

「今日からできるスパースモデリング」

大関真之¹

京都大学 大学院情報学研究科システム科学専攻

1 目標

本講義では、「圧縮センシング」と「ボルツマン機械学習」を通して、高次元データからの確率的情報処理に基づき、本質を自動的に抽出するスパースモデリングの実践的学習を行う。「今日からできる」とあるように、実際に何を計算すれば所望の量が得られるのか、ということを中心に紹介する。奥深い数理的な背景に至るまで網羅するつもりはなく、即実践できるようにするのが目的である。

そもそもスパースモデリングとは何か。スパース性と呼ばれる本質部分に関わるものが疎であるという性質を駆使して、高次元データから説明変数のうち本質的な寄与をするものだけを自動的に抽出することができるように工夫をする枠組みである。その威力を最も分かりやすい形で発揮する圧縮センシングを取り上げ、高次元データからの本質の抽出を行い、高精度の識別精度を持つ深層学習の基礎ともなるボルツマン機械学習を取り上げる。

2 圧縮センシングの基礎

知りたいものがあるときには、必ずその知りたいものに触れる必要がある。その結果、知りたいものに関する情報を収集すること、これを観測と呼ぶ。情報の収集を行う計測の困難さに伴うコストがかかり、十分な情報を得ることができない場面が多々ある。そのときに不足した情報から、知りたいものを満足の行く形で再現できるか？できるとすれば、少ない情報から多くの重要な知見を得ることができる。できるとすれば、コストのかかる観測を減らすことができる。圧縮センシングは、観測過程と不足した情報から知りたいものを再構成する技術を組み合わせることで、上記の期待に応える新技術である。

2.1 連立方程式

中学生のときから登場した連立方程式。いい思い出も悪い思い出もあるかもしれない。このときに教わったことは、「未知数の数と方程式の数と同じ」ではないといけないよ、ということだ。例えば、

$$2x_1 + x_2 = 1 \quad (1)$$

を解きなさいといわれたとき、どうだろう。 $x_2 = -2x_1 + 1$ と式変形をして戸惑うだろう。いろんな x_1 に対して、 x_2 が決まり、結局一組の (x_1, x_2) を完全に決定することはできない。いわゆる不定というやつだ。この事実は大学生になり、行列とベクトルを自在に操るようになり、線形代数に親しんだ後でも変わらない。 N 次元の実数値ベクトル \mathbf{x} にたいして、以下の方程式が存在するとして、

$$\mathbf{y} = A\mathbf{x} \quad (2)$$

このときに M 次元の実数値ベクトル \mathbf{y} の詳細が分かっているとする。また行列 A は $M \times N$ 行列であり、方程式の中身について表しており、これも分かっているとする。そのとき未知数 \mathbf{x} を求めなさい。立派な連立方程式の問題である。

¹E-mail: mohzeki@i.kyoto-u.ac.jp

あ、逆行列を求めればいいんだ！大学の学部教育の薫陶を受けただけある、良い反応だ。しかしながら $M = N$ の場合でなければ逆行列を求めることはできない。わざわざ M と N と文字を変えたのだから、異なる場合を考えたい。しかも $M < N$ という、方程式の数が少ない**劣決定系**を扱う。(逆は**優決定系 (過剰決定系)**と呼ばれる。) 教科書通りでは、ここで終わってしまう。

2.2 解選択

方程式の数が足りないということは、決定する条件が足りないということだ。条件が足りないのだから、もう少し条件があれば解を決定することができる。例えば予め解が分かっているのであればそれを代入しておくことで、実効的に N を減らすことが可能である。それでは予め解が分かっている、 \mathbf{x} の各成分がほとんど0であるとすると、この場合は0をとる成分は方程式から除外することができる。非零の個数を K とすると、 M 個の方程式から、実質的には K 個の非零成分を求めるという話になるのだから $M < N$ であったとしても、 $M > K$ であれば、解くことが可能である。このようにほとんどの成分が零を持つ、または持つだろうと期待される性質を**スパース性**と呼ぶ。またそうした性質を持つ解を**スパース解**と呼ぶ。

劣決定系におけるスパース解

N 次元の未知ベクトル \mathbf{x} に対して、 M 次元の実数値ベクトル \mathbf{y} と $M \times N$ の観測行列 A により、以下の関係を持つとする。

$$\mathbf{y} = A\mathbf{x} \quad (3)$$

ここで $M < N$ であっても \mathbf{x} の成分のうちほとんどが零をとる (**スパース性**を持つ) とき、非零成分の個数 K が $M > K$ であれば、解が求められる。

しかしその本質的な K 個の非零成分はどこにあるのか？については未知であるでしょう。そのときにどうすれば解が求められるか？残念ながら、決定的な方法は存在せず、 N 個の成分の中から K 個の成分を取り出し、ひたすら $\mathbf{y} = A\mathbf{x}$ を満たすものを探すという方法をとらざるを得ない。 N 個のなかから K 個を取り出す組み合わせの数は、 N が大きくなるにつれて、一般に指数関数的に増大していく。非常に高次元な問題について、このような計算を行うのは現実的ではない。また K 個の非零と気楽にいうが、 K という数字は知っているのだろうか？これも知っているとは限らない。そのため、この非零成分の個数についても未知であるときに、どのような方策をとり、 $\mathbf{y} = A\mathbf{x}$ の方程式を満たす解を探せばよいだろうか。

2.3 L_1 再構成

解がスパース性を持つとすると、方策として考えられるのは、できるだけ非零の成分を少なくして (零成分を多くして) 方程式の解を満たすものを探すというものである。

L_0 再構成

スパースな解を求めるために、以下の最小化問題を解く。

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = A\mathbf{x} \quad (4)$$

ここで $\|\mathbf{x}\|_0$ を \mathbf{x} の L_0 ノルムと呼び、 \mathbf{x} の非零成分の個数を表す。

しかしながらこれを厳密に解くためには、非零成分の個数を動かしながら、そのときにどこが0になるのかを仮置きして方程式を満たすものをひたすら探すしかない。そのため計算量爆発の問題があり現実的ではない。そこで、上記の L_0 ノルムの代わりに、 L_1 ノルムを使ってみよう。 L_1 ノルムとは、各成分の絶対値の和のことである。

$$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \dots + |x_N| \quad (5)$$

この L_1 ノルムは、零成分を持たば持つほど小さくなり、非零成分については大きさが影響するため、 L_0 ノルムとは違う性質を持つが、 L_1 ノルムをできるだけ小さくすることで、零成分が比較的多めの解を見つけ出すことができるのではないかと、という苦肉の策である。しかしこの苦肉の策が意外に良い結果を導いてくれるというのが、スパースモデリングと呼ばれる枠組みの成功の鍵を握る。

L_1 再構成

スパースな解を求めるために、 L_0 再構成の代わりに、以下の最小化問題を解く。

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (6)$$

ここで $\|\mathbf{x}\|_1$ を \mathbf{x} の L_1 ノルムと呼び、 \mathbf{x} の各成分の絶対値の和を表す。

なるほどそういうノリでよければ、 L_2 ノルムはどうだろうか？ L_2 ノルムは、各成分の2乗和を取り、ルートをとったものである。

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_N^2} \quad (7)$$

L_2 再構成？

スパースな解を求めるために、 L_0 再構成の代わりに、以下の最小化問題を解いたらどうだろうか？

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (8)$$

ここで $\|\mathbf{x}\|_2$ を \mathbf{x} の L_2 ノルムと呼び、 \mathbf{x} の各成分の二乗和をとり、ルートをとったものである。

それでは L_1 ノルムと L_2 ノルムによる最小化問題を比較しながら、スパースな解を求めることができるか、試してみよう。

例題 CS-I: L_1 ノルムと L_2 ノルムの比較

以下のふたつの最小化問題を解いてみよう。

$$\min_{x_1, x_2} \{|x_1| + |x_2|\} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (9)$$

$$\min_{x_1, x_2} \left\{ \sqrt{x_1^2 + x_2^2} \right\} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (10)$$

ここで観測行列 \mathbf{A} は、単なる行ベクトルに過ぎず、

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \end{pmatrix} \quad (11)$$

とする。 $y = 1$ とする。

さて、このまま最小化問題を解く前に、制約条件にされている $\mathbf{y} = \mathbf{A}\mathbf{x}$ から、

$$1 = 2x_1 + x_2 \quad (12)$$

であることに注目したい。 (x_1, x_2) の2次元座標空間上で、傾き -2 の直線を描くことがわかる。 \mathbf{x} を知るために観測行列 \mathbf{A} を作用させた結果、たったひとつの y を得た。その y から、元の \mathbf{x} を求めたいというのがやりたいことである。しかし得られた y は1次元であり、一方知りたいものは2次元であるので、この条件だけでは、どの1点を知りたいものの素性とするべきかは定まっていない状況というわけだ。まずは x_2 について解いてみると、

$$x_2 = -2x_1 + 1 \quad (13)$$

を得るので、この結果を最小化するべきふたつのノルムに組み入れてみよう。最小化（または最大化）され

る関数のことをコスト関数と呼ぶ。コスト関数が L_1 ノルムの場合は

$$|x_1| + |-2x_1 + 1| \quad (14)$$

である。絶対値記号を外すために、場合分けを駆使すると

$$|x_1| + |-2x_1 + 1| = \begin{cases} 3x_1 - 1 & (1/2 \leq x_1) \\ -x_1 + 1 & (0 \leq x_1 < 1/2) \\ -3x_1 - 1 & (x_1 < 0) \end{cases} \quad (15)$$

となることがわかる。最小値をとる (x_1, x_2) を求めてみると、 $(x_1, x_2) = (1/2, 0)$ であり、スパースな解を選ぶことに成功している。

一方 L_2 ノルムの場合は、最小化するべきコスト関数は

$$\sqrt{x_1^2 + (-2x_1 + 1)^2} = \sqrt{5 \left(x_1 - \frac{2}{5}\right)^2 + \frac{1}{5}} \quad (16)$$

となり、最小値をとる (x_1, x_2) を求めてみると、 $(x_1, x_2) = (2/5, 1/5)$ であり、スパースな解とはならないことがわかる。このように明確に最小化するべきコスト関数部分により、うまく解が選択されて、スパースな解を求めるには、 L_1 ノルムが有効であることがわかる。

