spline	$\arctg(10*x)$	Lasso	0,2268	0,2184		
Trig Spline	$\arctg(10*x)$	Lasso	0,2271	0,2184		
gabor	$0,2*\cosh(5*x)$	OMP	0,0501	0,0300	0,0819	0,0343
DCT	$0,2*\cosh(5*x)$	OMP	0,2154	0,0810	0,0272	0,0028
B-Spline	$0,2*\cosh(5*x)$	OMP	0,2413	0,1292	0,0543	0,0790
Гип Spline	$0,2*\cosh(5*x)$	OMP	0,2413	0,1292	0,0543	0,0790
Trig Spline	$0,2*\cosh(5*x)$	OMP	0,2412	0,1292	0,0543	0,0790
gabor	$0,2*\cosh(5*x)$	StOMP	0,0501	0,0300	0,0819	0,0345
DCT	$0,2*\cosh(5*x)$	StOMP	0,2154	0,0810	0,0272	0,0279
B-Spline	$0,2*\cosh(5*x)$	StOMP	0,2413	0,1292	0,0543	0,0790
Гип Spline	$0,2*\cosh(5*x)$	StOMP	0,2413	0,1292	0,0543	0,0790
Trig Spline	$0,2*\cosh(5*x)$	StOMP	0,2412	0,1292	0,0543	0,0790
gabor	$0,2*\cosh(5*x)$	Lasso	0,2826			
DCT	$0,2*\cosh(5*x)$	Lasso	0,1505	0,0809		
B-Spline	$0,2*\cosh(5*x)$	Lasso	0,1302	0,1207		
Гип Spline	$0,2*\cosh(5*x)$	Lasso	0,1302	0,1207		
Trig Spline	$0,2*\cosh(5*x)$	Lasso	0,1302	0,1207		

Результаты экспериментов на синтетических функциях показали, что точность работы OMP и StOMP практически не отличается. Это связано с тем, что алгоритмы, следуя похожему принципу, отбирают большое число атомов и сводят погрешность калибровки близкой к нулю. Соответственно погрешность целиком зависит от выбранного словаря.

На функции x^2 словарь Габора показывает отличные результаты, и только на 10000 точках DCT начинает опережать его. DCT в среднем кажется более удачным словарём, однако на небольших объёмах данных сплайновые словари приводят к более высокой точности.

Словари из минимальных сплайнов с разными генерирующими вектор-функциями привели к почти идентичной точности приближения, проигрывающей другим словарям.

