

Compressive Sensing Experiments

Михаил Богомолов

1 Теория

Введём пару определений.

Определение 1. Назовём k -разреженным (k -sparse) вектор, который имеет k ненулевых компонент.

Определение 2. Назовём сжимаемым (compressible) вектор, в котором отсортированные абсолютные значения компонент убывают по степенному закону (т.е. k -тая порядковая статистика мажорируется функцией $\frac{c}{k^\alpha}, \alpha > 0$).

Утверждается, что $\forall s, n \exists m \ll n$ и такая матрица \mathbf{A} размера $m \times n$, что по вектору $\mathbf{y} = \mathbf{Ax}$ можно однозначно восстановить вектор \mathbf{x} , если \mathbf{x} был s -разреженным.

То есть, решение задачи $\|\mathbf{x}\|_0 \leq s, s.t. \mathbf{Ax} = \mathbf{y}$ единственно. Однако в силу того, что минимизация l_0 -псевдонормы — NP-полная задача (эквивалентна SUBSET-SUM), уже для небольших n решить исходную задачу не представляется возможным. Поэтому вместо такой задачи решается следующая:

$$\|\mathbf{x}\|_1 \rightarrow \min, s.t. \mathbf{Ax} = \mathbf{y}$$

В действительности оказывается, что это не только вычислительный трюк. Есть теорема, которая утверждает, что при некоторых условиях минимизация l_1 -нормы даёт не просто приемлемый результат, а действительно точный.

Условие это такое:

Пусть существует такое δ_s , что для любых s -разреженных векторов выполнено следующее условие:

$$(1 - \delta_s)\|\mathbf{x}\|_2^2 \leq \|\mathbf{Ax}\|_2^2 \leq (1 + \delta_s)\|\mathbf{x}\|_2^2.$$

Тогда будем называть матрицу \mathbf{A} подчиняющейся restricted isometry property (RIP) порядка s с константой δ_s .

Теорема. Для того, чтобы минимизация L_1 -нормы давала точное решение для всех s -разреженных векторов, требуется, чтобы \mathbf{A} подчинялось RIP порядка $3s$, и при этом $\delta_{2s} + \delta_{3s} < 1$.

Это довольно сложное условие, его тяжело просто так проверить. Однако для некоторых типов матриц (в основном, сгенерированных из некоторого случайного распределения) доказано, что они подчиняются RIP.

Например:

1. Матрица размера $m \times n$, элементами которой являются i.i.d. гауссовские случайные величины с дисперсией $1/m$.
2. I.i.d. случайные величины из распределения с двумя равновероятными значениями $\pm \frac{1}{\sqrt{m}}$.
3. Случайные столбцы из базиса Фурье.

Для первых двух матриц утверждается, что они удовлетворяют теореме с вероятностью $1 - O(e^{-\gamma n})$, если $s = O(m/\log(n/m))$ или $m = O(s \log(n/s))$. А значит, матрицы из i.i.d. гауссовских и бернуллиевских величин (с любыми матожиданиями и дисперсиями) хорошо подходят для compressive sensing.

Для третьего типа матриц утверждается, что они позволяют восстанавливать разреженный вектор, если $m = O(s(\log n)^4)$.

Первый тип матриц мне кажется особенно удачным для практического применения в анализе многих данных. Это связано с тем, что обычно данные представлены не в разреженном базисе, но существует некоторый ортогональный базис, в котором они разрежены. При этом произведение ортогональной матрицы и матрицы из i.i.d. гауссовских случайных величин даёт матрицу из того же распределения!

То есть: пусть Ψ — базис, в котором измерения являются разреженными, \mathbf{s} — разреженный вектор измерений, $\mathbf{x} = \Psi\mathbf{s}$ — наблюдаемый вектор измерений, Φ — матрица из гауссовских случайных величин. Так как $\Phi\Psi$ — тоже гауссовская матрица, то разреженное решение системы $\mathbf{y} = \Phi\Psi\mathbf{s}$ единственно, и совпадает с \mathbf{s} .