ここで強調したいのは、制約条件だけでは定まらない、すなわち観測情報だけでは完全に解を定めることができない。しかしその中で、適切なコスト関数の利用により、スパースな解を選ぶことにしている。その解が正しいかどうかはまだ議論していないが、**スパースな解を選んでいる**ことに注意したい。少なくともスパースな解を得たいというとき、スパースな解が十分に期待される場面では強力な処方箋であることがわかる。

この事実は読み方を変えれば、劣決定系の問題が与えられたときに、色々な解がある中で積極的にスパースな解を選ぶことにより、その条件を満たすために重要な成分はどこなのか？ということに答えることができる。これがスパースモデリングのひとつの重要な技術となる**変数選択**である。

2.4 直観的な理解

上記の具体的な計算による慣れがあるなら、次の図の説明による理解が更に助けとなることだろう。2次元座標空間 x_1, x_2 上で、これまで扱ってきた例題 CS-I について幾何学的に考察してみよう。まず制約条件は $x_2 = -2x_1 + 1$ という直線を描く。 L_2 ノルムは、 $x_1^2 + x_2^2 = r^2$ とおけば、半径 r の円を描くことが分かる。制約条件を満たす原点を中心とする円を描くと、ちょうど円と直線が接するときであることがわかる(図1)。一方 L_1 ノルムは絶対値の場合分けを繰り返すことにより、 x_1, x_2 軸上に先端を持つ原点を中心とする菱形で表すことができる。制約条件を満たす菱形を描くと、同様に菱形と直線が接するときであることがわかる。なおかつ原点に近い先端を持つものが最小化問題の解である。(図2) この図を見ると明らかのように、 L_1 ノルムは尖っているために、スパースな解を出す能力に長けているということがわかる。そうかなるほど、尖っているならばそれで良いではないか、他のノルムはどうだろうか？ベクトルの L_p ノルムを以下のように定義する。

$$\|\mathbf{x}\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + \dots + |x_N|^p} \quad (17)$$

用語が割と混同されて使われることが多いので適宜定義に注意するべき量である。この p を $p \rightarrow 0$ と持っていった極限が上記で利用した L_0 ノルムである。しかし L_0 ノルムは、ノルムの性質(斉次性: $f(a\mathbf{x}) = a^k f(\mathbf{x})$)が満たされないため、やや違和感が残るが、圧縮センシングでは便宜のため、 L_0 「ノルム」と呼ぶ。さて $0 \leq p \leq 1$ であれば、図で描けば分かるように、「尖っている」ため、 L_1 ノルム同様にスパース解を得る能力は十分にある(図3)。

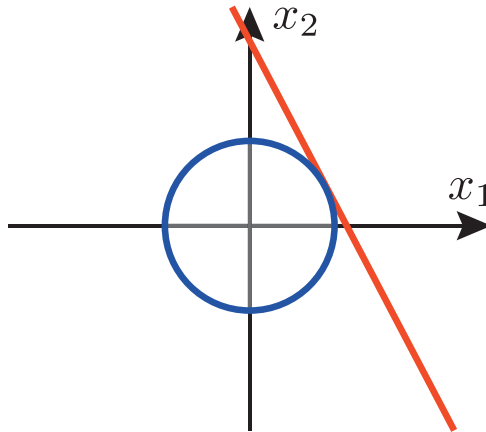


図 1: L_2 ノルム最小化の様子.

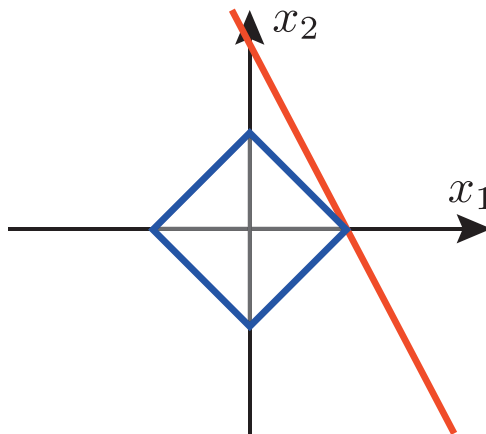


図 2: L_1 ノルム最小化の様子.

2.5 圧縮センシング

さてノルムの性質により、このような解選択の技術が得られることがわかった。そのときに問うべき問題は、本当にスパースな解が欲しいのか？本当の解はスパースな解なのか？である。

前者は、変数選択について焦点を絞ったものである。方程式の真の解に興味無く、少なくともその方程式を満たすことのできる数少ない変数の組み合わせは無いか？という探索を行う場合に重要な問いである。これは身に覚えがある人がいるだろう。色々あれやこれやと説明を繰り返されたときに、要するに何が重要な？という問いをしたくなるときがある。その**重要な部分を自動的に抽出する**のが変数選択である。これは多くの実験屋が求めている方策ではないだろうか。

次に後者は、変数選択というよりも真の解を求めるという立場に立ったときに、スパースな解が妥当であるかどうかについて考える必要がある。本来的にスパースな解を持つ方程式の問題に対して、スパース解を選択的に選び出す方法は絶大な効果を持つ。

圧縮センシング

知りたいものを表す N 次元の入力ベクトル \mathbf{x}_0 に対して、観測と称して線形変換を施す。その変換行列を A としたときに、 M 次元の出力ベクトル \mathbf{y} を得るとする。

$$\mathbf{y} = A\mathbf{x}_0 \quad (18)$$

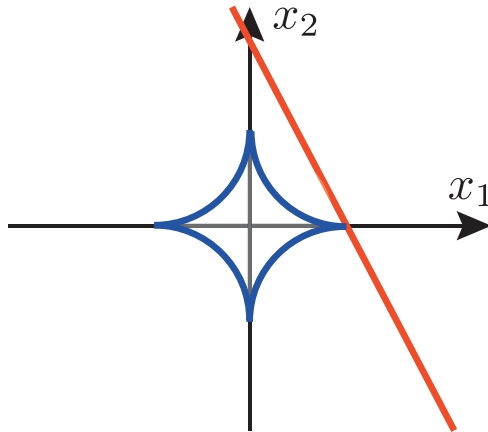


図 3: L_p ノルム最小化の様子.

このとき、知りたいもの \mathbf{x}_0 はスパース性を期待することができるとする. そのとき L_1 ノルム等を利用してスパースな解 \mathbf{x} を得るときに、ある条件のもとでは $\mathbf{x} = \mathbf{x}_0$ となることがある.

実際の実例を紹介しよう. $N = 1000$ 次元のうち $K = 20$ 個のみが正規分布に従うランダム変数を持つ原信号と正規分布に従うランダム変数を持った観測行列を適用して得られた $M = 100$ 次元の出力ベクトルを図 4 に示す. これを L_1 ノルム最小化によってスパースな解を得ると見事に入力ベクトルと一致する結果を

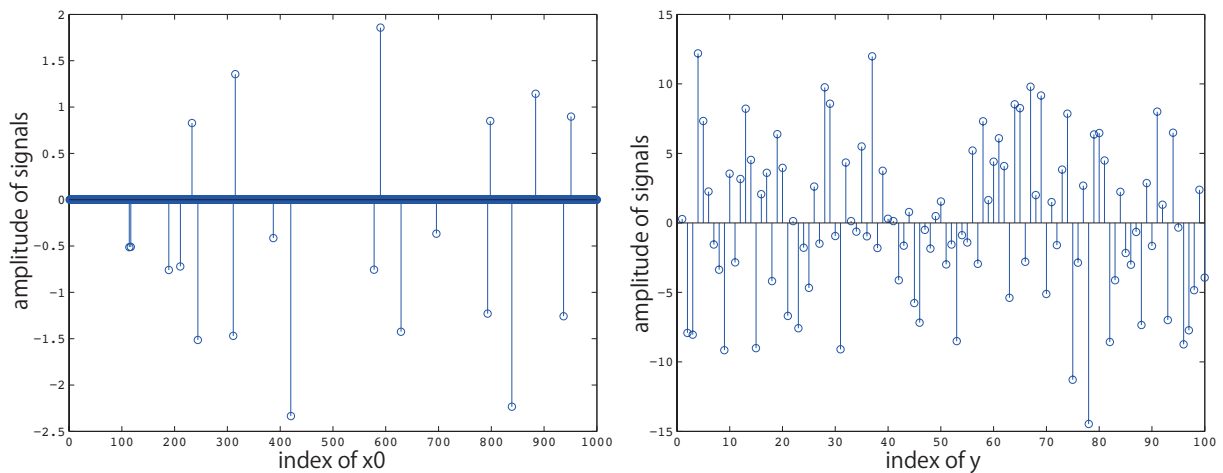


図 4: $N = 1000$ 次元のうち $K = 20$ 個の非零をもつ入力ベクトル (左) と $M = 100$ 次元の出力ベクトル (右).

得る. 一方 L_2 ノルムではそのような結果は得られない (図 5). ここで注意したいのは、圧縮センシングという枠組みは、劣決定系の方程式からスパースな解を得るノルムの性質を活かして、更に厳密に知りたいものを当てる. 情報科学の名探偵なのだ. 特に L_1 ノルムの最小化による原情報推定を**基底追跡 (Basis Pursuit)** と呼ぶ.

基底追跡 (Basis Pursuit)

以下の最小化問題を解くことにより、 M 次元の出力ベクトル \mathbf{y} からスパースな原情報 \mathbf{x}_0 を高い

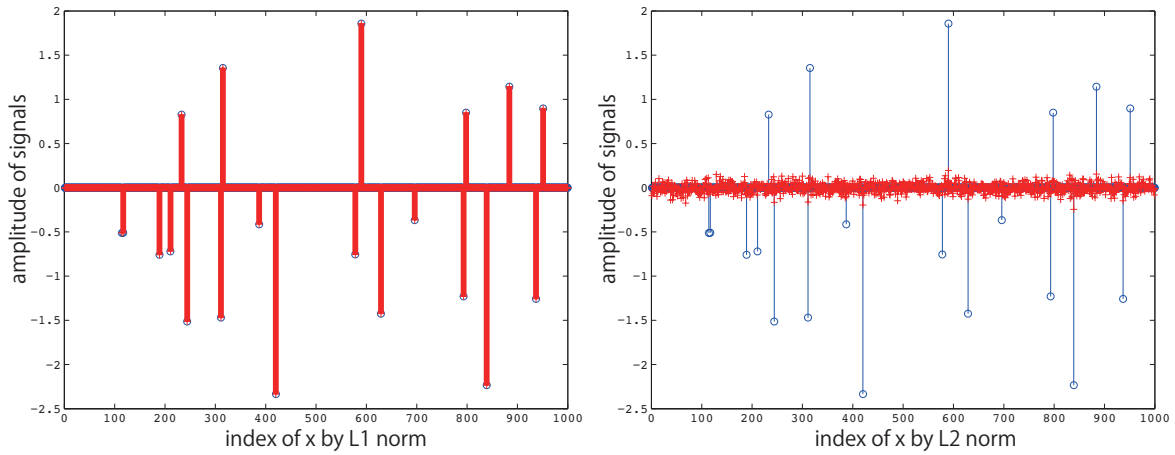


図 5: L_1 ノルムによる再構成結果 (左) と L_2 ノルムによる再構成結果 (右).