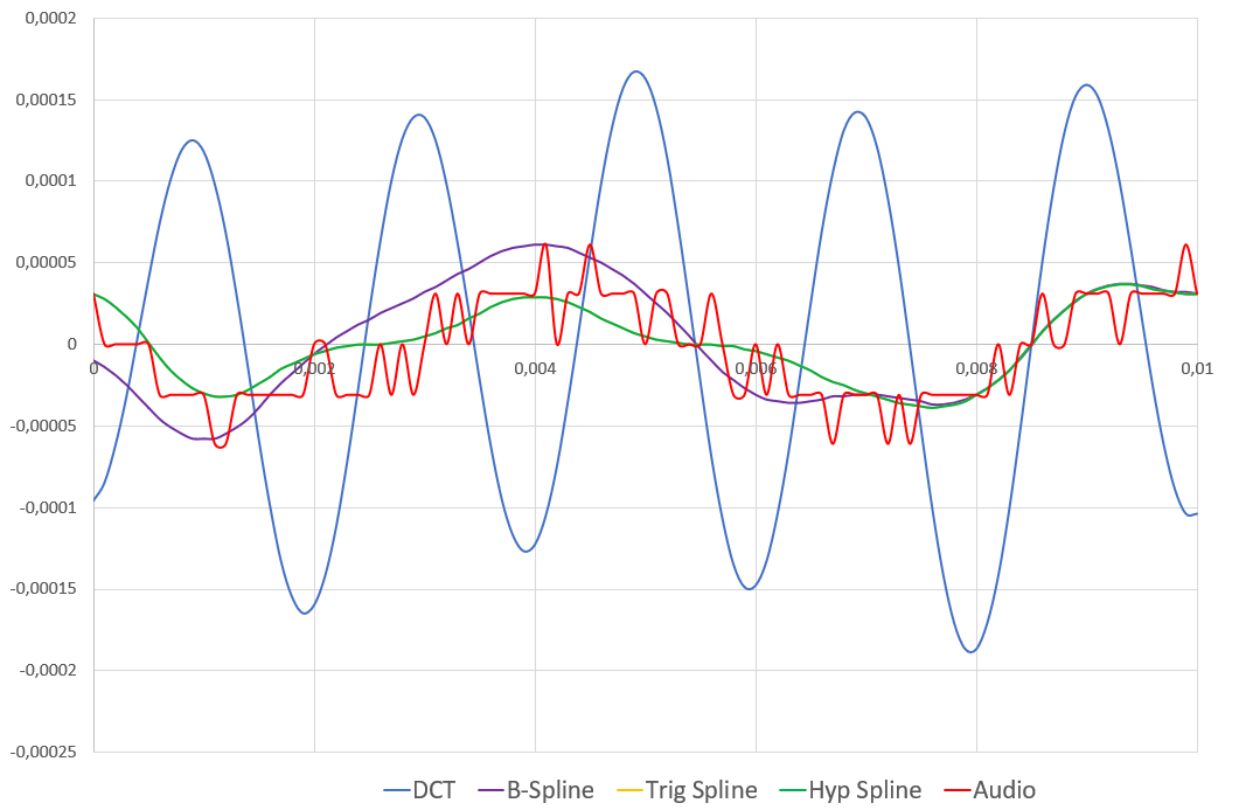
5.2. Аудиосигналы

Был проведён ряд тестов на разных фрагментах аудиофайла различного размера. Представленный в эксперименте аудиофайл состоит из 296948 точек после разбора, соответствующий сигнал имеет размер 580 КиБ, а соответствующий словарь – 164 ГиБ, поэтому было принято решение проводить эксперименты на фрагментах аудиофайла, в ходе экспериментов была установлена верхняя планка ресурсоёмкости, при которой становится возможно проведение эксперимента на среднестатистическом персональном компьютере. Таким образом, были установлены допустимые параметры эксперимента — 1000 атомов и 10000 точек, однако ограничения используемых внешних библиотек не позволили провести эксперименты на этих алгоритмах (библиотека Lasso-Regression из-за утечек памяти переполняла ОЗУ). По этим причинам далее будут представлены аппроксимации сигнала из тестового файла лишь при помощи алгоритмов OMP и StOMP с вышеуказанными параметрами при использовании различных словарей.

Результаты эксперимента оказались следующими:

Словарь	Алгоритм	Точность
Gabor	OMP	2,140471103
DCT	OMP	0,000207329
Spline	OMP	0,000063724
MinSpline	OMP	0,000063724
TrigSpline	OMP	0,000063724
Gabor	StOMP	2,378640317
DCT	StOMP	0,000206408
Spline	StOMP	0,000062573
MinSpline	StOMP	0,000062573
TrigSpline	StOMP	0,000062573

Такие результаты, скорее всего, связаны с тем, что аудиофайл имеет достаточно резкие небольшие всплески, что тяжело смоделировать при помощи функций из DCT словаря, каждый атом которого покрывает все точки, в то время как сплайновый словарь, атомы которого имеют локальный характер, значительно лучше справляется с этой задачей, также можно отметить, что минимальные сплайны на некоторых отрезках имеют лучшую аппроксимацию, чем B-сплайны, однако между собой они совпадают, именно этим можно объяснить отсутствие одного из них на графике. Ниже можно увидеть результат аппроксимации первых ста точек аудиофайла.



6. Заключение

В рамках проведённого исследования был проведён обзор существующих алгоритмов разреженной аппроксимации, как хорошо известных и активно используемых в практических приложениях, так и современных. Были выбраны алгоритмы для проведения практических экспериментов.

Также был выбран ряд словарей, активно используемых на практике, для определения эффективности их использования в сравнении со сплановыми словарями.

Был исследован вопрос, на каких данных исследователи склонны оценивать качество алгоритмов разреженной аппроксимации, и принято решение сначала провести сравнительный анализ на ряде синтетических функций, а затем на ряде сигналов, упомянутых в данной работе.

После была произведена разработка программного комплекса для постановки экспериментов оценки работы алгоритмов разреженных аппроксимаций и сравнения словарей. В рамках данной разработки был написан алгоритм StOMP при помощи библиотеки математических функций, написан адаптер для алгоритма Лассо и имплементирован алгоритм OMP, написанный разработчиками, но нуждавшийся в полном рефакторинге. Помимо алгоритмов были реализованы 5 словарей, в том числе три словаря, основанные на сплайнах, два из которых минимальные.

В рамках работы были проведены численные эксперименты как на различных синтетических функциях, так и на реальных сигналах, демонстрирующие эффективность использования тех или иных словарей в алгоритмах разреженной аппроксимации.

После проведения экспериментов было установлено, что на некоторых реальных данных использование сплайновых словарей может значительно повысить точность аппроксимации. Использование минимальных сплайнов вместо рассмотренных ранее В-сплайнов позволило повысить точность получаемых приближений на реальных данных и некоторых синтетических функциях. Однако для оценки эффективно-

сти использования минимальных сплайнов нужно проводить дальнейшие численные эксперименты. Тем не менее, уже сейчас можно утверждать, что применение словарей, основанных на минимальных сплайнах, имеет перспективы.