Третий тип удачнее первых двух, поскольку он требует гораздо меньшего количества дополнительных данных ($O(n)$ против $O(mn)$). Его можно использовать для сжатой передачи информации.

2 Что получилось сделать

Я решил проверить, что сочетание удачной sensing matrix и минимизации L_1 -нормы даёт хороший результат.

2.1 Sparse vectors

Я свёл задачу минимизации L_1 -нормы к задаче линейного программирования. Исходная задача такая:

$$\|\mathbf{x}\|_1 \rightarrow \min, s.t. \mathbf{Ax} = \mathbf{y}.$$

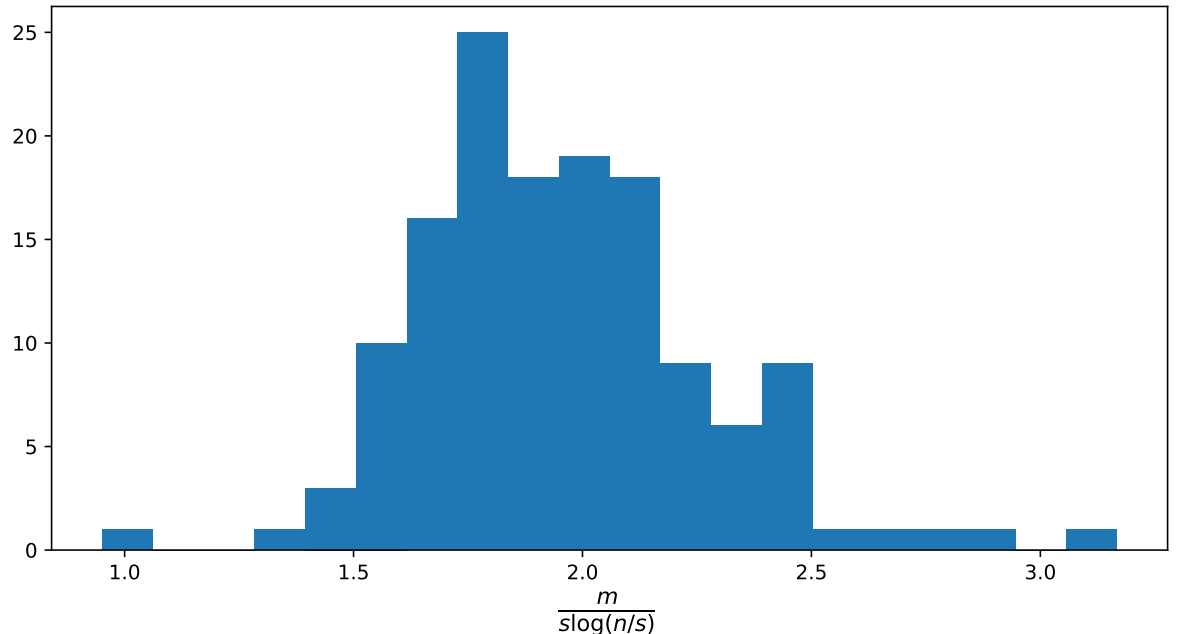
Задача линейного программирования получается такая:

$$\begin{aligned} x_1 + \dots + x_n &\rightarrow \min, \\ s.t. \mathbf{Ax} &= \mathbf{y}, \\ -\mathbf{t} &\leq \mathbf{x} \leq \mathbf{t}, \\ \mathbf{t} &\geq 0 \end{aligned}$$