確率で得ることができる.

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (19)$$

例題 CS-II:原信号推定

スパースな原信号ベクトル $\mathbf{x}_0^T = (1/2, 0)$ に対して、 $A = (2, 1)$ を作用して $y = \mathbf{A}\mathbf{x}_0 = 1$ のみが得られているとする. 基底追跡により、スパース解を選択することにより原信号ベクトルを推定せよ.

例題 CS-I と解くべき最小化問題は同じである. そのため推定される解は $\mathbf{x}^T = (1/2, 0)$ である. ぴったりと原信号と一致しており、正解を当てていることがわかる. おお、すごいと単純に思ってしまう. スパースな原信号ベクトルが実は $\mathbf{x}_0^T = (0, 1)$ であった場合はどうだろうか? そのとき $A = (2, 1)$ を作用すると $y = \mathbf{A}\mathbf{x}_0 = 1$ が再び得られるため、推定される解は $\mathbf{x}^T = (1/2, 0)$ であるから、原信号と異なる. しかしながらそういう意地悪な例でなければ、とりあえず当てることができるようだ. この性質は一般に成立するのだろうか? その問いに答えるには積分幾何学 [1, 2] または情報統計力学 [3, 4] による解析を行う必要がある. とりあえずは解答をするに留める.

圧縮センシングの性質

観測行列 A が各成分平均 0、分散 1 のガウス分布から生成されるとき、以下の条件を満たす曲線を境にして、 α が大きく、 ρ が小さい領域では、 L_1 ノルム最小化により非常に高い確率で原信号の推定に成功する.

$$\frac{1}{\alpha} = 1 + \sqrt{\frac{\pi}{2}} t e^{\frac{t^2}{2}} \{1 - 2Q(t)\} \quad (20)$$

$$\frac{\rho}{1 - \rho} = 2 \left(\frac{e^{-\frac{t^2}{2}}}{t\sqrt{2\pi}} - Q(t) \right) \quad (21)$$

ここで $\alpha = M/N$ 、 $\rho = K/N$ であり、

$$Q(t) = \int_t^\infty \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} dx \quad (22)$$

である (図 6).

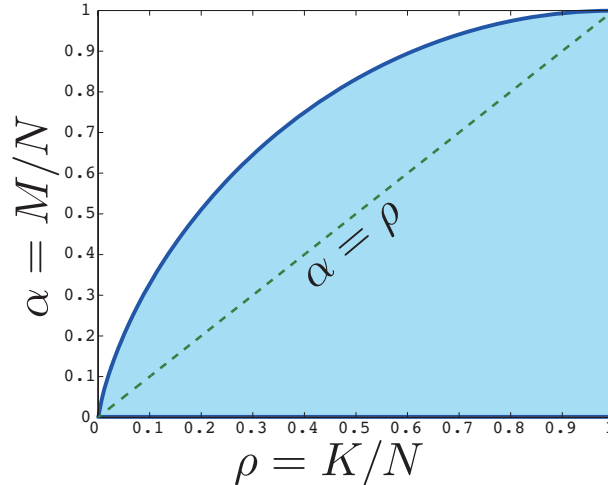


図 6: L_1 ノルム最小化による再構成が可能な領域と不可能な領域の相境界の様子。

この発見が圧縮センシングの隆盛のきっかけとなり、実際に MRI 画像等の応用例 [5] により爆発的な流行を迎えた。ただスパース解を選択するだけでなく、そこに正解を当てるという要素があり、適当なものを選ぶというわけではないところが重要である。そこでより観測の実態に根ざした現実的な状況において、圧縮センシングの枠組みを適用することを考えて行こう。

2.6 ノイズ有り圧縮センシング

観測にはノイズがつきものであり、これまで扱って来た等式制約が適切ではない場合がある。加法的ノイズが存在する状況で、線形変換で記述される観測を行った場合に得られる信号は、

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \sigma_0\mathbf{w} \quad (23)$$

と表すことができる。ここで \mathbf{w} は M 次元のノイズを表すベクトルであり、 σ_0 はノイズの強度を表す。ノイズの存在により、 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満たすものからスパースな解を選択するというのはやや無理がある。まずそもそも等式を満たすことができるとは限らないためだ。そこで \mathbf{y} と $\mathbf{A}\mathbf{x}$ のズレをできるだけ小さくしつつ、 L_1 ノルムを小さくするという方策が考えられる。そこで観測に加法的ノイズが入った場合には、以下の最小化問題を解くことでスパースな信号 \mathbf{x} を推定しようとする。

$$\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \right\} \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq a \quad (24)$$

これは Tibshirani によって提案され、LASSO (Least Absolute Shrinkage and Selection Operators) と呼ばれる [6]。不等式制約条件は、スパースな解を得るために導入したものであり、 a はスパース性を制御するパラメータである。コスト関数部分は、最小二乗法で用いられるものであり、観測結果との誤差を押さえつつ、スパース性を獲得した解を得るものと解釈できる。ラグランジュ未定乗数を持ちいて、以下の最小化問題に帰着させることができる。

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\} \quad (25)$$

$\lambda > 0$ はどの程度ズレを許容するかを表すパラメータ (ラグランジュ未定乗数) である。 $\lambda \rightarrow 0$ とすれば、少しでも \mathbf{y} と $\mathbf{A}\mathbf{x}$ の間にズレが生じるとコスト関数が非常に大きな値をとり、ズレに対して鋭敏な変化を示すようになる。少しのズレも許容しない。つまり等式制約 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を課した場合と同様となる。一方 λ を緩めて大きな値を使うと、多少のズレがあっても許容できるするという格好だ。ノイズの強度があまりに

大きくて、 \mathbf{y} が信用ならない場合には λ を大きめ、逆に \mathbf{y} を大事にしたい場合は λ を小さめにするという直観的な方策のもとパラメータを決定する。圧縮センシングを実際のデータに適用する場合には、式 (25) を解くことにより実践する。(また式 (25) を称して LASSO と呼ぶことも多い。)

2.7 ベイズ推定との関係

LASSO は素直に導入できる最小化問題であり、取得されたデータ \mathbf{y} と観測方法における詳細 A から原情報にあたる \mathbf{x} を推定する直観的なアルゴリズムといえる。もう少しだけ数理的背景を理解するために、入出力関係に基づいて得られた出力情報から背後にある入力関係を推定する確率的推論の枠組みについて紹介しよう。

ある入力情報にある変換を経たのち、出力情報が得られるときの関係を入出力関係と呼ぶ。このときに不確実な要素が取り込まれた場合に確率的推論の枠組みが必要となる。決定的な入出力関係であれば、それを表現するには関数が用いられる一方で、確率的な入出力関係を取り扱うためには、条件付き確率が用いられる。今の場合、入力情報 \mathbf{x} と観測の詳細 A が与えられたときに、出力情報 \mathbf{y} を得る確率であるから、 $P(\mathbf{y}|A, \mathbf{x})$ という条件付き確率を考える。条件付き確率に慣れ親しんでいない読者もいるだろう。この条件付き確率は、一般に以下の定義を持つ。

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A) \quad (26)$$

ここで $P(A, B)$ は A と B の事象が同時に起こる確率。 $P(A|B)$ は B が起こった上で、 A が起こる条件付き確率、 $P(B|A)$ は A が起こった上で、 B が起こる条件付き確率である。3つ以上の事象間の関係でも同様である。

$$P(A, B, C) = P(A|B, C)P(B, C) = P(A, B|C)P(C) \quad (27)$$

さてこの定義にもとづき、ベイズの定理を紹介しよう。

ベイズの定理

条件付き確率の間には、以下の関係が成立する。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(A, B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)} \quad (28)$$

ふたつめの等式では $P(B) = \sum_A P(A, B)$ となる事実を用いた。

ベイズの定理を用いると、 A と B の事象の順番が反転していることに注意したい。またベイズの定理を利用するときにはこの表式に拘る必要はない。結合確率と条件付き確率の定義から導けばよい。

このベイズの定理を用いて、出力情報 \mathbf{y} 、観測の詳細 A を知っているときに入力情報を**確率的に推定する**方法を考える。まずベイズの定理（結合確率と条件付き確率の関係）を機械的に用いれば、

$$P(A, \mathbf{x}, \mathbf{y}) = P(\mathbf{x}|A, \mathbf{y})P(A)P(\mathbf{y}) = P(\mathbf{y}|A, \mathbf{x})P(A)P(\mathbf{x}) \quad (29)$$

であるから、

$$P(\mathbf{x}|A, \mathbf{y}) = \frac{P(\mathbf{y}|A, \mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \quad (30)$$

となる。これを用いて入力情報 \mathbf{x} の推定値を計算すればよい。ここで少しか用語の説明をする。まず $P(\mathbf{y}|A, \mathbf{x})$ であるが、条件付き確率と同じ格好をしているが意味合いは異なる。 \mathbf{y} は既に得られているので、の実現値を代入するべきで、その実現値が A, \mathbf{x} に対してどれだけ尤もらしいかを表す**尤度関数**と呼ばれる。 A はもちろんのこと \mathbf{x} もどのようなものを入れるかにより、 \mathbf{y} の実現があり得るかあり得ないかは変わるはずで、その度合いを示している。さらに $P(\mathbf{x})$ は \mathbf{x} に対する事前確率と呼ばれており、 \mathbf{x} が選択される確率を示している（実現値が存在しないのでこちらは確率！）これらの積からできる左辺は、 $P(\mathbf{x}|A, \mathbf{y})$ は事