Список литературы

- [1] О. О. Луковенкова «Моделирование и обработка импульсных сигналов геоакустической эмиссии на базе разреженной аппроксимации», диссертация на соискание учёной степени ктн, (2016).
- [2] R. Rubinstein, A. M. Bruckstein, M. Elad «Dictionaries for Sparse Representation Modeling», Proceedings of the IEEE, Vol. 98(6), (2010).
- [3] L. Rebollo-Neira, A. Plastino «Sparse representation of Gravitational Sound», Journal of Sound and Vibration vol. 417, pp. 306–314, (2017).
- [4] M. Elad «Sparse and redundant representations», From theory to applications in signal and image processing, 376 p., (2010).
- [5] M. Andrecut «Sparse random approximation and lossy compression», IAENG International Journal of Computer Science, Vol. 38, pp. 205–214, (2011).
- [6] L. Rebollo-Neira «A dedicated greedy pursuit algorithm for sparse spectral representation of music sound», Journal of the Acoustical Society of America, vol. 140, no. 4, pp. 2933–2943, (2016).
- [7] P. Breen «Algorithms for Sparse Approximation», Year 4 Project, School of Mathematics University of Edinburgh, (2009).
- [8] M. Unser, E. Soubies, F. Soulez, M. McCann, L. Donati «GlobalBioIm: A Unifying Computational Framework for Solving Inverse Problems», Imaging and Applied Optics 2017 (3D, AIO, COSI, IS, MATH, pcAOP), (2017)
- [9] L. Rebollo-Neira, J. Bowley, A. G. Constantinides, A. Plastino «Self Contained Encrypted Image Folding», Physica A: Statistical Mechanics and its Applications 391 p., (2012).
- [10] M. A. Hameed «Comparative analysis of Orthogonal Matching Pursuit and Least Angle Regression», Michigan State University, Master of Science Thesis, (2012).

- [11] D. Donoho, Y. Tsaig, I. Drori, J.L. Starck «Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit», *IEEE Transactions on Information Theory*, vol. 58(2), pp. 1094–1121, (2006).
- [12] L. Pu, Z. Jiangtao, X. Kewen, Z. Qiao, H. Ziping «Research on Improvement of Stagewise Weak Orthogonal Matching Pursuit Algorithm», *Pattern Recognition and Image Analysis*, vol. 29(4), pp. 613–620, (2019).
- [13] Y. Zhang, G. Sun «Stagewise Arithmetic Orthogonal Matching Pursuit», *International Journal of Wireless Information Networks*, vol. 25(2), pp. 221–228, (2018).
- [14] S. Yao, Q. Guan, S. Wang «Fast sparsity adaptive matching pursuit algorithm for large-scale image reconstruction», *Journal of Wireless Communications and Networking*, vol. 78, (2018).
- [15] J. Xiang, H. Yue, Y. Xiangjun «Orthogonal Matching Pursuit Algorithm Via Improved Matching Criterion», *Proceedings of the 2nd International Conference on Digital Signal Processing - ICDSPP*, pp. 27–31, (2018).
- [16] <http://www.nonlinear-approx.info/code/matlab/uniform.html>, дата обращения 11.06.2021
- [17] M. Andrieu, L. Rebollo-Neira «Cardinal B-spline dictionaries on a compact interval», *Applied and Computational Harmonic Analysis*, vol. 18(3), pp. 336–346, (2005).
- [18] S. Chu, S. Narayanan, C.-C. J. Kuo. «Environmental Sound Recognition With Time–Frequency Audio Features», *IEEE Transactions On Audio, Speech And Language Processing*, vol. 17(6), pp. 1142 –1158, (2009).
- [19] R. Tibshirani «Regression shrinkage and selection via the lasso»,

Journal of the Royal Statistical Society. Series B (Methodological), pp. 267–288, (1996).

- [20] Z. Chen, W. Yang, J. Li, T. Yi, J. Wu, D. Wang «Bridge influence line identification based on adaptive B-spline basis dictionary and sparse regularization», Struct Control Health Monitoring, vol. 26(6), (2019).
- [21] И. Г. Бурова, Ю. К. Демьянович. «Минимальные сплайны и их приложения», Издательство СПбГУ, 365 стр., (2010).
- [22] Ю. К. Демьянович, А. А. Макаров «Необходимые и достаточные условия неотрицательности координатных тригонометрических сплайнов второго порядка», Вестник СПбГУ. Математика. Механика. Астрономия. т. 4(62), вып. 1, стр. 9–16, (2017).
- [23] Е. К. Куликов, А. А. Макаров «Об аппроксимации гиперболическими сплайнами», Записки научных семинаров ПОМИ, т. 472, стр. 179–194, (2018).
- [24] O. Kosogorov, A. Makarov «On Some Piecewise Quadratic Spline Functions», Lecture Notes In Computer Science, Numerical Analysis and Apolications, pp. 448–455, (2017).