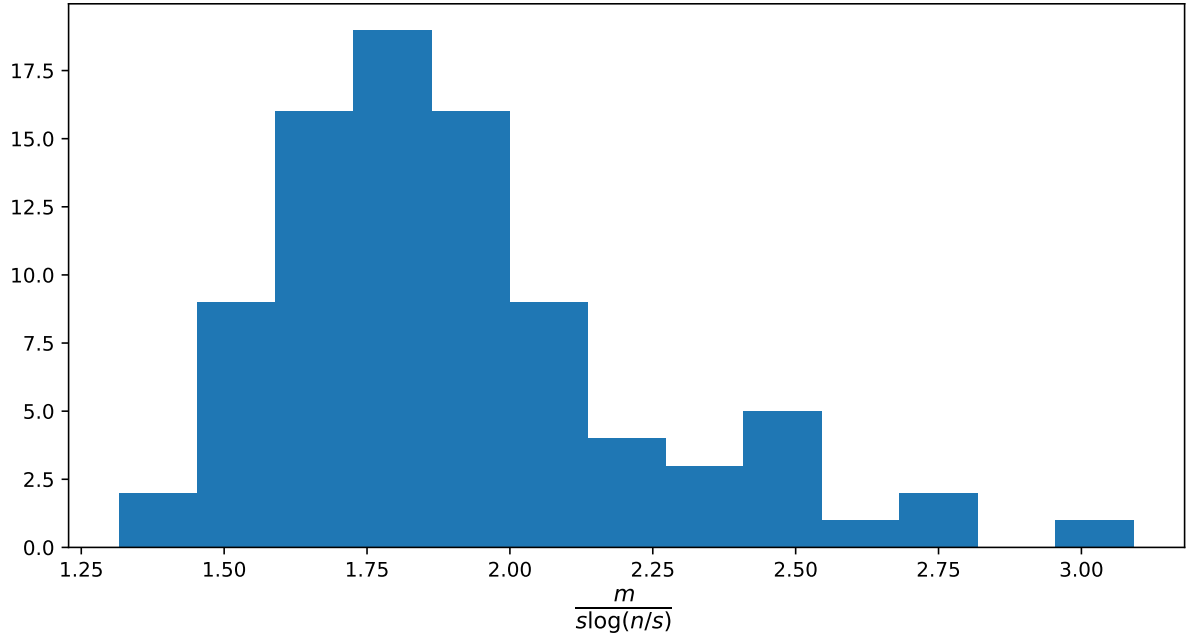
Затем я брал различные типы sensing matrices, генерировал случайные k -разреженные вектора и подбирал бинпоиском минимальное значение m , что разность настоящего ответа и найденного решения отличается на малую величину.

Получившиеся результаты для различных типов матриц:

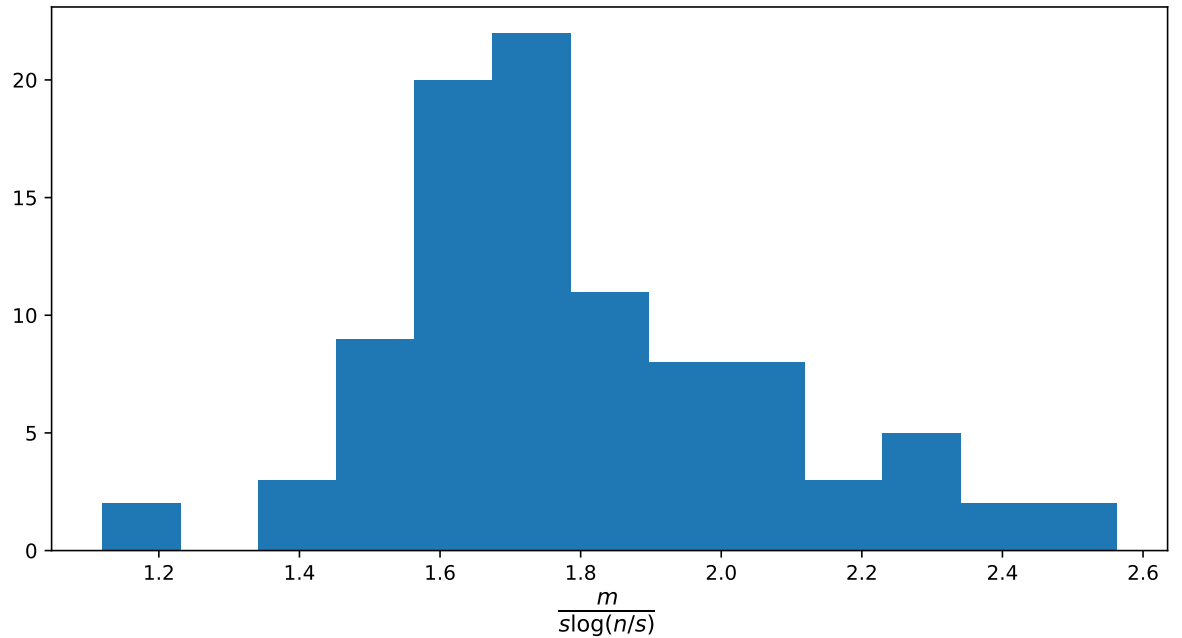
1. Гауссовские случайные величины ($50 \leq n \leq 200$, $k \leq n$, m подобрано).