後確率と呼ばれる。 \mathbf{y} および A という実現値を持っているときに \mathbf{x} が選ばれる確率を示す。この事後確率にもとづき、確率的に \mathbf{x} を得ることで確率的推論を行う。後に述べるマルコフ連鎖モンテカルロ法などにより、サンプリング的に \mathbf{x} を求めることも可能であり、この事後確率の最大化により最頻値の \mathbf{x} を求めることも可能である。特にここでは**最大事後確率 (Maximum A Posteriori) 推定**を紹介する。最大事後確率推定は以下の最大化問題を解くことで実行できる。

$$\mathbf{x}_{\text{MAP}} = \arg \max_{\mathbf{x}} \{\log P(\mathbf{x}|A, \mathbf{y})\} \quad (31)$$

それでは圧縮センシングの問題において、この事後確率を求めてみよう。まず必要なのは、尤度関数 $P(\mathbf{y}|A, \mathbf{x})$ である。これは観測行列 A 、入力情報 \mathbf{x} が与えられたときに \mathbf{y} がどのような確率に従って出力されるかを考えれば良い。ノイズ有り圧縮センシングにおいては、式 (23) をみてもらうと、ノイズという不確実要素がある。この不確実要素であるノイズが、ガウス分布に従うと仮定する。

$$P(\mathbf{w}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \mathbf{w}^T \mathbf{w}\right) \quad (32)$$

A と \mathbf{x} からノイズを含んだ形での出力情報 \mathbf{y} を得る確率は、ノイズがどのような値をとるかの確率によってきまる。そこでノイズ \mathbf{w} について式 (23) を変形すると、

$$\mathbf{w} = \frac{1}{\sigma_0} (\mathbf{y} - A\mathbf{x}) \quad (33)$$

であるから、尤度関数は

$$P(\mathbf{y}|A, \mathbf{x}) = P(\mathbf{w}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_0^2} \|\mathbf{y} - A\mathbf{x}\|_2^2\right) \quad (34)$$

であることがわかる。次は事前確率である。 \mathbf{x} についてどんなものがあり得るかを導入することができる。例えば大きさがそこまで大きくならないとすれば、

$$P(\mathbf{x}) = \sqrt{\frac{\kappa}{2\pi}} \exp\left(-\frac{\kappa}{2} \|\mathbf{x}\|_2^2\right) \quad (35)$$

というものを設定しても良いかもしれない。このときに最大事後確率推定を行えば、解くべき最適化問題は、

$$\mathbf{x}_{\text{MAP}} = \arg \max_{\mathbf{x}} \left\{ -\frac{1}{2\sigma_0^2} \|\mathbf{y} - A\mathbf{x}\|_2^2 - \frac{\kappa}{2} \|\mathbf{x}\|_2^2 \right\} \quad (36)$$

となる。符号を変えれば、これは LASSO の L_1 ノルムの代わりに L_2 ノルムを利用したものとなっていることがわかる。そうなれば要領は得ただろう。事前確率として L_1 ノルムを用いた形

$$P(\mathbf{x}) \propto \exp(-\kappa \|\mathbf{x}\|_1) \quad (37)$$

を利用すれば、最大事後確率推定により、

$$\mathbf{x}_{\text{MAP}} = \arg \max_{\mathbf{x}} \left\{ -\frac{1}{2\sigma_0^2} \|\mathbf{y} - A\mathbf{x}\|_2^2 - \kappa \|\mathbf{x}\|_1 \right\} \quad (38)$$

$\lambda = \kappa\sigma_0^2$ とすれば、LASSO の最小化問題 (25) に帰着することが分かる。つまり圧縮センシングは、解の事前情報としてスパース性を採用した最大事後確率推定により原情報推定を行うことと理解することができる。

3 圧縮センシングの実践

ここまでくれば、圧縮センシングの心自体は大分理解できているものと思われる。出力情報と入力情報の関係性について注目しつつ、入力情報の事前知識にもとづいて解の選択を行う。実際に解がスパース性を持つものであれば、 L_1 ノルムによる解選択の性質を利用して正解に到達することができる。それでは次に実践編として、LASSO 型の最小化問題を解く方法について学んで行こう。

3.1 そもそもスパースな解ってなんだろう？

圧縮センシングは、解のスパース性に注目して劣決定系の連立方程式を解くことで、少ない観測数から重要な情報部分となる非零成分を推定する。その解のスパース性は、そもそもどれだけ期待できるものなのだろうか。その回答の典型的なものが、実際の画像圧縮に用いられるフーリエ変換やウェーブレット変換による疎性の獲得であろう。ここでウェーブレット変換をしたのちに、ウェーブレット係数の 95% を絶対値が小さい順から 0 にして、ウェーブレット逆変換を施して得られた画像をみてみよう (図 7)。見た目にはそれほど影響しないことが見て取れるであろう。このように \mathbf{x} そのものではなく、何らかの変換を経て、



図 7: 原画像と絶対値の小さいものから 95% のウェーブレット係数を 0 にしたものからの再構成結果の比較。

そして本当には零となっていないものの、本質的な部分だけに注目すれば、その部分だけを残すことで、スパース性を獲得することが期待される場合がある。そのようなスパース性を促す変換を求めることも、圧縮センシングの幅広い応用を可能とさせるため重要な発展に資する。このようにある線形変換 B をした先で、スパース性が期待される場合、解くべき最小化問題は、

$$\min_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \|B\mathbf{x}\|_1 \right\} \quad (39)$$

とすべきであろう。この場合は逆変換 B^{-1} が存在する場合、 $\mathbf{z} = B\mathbf{x}$ とおき、

$$\min_{\mathbf{z}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - AB^{-1}\mathbf{z}\|_2^2 + \|\mathbf{z}\|_1 \right\} \quad (40)$$

を代わりに解けば良いので、本質的に LASSO を解けば良いことに変わりはない。逆変換が非自明な場合は、このような単純な計算回避はできないが、計算アルゴリズムの工夫により回避する。

3.2 観測過程

次に観測行列 A についてである。そもそも線形観測で表現できる問題があるのだろうか。圧縮センシングは MRI を始めとする医療画像での応用例が豊富である。MRI ではプロトンの密度を測定するために、核磁気共鳴を利用して、核スピンの歳差運動から生じる電磁波を外部の受信コイルで誘導起電力として感受することにより情報取得を行う。この過程は実はフーリエ変換、つまり線形変換で記述することができる。そのため圧縮センシングの典型的な定式化に乗せることが容易い。本来は $M = N$ に及ぶ情報取得を行わなければならないところを、圧縮センシングの適用により、 $M < N$ として縮減することが可能であり高速撮像が実現する。スパースモデリングの利用例で挙げられる (NHK サイエンス ZERO でも取り上げられ

た) ブラックホール撮像の例では、電波望遠鏡で受光する情報がやはり同様にフーリエ変換で記述されるものである [7]。この場合は本質的に観測数が不足しているため、見られなかったものを見られるようにするテクノロジーとして圧縮センシングが利用されている。

これまでは実空間上で観測対象を順序よく観測していたものを、あえて変更して線形変換で書かれる観測過程にすることで圧縮センシングの適用を試みるものもある。その最たる例がシングルピクセルカメラである [9]。

また更に言えば、先ほど見たように圧縮センシングの背後にはベイズ推定があり、解くべき最小化問題は最大事後確率推定から定式化されることがわかっている。特に観測にかかるノイズがガウス分布に従うものと仮定したときに LASSO 型の最小化問題に帰着した。ガウス分布以外の特性を持つノイズの場合であっても、同様に最大事後確率推定から定式化することによって圧縮センシングの適用範囲を拡大することができる。非常に広い応用範囲があることが理解されよう。

3.3 再構成アルゴリズム

さてそれでは圧縮センシングの重要性や数理的背景をおさえたところで、実際に使うために、原情報推定のための再構成アルゴリズムを理解していこう。非常に多くのアルゴリズムが提案されているが、根本となる基本知識で十分に理解ができるものを選んで紹介する。

L_1 ノルムを含む最小化問題で一番厄介だったのは絶対値関数の扱いであろう。その絶対値関数部分を外すためには場合分けをしていたためだ。そこで絶対値関数があることの難しさについて、再び考えてみよう。最小化（または最大化）問題を解くときに、慣れ親しんだ方法はどんな手法があるか？微分を調べるといふものであろう。微分が 0 を取るとき、関数の変化が上昇から下降、または下降から上昇に転じるためだ。そのため最小値、最大値を調べるときには微分をとるとというのが常套手段であった。この性質を数値計算の方法として採用したものが勾配法である。

勾配法

以下で与えられる最小化問題を解くことを考える。

$$\min_{\mathbf{x}} \{f(\mathbf{x})\} \quad (41)$$

コスト関数 $f(\mathbf{x})$ が微分可能であるとき、その微分により、適当な初期点 $\mathbf{x}[0]$ から次の反復計算を行う。

$$\mathbf{x}[t+1] = \mathbf{x}[t] - \delta \nabla f(\mathbf{x}) \quad (42)$$

ここで δ は更新幅を決めるパラメータである。

ここでコスト関数は凸関数であるとする。凸関数である場合には単純な勾配法の適用と適切な更新幅によって、最適値に行き着くことが知られている。圧縮センシングで取り扱われる問題は、幸いなことに凸関数であるため、勾配法の適用で、最適値に行き着くことができそうだ。しかし L_1 ノルムを含むコスト関数の最小化をしようと考えると、絶対値関数は微分可能ではない関数のため、素朴な勾配法の適用ができない。これは困ったことだ。ただし以下で考察するように最小化問題を解くために必ずしも微分を必要とはしないことに注意しよう。

例題 CS-III: 絶対値関数のある最小化問題

以下の最小化問題を解け。

$$\min_{x} \left\{ |x| + \frac{1}{2\lambda} (y-x)^2 \right\} \quad (43)$$

ここで y は適当な実数、 $\lambda > 0$ である。

絶対値関数があるときは、場合分けをすればよい。まず $x > 0$ のときを考えてみよう。

$$\min_{x>0} \left\{ x + \frac{1}{2\lambda} (y-x)^2 \right\} \quad (44)$$

これは簡単な平方完成で解くことができる。まずコスト関数は、以下のように変形することができる。

$$\frac{1}{2\lambda} \{x - (y - \lambda)\}^2 + y - \frac{\lambda}{2} \quad (45)$$

頂点の位置から $x = y - \lambda$ が最小値をとるところであるといいたい。しかし $x > 0$ という条件に注意しなければならない。 $y - \lambda > 0$ のときは、頂点を採用して、一方、 $y - \lambda \leq 0$ のときは、放物線が頂点から離れるほど単調増加であることから、 x の定義域のなかで頂点に最も近い $x = 0$ が最小値を与える (図 8)。 よって、 $x \leq 0$ の場合も同様にして最小値 x^* は、