2. Бернуллиевские случайные величины ($50 \leq n \leq 125$, $k \leq n$, m подобрано).



3. Рандомные столбцы из базиса Фурье ($50 \leq n \leq 125$, $k \leq n$, m подобрано).



Вообще, для последнего типа sensing matrix нужно было проверять не $m = s \log(n/s)$, а $m = O(s(\log n)^4)$, что заметно хуже, но почему-то $m = s \log(n/s)$ тоже хорошо получилось. Возможно, это связано с тем, что оценка с четвертой степенью — детерминированная оценка худшего случая, а я тестировал на случайных векторах. Но это всё равно здорово: получается, такой тип матриц не хуже полностью рандомных, при условии, что сами данные имеют случайное распределение.

2.2 Compressive vectors

Обычные данные, скорее, являются не разреженными, а сжимаемыми: в них кроме нескольких значимых компонент существует ещё много совсем небольших. Поэтому я решил ещё проверить, как хорошо можно восстановить такие вектора. Однако у меня не получилось восстанавливать эти небольшие компоненты точно. В случае разреженных векторов можно было использовать в качестве предиката в бинарном поиске условие, что найденный вектор по норме отличается от истинного на малое постоянное значение ε (например, 10^{-9}). Когда я начал зашумлять разреженные данные на весьма малый гауссовский шум (со стандартным отклонением $\varepsilon = \frac{1}{cn}, c > 1$), т.е. прибавлять к разреженному вектору \mathbf{x} гауссовский вектор \mathbf{x}_ε , оказалось, что ни \mathbf{x} , ни $\mathbf{x} + \mathbf{x}_\varepsilon$ восстановить нельзя, а в лучшем случае получается какой-то вектор, близкий к $\mathbf{x} + \mathbf{x}_\varepsilon$ и отличающийся от него по норме на что-то порядка $\varepsilon n/10$ (не очень понимаю, почему такая константа, но близкие числа получались при различных значениях n, m, k).

Я пробовал разные способы восстановления зашумлённого разреженного вектора:

1. Способ, описанный выше: простая минимизация $\|\mathbf{x}\|_1$ при условии $\mathbf{A}\mathbf{x} = \mathbf{y}$, где \mathbf{x} — уже зашумленный вектор. Это не позволяет отделить шум, но об этом дальше.
2. Модифицированный способ 1.

$$\begin{aligned} x_1 + \dots + x_n &\rightarrow \min, \\ s.t. \mathbf{y} - 2\varepsilon &\leq \mathbf{A}\mathbf{x} \leq \mathbf{y} + 2\varepsilon, \\ -\mathbf{t} &\leq \mathbf{x} \leq \mathbf{t}, \\ \mathbf{t} &\geq 0 \end{aligned}$$

Идея этого метода была в том, чтобы искать такой вектор \mathbf{x} с маленькой нормой, чтобы $\mathbf{A}\mathbf{x}$ отклонялся от \mathbf{y} не слишком сильно.