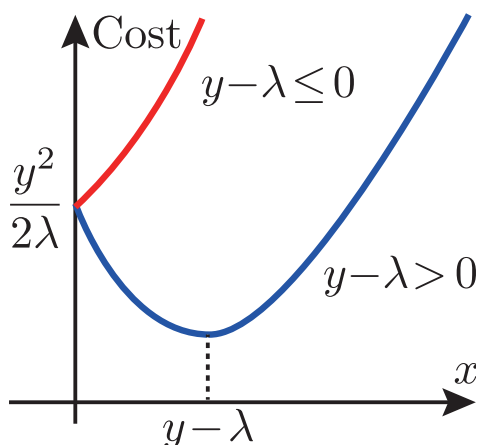


図 8: $x > 0$ に限った場合のコスト関数の振る舞い。

$$x^* = S_\lambda(y) \quad (46)$$

となる。

[問：確認せよ]

ここで

$$S_\lambda(y) = \begin{cases} y - \lambda & (y > \lambda) \\ 0 & (-\lambda \leq y \leq \lambda) \\ y + \lambda & (y < -\lambda) \end{cases} \quad (47)$$

とした。 $S_\lambda(y)$ を軟判定しきい値関数 (Soft thresholding function) と呼ぶ (図 9)。 このように絶対値関数でも、1次元であれば頑張って場合分けをして最小化問題をとくことはできる。多次元であっても、以下のような状況であれば解くことができる。

例題 CS-II: L_1 ノルムのある最小化問題

以下の最小化問題を解け。

$$\min_{\mathbf{x}} \left\{ |\mathbf{x}| + \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{x}\|_2^2 \right\} \quad (48)$$

ここで λ, \mathbf{y} は適当な実数である。

L_1 ノルムも、 L_2 ノルムも**分離性**を持つ。分離性とは、各成分独立に計算が実行できる形をもっていることである。実際、 L_1 ノルムはその定義から

$$\|\mathbf{x}\|_1 = \|x_1\| + \|x_2\| + \dots + \|x_N\| \quad (49)$$

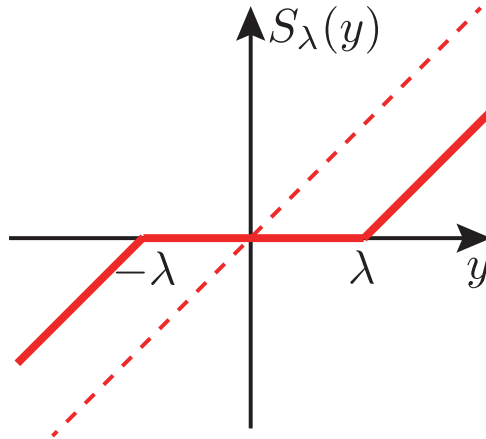


図 9: 軟判定しきい値関数の振る舞い.

と各成分の絶対値の和であり、 L_2 ノルムも、2 乗すれば、

$$\|\mathbf{x}\|_2^2 = x_1^2 + x_2^2 + \cdots + x_N^2 \quad (50)$$

と各成分の 2 乗和であるので、各成分毎に解くことが許される。よって最小値は、各成分毎に軟判定しきい値関数を適用することで、

$$x_i^* = S_\lambda(y_i) \quad (51)$$

となる。これを以下のように簡単に表記することにする。

$$\mathbf{x}^* = S_\lambda(\mathbf{y}). \quad (52)$$

しかしながら圧縮センシングの問題では、解きたい最小化問題はもう一步だけ込み入っている。行列による線形変換を経ているため、ここまで単純ではない。だが絶対値関数があるからといって諦める必要はなく、ちょっと考えてみる価値がありそうである。

Fast Iterative Shrinkage Thresholding Algorithm (FISTA) 行列による線形変換を経て、 L_2 ノルム部分がやや込み入っていることが問題であるなら、その部分を変化させることを目標としたのが Iterative Shrinkage Thresholding Algorithm (ISTA) ないしその高速版である FISTA である [8].

その基本原理は、**メジャライザー最小化 (Majorizer Minimization)** と呼ばれる手法である。

メジャライザー最小化

微分がリプシッツ定数 L のリプシッツ連続である凸関数 $g(\mathbf{x})$ に対して、以下の関数を定義する。

$$q_L(\mathbf{x}, \mathbf{y}) = g(\mathbf{y}) + (\nabla g(\mathbf{y}))^\top (\mathbf{x} - \mathbf{y}) + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \quad (53)$$

このとき $g(\mathbf{x}) \leq q_L(\mathbf{x}, \mathbf{y})$ が成立する。この $q_L(\mathbf{x}, \mathbf{y})$ を $g(\mathbf{x})$ の**メジャライザー**と呼ぶ。元の関数 $g(\mathbf{x})$ による最小化の代わりに、メジャライザーの逐次最小化を考える手法をメジャライザー最小化と呼ぶ。

まずリプシッツ連続という慣れない言葉が出てきた。ここでそのリプシッツ連続の定義を紹介する。

リプシッツ連続

関数 $f(\mathbf{x})$ がリプシッツ定数 L のリプシッツ連続であるとき

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2 \quad (54)$$

を満たす。

右辺がユークリッド距離であることを考慮して、イメージをすると、遠くなればなるほど関数の値は左辺のように確かにそれなりに変化するが、それほど急激な変化を起こすわけではなく、ユークリッド距離程度で押しえられるということを示している。ここである凸関数 $g(\mathbf{x})$ の微分がリプシッツ定数 L のリプシッツ連続であるということは、

$$\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2 \quad (55)$$

を満たす。このときメジャライザー $q_L(\mathbf{x}, \mathbf{y})$ は、平方完成から $q_L(\mathbf{x}, \mathbf{y}) \leq g(\mathbf{y})$ を満たすことがわかる

$$q_L(\mathbf{x}, \mathbf{y}) = g(\mathbf{y}) - \frac{1}{2L} \|\nabla g(\mathbf{y})\|_2^2 + \frac{L}{2} \left\| \mathbf{x} - \left(\mathbf{y} - \frac{1}{L} \nabla g(\mathbf{y}) \right) \right\|_2^2 \quad (56)$$

メジャライザーの最小値は、必ず $g(\mathbf{y})$ を下回る。 等号成立条件は $\nabla g(\mathbf{y})$ が $\mathbf{0}$ となるときである。またこのメジャライザーの平方完成による表式をみると、**ベクトルの L_2 ノルムで書くことができる** ことがわかる。ここで思い出してほしい。絶対値関数が出てきたとしても、最小化問題は分離性があれば解くことがそこまで難しくない。このベクトルの L_2 ノルムで表現されるというメジャライザーの性質を利用して、圧縮センシングに登場する最小化問題を解くことを目指す。

逐次的にアルゴリズムを実行することを想定して、 t ステップ時に得られた解を $\mathbf{x}[t]$ とするとき、メジャライザーの最小値を求めると、

$$\mathbf{x}[t+1] = \arg \min_{\mathbf{x}} q_L(\mathbf{x}, \mathbf{x}[t]) \quad (57)$$

このときメジャライザーの性質から、以下の関係を得る。

$$g(\mathbf{x}[t+1]) \leq q_L(\mathbf{x}[t+1], \mathbf{x}[t]) \leq g(\mathbf{x}[t]) \quad (58)$$

が成立している。よって逐次的にこれを繰り返すことにより最小化問題の最適解へと到達することを目指す。リプシッツ連続の条件は区分的であってもよいので、全領域に渡って満たす必要は必ずしもない。メジャライザーが最小値をとるところまでの区間についてリプシッツ連続の条件が満たされており、メジャライザーの性質である $g(\mathbf{x}) \leq q_L(\mathbf{x}, \mathbf{x}[t])$ が成立していればよい (図 10)。

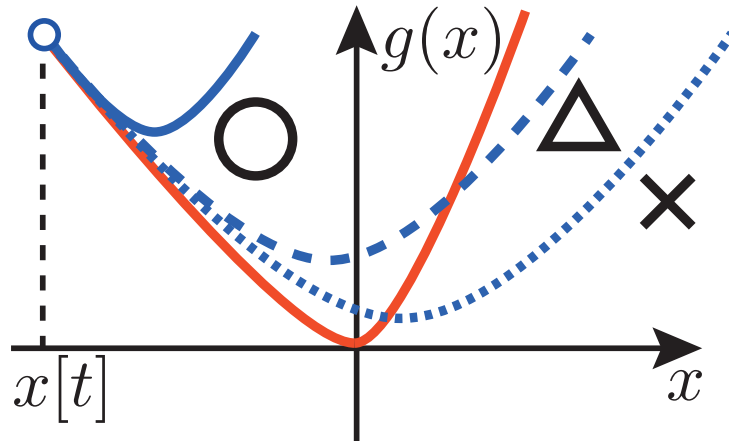


図 10: メジャライザーの振る舞い。赤線が $g(\mathbf{x})$ 。青線が全領域についてメジャライザーとなる場合、青破線が最小値を取るところまでメジャライザーとなる場合、青点線は最小値をとるところですらメジャライザーとはなり得ない。

さて解きたい問題は、式 (25) にある最小化問題である。ここで $g(\mathbf{x}) = \|\mathbf{y} - A\mathbf{x}\|_2^2 / 2\lambda$ 、 $\nabla g(\mathbf{x}) = -A^T(\mathbf{y} - A\mathbf{x}) / \lambda$ として、メジャライザーを構成する。このときリプシッツ定数は $L = \|A^T A\|_2 / \lambda$ であることが定義

に従った計算により分かる.

[問：リブシッツ定数を確認せよ.]

このように L_1 ノルムを取り入れても、メジャライザーの性質である上から押さえる $g(\mathbf{x}) + \|\mathbf{x}\|_1 \leq q_L(\mathbf{x}, \mathbf{x}[t]) + \|\mathbf{x}\|_1$ が保存されていることに注目する. そこで逐次的に解く最小化問題を

$$\mathbf{x}[t+1] = \arg \min_{\mathbf{x}} \{q_L(\mathbf{x}, \mathbf{x}[t]) + \|\mathbf{x}\|_1\} \quad (59)$$

として、 L_1 ノルムによる効果を取り入れて更新をしていくことで求めたい最小解へと目指す. ここで逐次的に解く最小化問題が手で以下のように解けることを利用する.

$$\mathbf{x}[t+1] = \arg \min_{\mathbf{x}} \left\{ \frac{L}{2} \left\{ \mathbf{x} - \left(\mathbf{x}[t] + \frac{1}{L\lambda} A^T(\mathbf{y} - A\mathbf{x}[t]) \right) \right\}^2 + \|\mathbf{x}\|_1 \right\} \quad (60)$$

$$= S_{1/L} \left(\mathbf{x}[t] + \frac{1}{L\lambda} A^T(\mathbf{y} - A\mathbf{x}[t]) \right) \quad (61)$$

つまり逐次代入をするだけで、最小解へ到達することができる.

Iterative Shrinkage Soft-thresholding Algorithm (ISTA)

1. $t = 0$ とする. 初期化 $\mathbf{x}[0]$. (例えば $\mathbf{x}[0] = A^T \mathbf{y}$)
2. 平方完成により $g(\mathbf{x})$ の 2 次関数近似の頂点を求める.

$$\mathbf{v}[t] = \mathbf{x}[t] + \frac{1}{L\lambda} A^T(\mathbf{y} - A\mathbf{x}[t]) \quad (62)$$

3. 軟判定しきい値関数を適用する.

$$\mathbf{x}[t+1] = S_{1/L}(\mathbf{v}[t]) \quad (63)$$

4. 終了基準を満たすまでステップ 2-4 を繰り返す.

更新する際に過去の情報を利用することで、収束までの反復数を減らすことで、より高速な FISTA についても提案されている.

Fast Iterative Shrinkage Soft-thresholding Algorithm (FISTA)

1. $t = 1$ とする. 初期化 $\mathbf{x}[0]$, $\beta[1] = 0$, $\mathbf{z}[1] = \mathbf{x}[0]$
2. 平方完成により $g(\mathbf{x})$ の 2 次関数近似の頂点を求める.

$$\mathbf{v}[t] = \mathbf{z}[t] + \frac{1}{L\lambda} A^T(\mathbf{y} - A\mathbf{z}[t]) \quad (64)$$

3. 軟判定しきい値関数を適用する.

$$\mathbf{x}[t+1] = S_{1/L}(\mathbf{v}[t]) \quad (65)$$

4. [高速化部分] $\beta[t]$ を更新する.

$$\beta[t+1] = \frac{1}{2} \left(1 + \sqrt{1 + 4(\beta[t])^2} \right) \quad (66)$$

5. [高速化部分] $\mathbf{z}[t]$ を更新する.

$$\mathbf{z}[t+1] = \mathbf{x}[t+1] + \frac{\beta[t] - 1}{\beta[t+1]} (\mathbf{x}[t+1] - \mathbf{x}[t]) \quad (67)$$

6. 終了基準を満たすまでステップ 2-6 を繰り返す.

FISTA は導出を見たあとに実装を試みればわかるように、非常に簡単な作りをしているため、理解もしやすくメジャライザー最小化の枠組みと共に L_1 ノルムを伴う最小化問題の入り口として最適である。さて FISTA には以下のような別解釈も存在する。後に詳しく述べるが、制約条件を 2 次の罰金項として追加する罰金法というのがある。この罰金法を利用すると、FISTA で利用したメジャライザー最小化に次のような別解釈ができる。

- コスト関数に罰金項 $L \|\mathbf{x} - \mathbf{x}[t]\|_2^2 / 2$ の導入により、暫定解 $\mathbf{x}[t]$ の近傍にある解を探索する。
- 暫定解の近傍で最小化を行うので、コスト関数の一部 $g(\mathbf{x})$ を以下のように近似することができる。

$$g(\mathbf{x}) \approx g(\mathbf{x}[t]) + (\nabla g(\mathbf{x}))^T (\mathbf{x} - \mathbf{x}[t]) \quad (68)$$

このふたつの操作により、得られるのがメジャライザー最小化の方法ともいえる。(但し罰金項の係数については非自明。)

Bregman 反復法 (Bregman iterative method) ノイズがある場合に LASSO 型の最小化問題はそれなりに意味のあるものであることはベイズ推定の観点から理解できる一方で、ノイズが全くない場合には非常に小さい λ による LASSO 型の最小化問題を解いても、 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満たすものを得るのは不可能である。仮に $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満たす解のなかで、LASSO 型の最小化問題と基底追跡型の最小化問題の両者の解になるものを探すと $\mathbf{x} = \mathbf{0}$ という自明な解を得ることがわかる。そこで本質的にノイズのない問題を取り扱う場合に、基底追跡問題を解く際に LASSO 型の最小化問題を經由して最適解へ到達することを可能にするのが以下で紹介する Bregman 反復法である。

次の Bregman divergence (以下 BD) を、凸関数 $f(\mathbf{x})$ に対して定義する。

$$D_f^{\mathbf{p}}(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - f(\mathbf{y}) - \mathbf{p}^T(\mathbf{x} - \mathbf{y}) \quad (69)$$

ここで、 \mathbf{p} は関数 $f(\mathbf{x})$ の劣勾配とする。劣勾配については聞き慣れないひともいるだろう。まずは劣勾配の定義の為に劣微分を定義する。

$$\partial f(\mathbf{x}) := \{\mathbf{p} | f(\mathbf{u}) \geq f(\mathbf{x}) + \mathbf{p}^T(\mathbf{u} - \mathbf{x}) \forall \mathbf{u}\} \quad (70)$$

劣勾配は、この劣微分のなかの特定の実現ベクトルを指す。任意の \mathbf{u} について、という条件から、(非常に小さい \mathbf{u} を考えていわゆる微分を行う) 微分可能な関数の場合には劣勾配は唯一に定まる。一方で微分可能でない場合は、劣微分は集合となる。要するにちゃんと決められない (図 11)。Bregman 反復法では、逆にこの性質を用いて、この劣微分を順次定めていきながら、解きたい最小化問題の解へと近づくことを目指す。

[問：BD が非負であることを確認せよ。]

この BD を用いるとどんなことができるだろうか。まず L_1 ノルムに現れる絶対値関数は、微分可能でない関数の代表例である。このときに BD を計算してみよう。 $y < 0$ のときに BD を調べてみると、

$$D_f^{-1}(x, y) = |x| - |y| + (x - y) = |x| + x = \begin{cases} 0 & (x < 0) \\ 2x & (x \geq 0) \end{cases} \quad (71)$$

となり、0 から急激に増大するようになる (図 12)。 $y > 0$ のときも同様に、

$$D_f^1(x, y) = |x| - |y| - (x - y) = |x| - x = \begin{cases} -2x & (x < 0) \\ 0 & (x \geq 0) \end{cases} \quad (72)$$

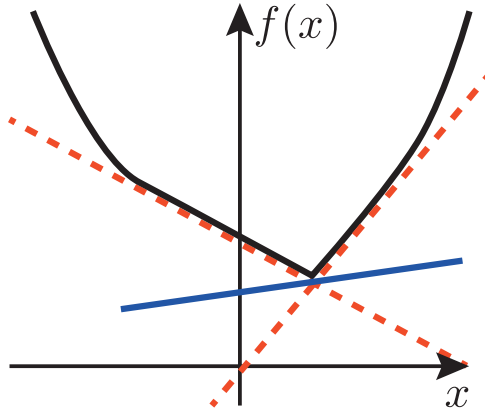


図 11: 劣勾配の様子. 微分可能でない関数は、要するに尖っているところで勾配（青線）を一意に定めることができない. 赤点線で囲まれた領域内を尖っているところを中心として回転する.

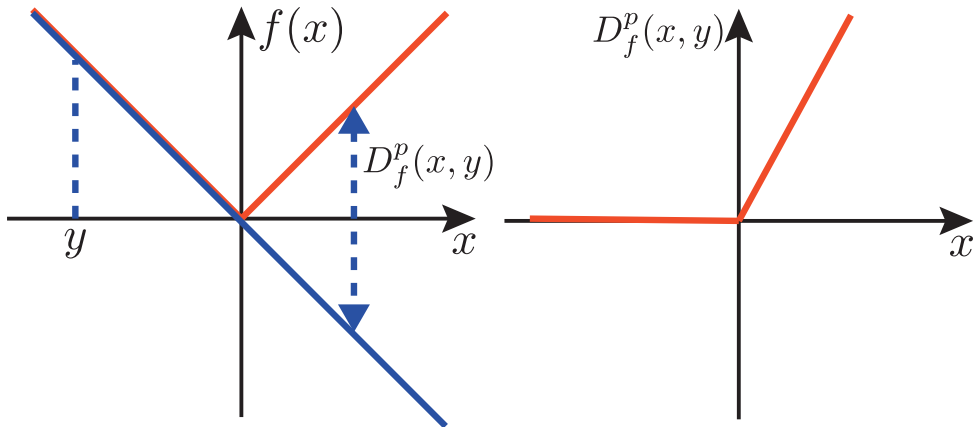


図 12: 絶対値関数に対する BD. $y < 0$ のとき.

となる. こちらも符号が変わるところで急激に増大することがわかる. $y = 0$ のところでは劣勾配の値を適当に仮定した上で、

$$D_f^p(x, y) = |x| - |y| - p(x - y) = |x| - px = \begin{cases} -(1+p)x & (x < 0) \\ (1-p)x & (x \geq 0) \end{cases} \quad (73)$$

少し傾いた形に絶対値関数に変更されていることがわかる (図 13). この観察から、BD は起点となる y における勾配で定まる一次関数からのズレが顕著になるとおおきくなることがわかる. この勾配は、最適化問題においては最適値を求めるための更新時の方向と距離を決める、いわば勢いである. L_1 ノルムの最小化の代わりにこの BD を用いるとどうなるだろうか. 勾配の指定するまま進むときに、元の関数からの乖離が激しくなるとペナルティとして BD が大きくなることがわかる. たとえば絶対値関数の場合、 $y < 0$ から始めると勾配 -1 で最小化の場合は正の方向に更新される. これは絶対値関数の最小化には有効である. しかし一旦 $x = 0$ を跨ぐと最小化にはならないので、BD の効果で勾配を変更するようにアルゴリズムを組めばよいだろう.