3. Однако \mathbf{y} в действительности часто мог отклоняться от нужного значения больше, чем на 2ε . Это происходило из-за того, что небольшое гауссовское отклонение по \mathbf{x} становилось довольно большим, из-за того, что в \mathbf{A} могли быть довольно большие сингулярные числа. Поэтому я решил воспользоваться сингулярным разложением, чтобы проконтролировать допустимую ошибку по \mathbf{y} .

Что я сделал: пусть \mathbf{x} зашумлён случайным гауссовским вектором \mathbf{x}_ε со стандартным отклонением ε . Пусть $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ — сингулярное разложение. Заметим, что $\mathbf{V}^\top \mathbf{x}_\varepsilon$ — тоже вектор i.i.d. гауссовских случайных величин со стандартным отклонением ε , т.к. \mathbf{V}^\top — ортогональная матрица. При домножении гауссовского вектора независимых случайных величин на диагональную матрицу $\mathbf{\Sigma}$ стандартные отклонения всех его компонент домножаются на соответствующие сингулярные числа. Получается вектор из независимых нормальных случайных величин со стандартными отклонениями, пропорциональными сингулярным числам. Будем решать следующую задачу:

$$\begin{aligned} \|\mathbf{x}\|_1 &\rightarrow \min, \\ s.t. \mathbf{U}^\top \mathbf{y} - 2\varepsilon \text{diag}(\mathbf{\Sigma}) &\leq \mathbf{\Sigma}\mathbf{V}^\top \mathbf{x} \leq \mathbf{U}^\top \mathbf{y} + 2\varepsilon \text{diag}(\mathbf{\Sigma}) \end{aligned}$$

По сути это задача поиска такого вектора \mathbf{x} с минимальной L_1 -нормой, что в его 2ε -окрестности есть такой вектор, который при домножении на \mathbf{A} даёт известный \mathbf{y} .

При подборе минимального размера m для матрицы \mathbf{A} третий способ показал наилучшие результаты, но почему-то, во-первых, эти результаты были не сильно лучше первого, а во-вторых, восстановление исходного разреженного вектора всё равно не удавалось (этот способ находил решение в 2ε -окрестности истинного, у которого L_1 -норма была чуть меньше).

2.3 Приложение этих подходов к аудиозаписям и изображениям

Я сначала пытался напрямую использовать compressive sensing для того, чтобы убрать шум с изображений и аудиозаписей, но у меня ничего не получалось сделать.

Я наблюдал такие вещи:

- Испорченные изображения гораздо быстрее и качественнее восстанавливаются с помощью KNN или медианного фильтра, нежели чем если сначала разложить изображение в разреженный базис двумерного преобразования Фурье, а затем решить compressive sensing задачу, т.е. отобразить вектор размера n в пространство меньшей размерности и попытаться найти близкий разреженный прообраз.
- С музыкой или речью то же самое. Плюс к тому, я обнаружил, что если звукозапись разбить на куски по n^2 сэмплов, затем свернуть это в квадратную матрицу, посчитать SVD-разложение и оставить совсем небольшое количество главных компонент, то результат при том же количестве информации получается чуть ли не лучше, чем если то же самое проделать с разложением Фурье (т.е. отбросить менее значительные компоненты).

Я разобрался, что пошло не так. Как я уже писал выше, хорошо удаётся восстанавливать sparse вектора, а compressible — не очень (малые компоненты в них сильно портятся). Так вот, ни про аудиозаписи, ни про изображения нельзя сказать, что они sparse. Из-за этого получается, что при решении задачи compressive sensing на них накладывается достаточно сильный шум, что мешает считать, что восстановление хорошее.

3 Итог

Если отобразить s -разреженный вектор в пространство размерности $m \sim 2s \log(n/s)$ с помощью матрицы из гауссовских/бернуллиевских случайных величин или случайного подмножества векторов из базиса Фурье, то этот вектор можно будет восстановить однозначно с помощью минимизации L_1 -нормы.