さて、上記の考察を経て、以下の最適化問題を解くことを考えよう.

$$\min_{\mathbf{x}} \{f(\mathbf{x}) + g(\mathbf{x})\} \quad (74)$$

ここで $g(\mathbf{x})$ は微分可能な関数であり、ノイズ有り圧縮センシングの問題設定では $\|\mathbf{y} - A\mathbf{x}\|_2^2 / 2\lambda$ のこと、 $f(\mathbf{x})$ は微分可能でない関数であり、 $\|\mathbf{x}\|_1$ のことだと思って良い. Bregman 反復法では、上記の最適化問題

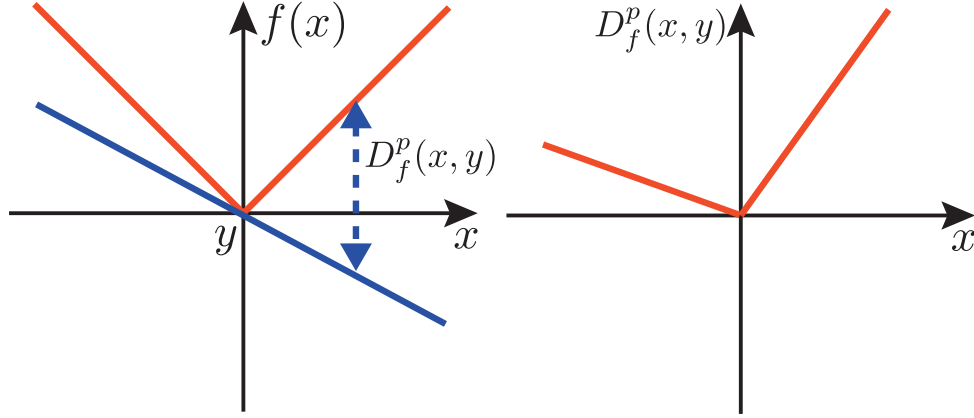


図 13: 絶対値関数に対する BD. $y = 0$ のとき.

の代わりに、

$$\min_{\mathbf{x}} \left\{ D_f^{\mathbf{p}[t]}(\mathbf{x}, \mathbf{x}[t]) + g(\mathbf{x}) \right\} \quad (75)$$

を逐次的に解く. 初期条件は $\mathbf{x}[0] = \mathbf{0}$ 及び $\mathbf{p}[0] = \mathbf{0}$ とする. こうすることで, LASSO 型の最小化問題から出発する. ここで $\mathbf{x}[t]$ は前回までの更新によって得られた値であり, 更新は上記の最適化問題を解くことにより、

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ D_f^{\mathbf{p}[t]}(\mathbf{x}, \mathbf{x}[t]) + g(\mathbf{x}) \right\} \quad (76)$$

とする. 一方, $\mathbf{p}[t]$ は前回の更新結果から以下のように定める.

$$\mathbf{p}[t] = \mathbf{p}[t-1] - \nabla g(\mathbf{x}[t]) \in \partial f(\mathbf{x}[t]) \quad (77)$$

なぜこのように定めるか? 最適化問題を解くということは, 微分が 0 となる場所を探せばよい. そこで最小化したいコスト関数の (76) の微分を調べてみると, $\mathbf{p} - \mathbf{p}[t] + \nabla g(\mathbf{x}[t])$ となることがわかる. ここで \mathbf{p} は劣微分 $\partial f(\mathbf{x})$ の劣勾配である. \mathbf{p} をいくつにするべきか迷うところが, 微分を 0 にしたいという最適性の要請と, 今回の更新で $\mathbf{x} = \mathbf{x}[t+1]$ とするので、

$$\mathbf{p}[t+1] = \mathbf{p}[t] - \nabla g(\mathbf{x}[t+1]) \in \partial f(\mathbf{x}[t+1]) \quad (78)$$

とすることにしようというわけだ.

このような更新則を選ぶと, 最適性を満たしつつ更新の度に $g(\mathbf{x})$ の実現値が減少していくことが分かる.

$$g(\mathbf{x}[t+1]) \leq g(\mathbf{x}[t+1]) + D_f^{\mathbf{p}[t]}(\mathbf{x}[t+1], \mathbf{x}[t]) \quad (79)$$

$$\leq g(\mathbf{x}[t]) + D_f^{\mathbf{p}[t]}(\mathbf{x}[t], \mathbf{x}[t]) \quad (80)$$

$$\leq g(\mathbf{x}[t]) \quad (81)$$

1 行目では BD の非負性を用いた. 2 行目では 1 行目右辺の量が最小値であることを用いた. 3 行目は BD の自明な性質を用いた. この性質により \mathbf{y} と $A\mathbf{x}$ を近づけることを優先する. LASSO 型の最小化問題に適用した場合, 劣勾配の更新は、

$$\mathbf{p}[t+1] = \mathbf{p}[t] + \frac{1}{\lambda} A^T(\mathbf{y} - A\mathbf{x}[t]) \in [-1, 1] \quad (82)$$

となる. この更新則を採用する場合, 一旦 $\mathbf{y} = A\mathbf{x}[t]$ となる解を到達したとき, それが L_1 ノルムの最小解を与えることを示すことができる [10]. 問題は最小化問題 (76) を解くところである. ここはメジャライザーを用意して ISTA や FISTA と同様の手続きによって最小化問題を解けば良い. (他の方法で解いてもよい.)

圧縮センシングの問題での具体的なアルゴリズムを以下に紹介しよう。

Bregman 反復法

1. $t = 0$ とする. 初期化 $\mathbf{x}[0] = \mathbf{0}$, $\mathbf{p}[0] = \mathbf{0}$
2. 劣勾配 $\mathbf{p}[t]$ を用いて、

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 - \mathbf{p}[t]^T (\mathbf{x} - \mathbf{x}[t]) + \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 \right\} \quad (83)$$

を解く. これは、サブルーチンとして、

$$\mathbf{v}[s] = \mathbf{x}[s] + \frac{1}{L} \left(\frac{1}{\lambda} A^T (\mathbf{y} - A\mathbf{x}[s]) + \mathbf{p}[t] \right) \quad (84)$$

を用意して、軟判定しきい値関数を適用する

$$\mathbf{x}[s+1] = S_{1/L}(\mathbf{v}[s]) \quad (85)$$

で解いてよい. (ISTA 同様、FISTA での加速法を利用しても良い.)

3. 劣勾配の更新をする.

$$\mathbf{p}[t+1] = \mathbf{p}[t] + \frac{1}{\lambda} A^T (\mathbf{y} - A\mathbf{x}[t]) \in [-1, 1] \quad (86)$$

4. 終了基準を満たすまでステップ 2-4 を繰り返す.

Bregman 反復法の内部で解かれる最適化問題 (76) について、メジャライザー最小化の方法を適用してみよう.

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ D_f^{\mathbf{p}[t]}(\mathbf{x}, \mathbf{x}[t]) + (\nabla g(\mathbf{x}[t]))^T (\mathbf{x} - \mathbf{x}[t]) + \frac{L}{2} \|\mathbf{x} - \mathbf{x}[t]\|_2^2 \right\} \quad (87)$$

ここで L は前述のリプシッツ定数である. このとき劣勾配の更新を以下のように変更する.

$$\mathbf{p}[t+1] = \mathbf{p}[t] - \nabla g(\mathbf{x}[t]) - L(\mathbf{x} - \mathbf{x}[t]) \in \partial f(\mathbf{x}[t+1]) \quad (88)$$

これを**線形化された Bregman 反復法**と呼ぶ [10]. 圧縮センシングにおける LASSO 型の最小化問題においては、

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 - \left(\frac{1}{\lambda} A^T (\mathbf{y} - A\mathbf{x}[t]) + \mathbf{p}[t] \right)^T (\mathbf{x} - \mathbf{x}[t]) + \frac{L}{2} \|\mathbf{x} - \mathbf{x}[t]\|_2^2 \right\} \quad (89)$$

となる. 但し Bregman 反復法そのものと同等の性能を示す L の値に設定しないと期待された性能を発揮しないので注意しよう.

この方法を FISTA と比較してみよう. LASSO 型の最小化問題に対して、メジャライザー最小化による方法を採用して、以下の最小化問題を逐次的に解くものが FISTA であった.

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ \|\mathbf{x}\|_1 - \left(\frac{1}{\lambda} A^T (\mathbf{y} - A\mathbf{x}[t]) \right)^T (\mathbf{x} - \mathbf{x}[t]) + \frac{L}{2} \|\mathbf{x} - \mathbf{x}[t]\|_2^2 \right\} \quad (90)$$

一次の項に注目すると劣勾配の存在の分が異なる. この項に注目してみると FISTA は $\mathbf{x}[t]$ の近傍を罰金項を導入して探索するものと解釈したが更に 1 次の項が加わり、同様に $(\mathbf{p}[t])^T (\mathbf{x} - \mathbf{x}[t])$ という形、すなわちラグランジュ未定乗数を利用して、制約条件 $\|\mathbf{x} - \mathbf{x}[t]\|_1 < \epsilon$ を課していることがわかる. つまり Bregman 反復法はラグランジュ未定乗数と罰金項の導入により、暫定解の近傍点を探索する方法であることがわかる.

拡張 Lagrangian 法 (Augmented Lagrangian method) このようにラグランジュ未定乗数と罰金項の両者を導入した最適化手法を拡張ラグランジュ法と呼ぶ。拡張 Lagrangian 法は、制約付きの最適化問題を解く方法として最近提案された強力な方法である。まず制約付きの最適化問題を解く基本について押さえておこう。以下の制約付き最適化問題を考える。

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad c_i(\mathbf{x}) = 0 \quad (i = 1, 2, \dots, m) \quad (91)$$

ここでラグランジュの未定乗数法により、制約条件を組み入れた新しいコスト関数を定義する。

$$L(\mathbf{x}; \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m h_i c_i(\mathbf{x}) \quad (92)$$

この新しいコスト関数について改めて最小化を考えるとというのがラグランジュ未定乗数法である。例題を解きながら、考えてみよう。

例題 AL-I: 制約付き最小化問題

以下の制約付き最小化問題を考える。

$$\min_{\mathbf{x}} \{x_1^2 + x_2^2\} \quad \text{s.t.} \quad ax_1 + bx_2 - c = 0 \quad (93)$$

ここで a, b, c は適当な実数であるとする。つまり直線上でしか動けない点 (x_1, x_2) があつたときに、原点からの距離が最小となる点を探せという問題である。解は原点から直線へ垂線を下ろせば直ちに分かる問題だ。

ラグランジュ未定乗数法により、新しいコスト関数を定義する。

$$L(x_1, x_2; h) = x_1^2 + x_2^2 + h(ax_1 + bx_2 - c) \quad (94)$$

元の最小化問題を代わりに、このコスト関数の最小化を考えよう。そのためには各変数による微分を 0 にする解を求めればよい。

$$\frac{\partial}{\partial x_1} L(x_1, x_2; \lambda) = 2x_1 + \lambda a = 0 \quad (95)$$

$$\frac{\partial}{\partial x_2} L(x_1, x_2; \lambda) = 2x_2 + \lambda b = 0 \quad (96)$$

$$\frac{\partial}{\partial h} L(x_1, x_2; h) = ax_1 + bx_2 - c = 0 \quad (97)$$

まず x_1, x_2 に関する微分から未定乗数 h に応じて、

$$x_1 = -\frac{ha}{2} \quad (98)$$

$$x_2 = -\frac{hb}{2} \quad (99)$$

となることがわかる。この解を用いて、未定乗数についての最適化問題を解く。このとき最小化問題から、最大化問題へと変化していることに注意したい。

$$L(x_1, x_2; h) = -(a^2 + b^2) \frac{h^2}{4} - hc = -\frac{a^2 + b^2}{4} \left(h + \frac{2c}{a^2 + b^2} \right)^2 + \frac{2c^2}{a^2 + b^2}. \quad (100)$$

この最大化問題の解は、 $h = -2c/(a^2 + b^2)$ が最適解であるとわかり、 $x_1 = ac/(a^2 + b^2)$ 、 $x_2 = bc/(a^2 + b^2)$ であるとわかる。確かに垂線となっていることも確認できる。このようにラグランジュ未定乗数法では、制約条件有りの最適化問題を解くことができる。簡単なものであれば、手で解くことが可能である。その際に未定乗数によるもうひとつの最適化問題が登場する。

ラグランジュ未定乗数法による解法

制約条件を満たした最適化問題を解く処方箋。未定乗数によるもうひとつの最適化問題が登場して、それを解けば最適解を求めることができる。

この新たに登場したもうひとつの最適化問題を双対問題と呼ぶ。このとき片方で最適解を持てば、もう片方も持ち、元の問題の最小値は、双対問題の最大値と一致することが知られている（双対定理）。双対問題は、制約条件を既に考慮したものであるから、その最適化問題は制約なしで比較的扱いやすいものとな変わっている。この双対問題に対して、これまでに考慮してきた勾配法を適用してもよい。

一方、ラグランジュ未定乗数とは異なり、制約条件を2乗をしたもの（罰金項）を加えたコスト関数を考えることで制約付き最適化問題を解く方法がある。罰金法（Penalty method）と呼ぶ。

$$L_{\text{pen.}}(\mathbf{x}) = f(\mathbf{x}) + \frac{\mu[t]}{2} \sum_{i=1}^m c_i^2(\mathbf{x}) \quad (101)$$

勾配に基づき更新して行く際に、罰金項の係数 $\mu[t]$ を徐々に大きくさせることで最適解に収束させられることが知られている。例題について、試してみよう。罰金項の係数を固定して、コスト関数の微分が0となることを調べる。

$$\frac{\partial}{\partial x_1} L_{\text{pen.}}(x_1, x_2; \mu[t]) = 2x_1 + \mu[t]a(ax_1 + bx_2 - c) = 0 \quad (102)$$

$$\frac{\partial}{\partial x_2} L_{\text{pen.}}(x_1, x_2; \mu[t]) = 2x_2 + \mu[t]b(ax_1 + bx_2 - c) = 0. \quad (103)$$

連立方程式をとくことにより、この最適化問題の解は、

$$x_1 = \frac{\mu[t]ac}{(2 + \mu[t](a^2 + b^2))} \quad (104)$$

$$x_2 = \frac{\mu[t]bc}{(2 + \mu[t](a^2 + b^2))} \quad (105)$$

となることがわかる。ここで $\mu[t]$ を非常に大きな値を取れば、確かに最適解に収束して行くことが分かる。この問題のように連立方程式が手で解けないような場合は、勾配法による更新を行い最適解に徐々に近づけていくことになる。但し、制約条件を満たすようにするためには、罰金項の係数 $\mu[t]$ を最終的に非常に大きな値にしていけないといけな。それに伴って勾配が大きくなり、単純な勾配法の適用下では収束しないことが考えられる。そこで更新幅を適切に小さくすると、今度は必要な計算時間が長くなる傾向にあるという問題点がある。

罰金法による解法

制約条件を満たした最適解に収束させるためには、罰金項を非常に大きくしないといけな。そのために計算時間が長くなる傾向があり厄介である。

制約条件付きの問題を解く場合には、上記のように、ラグランジュ未定乗数法及び罰金法がある。しかしラグランジュ未定乗数法は、双対問題を扱うことになり元々の問題が見えにくくなる。そして罰金法には安定性の問題がある。そこで両者の良いところを組み合わせるために、拡張ラグランジュ法が提案されている。拡張ラグランジュ法は、ラグランジュ未定乗数法と罰金法の両者を組み合わせた方法である。

$$L_{\text{aug.}}(\mathbf{x}; \mathbf{h}) = f(\mathbf{x}) + \sum_{i=1}^m h_i[t]c_i(\mathbf{x}) + \frac{\mu}{2} \sum_{i=1}^m c_i^2(\mathbf{x}) \quad (106)$$

ここで罰金項の係数は固定するところに注目したい。但し更新則はやや変わっており、未定乗数について、

$$h_i[t + 1] = h_i[t] + \mu c_i(\mathbf{x}) \quad (107)$$

と更新する。単純な罰金法と比較して、罰金項の係数を大きくする代わりにラグランジュ未定乗数による効果で制約条件を満たすようにしむけるという作戦である。

再び上記で挙げた例題について考えてみよう。コスト関数の微分が0となることを調べてみよう。

$$\frac{\partial}{\partial x_1} L_{\text{aug.}}(x_1, x_2; h[t], \mu) = 2x_1 + h[t]a + \mu a(ax_1 + bx_2 - c) = 0 \quad (108)$$

$$\frac{\partial}{\partial x_2} L_{\text{aug.}}(x_1, x_2; h[t], \mu) = 2x_2 + h[t]b + \mu b(ax_1 + bx_2 - c) = 0. \quad (109)$$

連立方程式をとくことにより、

$$x_1 = \frac{\mu ac - h[t]a}{2 + \mu(a^2 + b^2)} \quad (110)$$

$$x_2 = \frac{\mu bc - h[t]b}{2 + \mu(a^2 + b^2)} \quad (111)$$

を得る。この解を用いて、ラグランジュ未定乗数の更新がどうなるかを調べてみると、

$$h[t+1] = \frac{-\mu[t](a^2 + b^2)}{2 + \mu[t](a^2 + b^2)} \left(h[t] - \frac{c^2}{a^2 + b^2} \right) \quad (112)$$

となり、必ずしも大きくない $\mu[t]$ で $h[t]$ が最適解に収束することが分かる。

拡張ラグランジュ法による解法

ラグランジュ未定乗数と罰金法を組み合わせた方法。罰金項を大きくしなくても、ラグランジュ未定乗数も含めた最適解へと到達する。

拡張ラグランジュ法におけるコスト関数 $L_{\text{arg.}}$ は、平方完成により、

$$L_{\text{aug.}}(\mathbf{x}; \mathbf{h}) = f(\mathbf{x}) + \frac{\mu}{2} \sum_{i=1}^m \left(c_i(\mathbf{x}) + \frac{h_i[t]}{\mu} \right)^2 \quad (113)$$

と変形できる。このとき $u_i[t] = h_i[t]/\mu$ として利用すると見やすい。

$$L_{\text{aug.}}(\mathbf{x}; \mathbf{h}) = f(\mathbf{x}) + \frac{\mu}{2} \sum_{i=1}^m (c_i(\mathbf{x}) + u_i[t])^2 \quad (114)$$

このとき $u_i[t]$ の更新則は単純に、

$$u_i[t+1] = u_i[t] + c_i(\mathbf{x}) \quad (115)$$

となり、制約条件において満たされない誤差部分が $u_i[t]$ に追加されて補正されていくという描像が得られる。

拡張ラグランジュ法による基底追跡 これを素朴に圧縮センシングの代表的最適化問題である基底追跡型の最適化問題に適用してみよう。まず、最小化したい関数は $f(\mathbf{x}) = \|\mathbf{x}\|_1$ である。制約条件は $c(\mathbf{x}) = \mathbf{y} - \mathbf{A}\mathbf{x}$ である。拡張ラグランジュ法を利用すると、以下のコスト関数の最小化問題に帰着する。

$$L_{\text{arg.}}(\mathbf{x}; \mathbf{h}[t], \mu[t]) = \|\mathbf{x}\|_1 + (\mathbf{h}[t])^T (\mathbf{y} - \mathbf{A}\mathbf{x}) + \frac{\mu}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad (116)$$

ラグランジュ未定乗数の更新則は、

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu(\mathbf{y} - \mathbf{A}\mathbf{x}) \quad (117)$$

である。ここでラグランジュ未定乗数は M 個の制約条件の分 ($\mathbf{h}[t] = h_1[t], h_2[t], \dots, h_M[t]$)、利用していることに注意したい。ここでラグランジュ未定乗数の更新則の両辺に A^T (制約項の微分) をかけると、 $A^T \mathbf{h}[t] = \mathbf{p}[t]$ と置き換えることで、

$$\mathbf{p}[t+1] = \mathbf{p}[t] + \mu A^T (\mathbf{y} - \mathbf{A}\mathbf{x}) \quad (118)$$

となることがわかる。 $\mu = 1/\lambda$ とおいたとき、右辺第二項が $-\nabla g(\mathbf{x}) = A^T (\mathbf{y} - \mathbf{A}\mathbf{x})/\lambda$ であることから Bregman 反復法における更新則 (82) と等価であることが分かる。

Alternating Direction Method of Multipliers(ADMM) 上記の古典的な拡張ラグランジュ法を改良したものが次に紹介する ADMM である [11]. アイデアは単純である. 先ほどまで議論していた拡張ラグランジュ法を用いて、

$$\min_{\mathbf{x}} \{f(\mathbf{x}) + g(\mathbf{x})\} \quad (119)$$

という LASSO 型最小化問題のようなふたつのコスト関数の組み合わせによる最適化問題を制約付きの最適化問題と一旦考え直す.

$$\min_{\mathbf{x}, \mathbf{z}} \{f(\mathbf{z}) + g(\mathbf{x})\} \quad \text{s.t.} \quad \mathbf{x} - \mathbf{z} = 0 \quad (120)$$

基本的には、この制約付き最適化問題に対して拡張ラグランジュ法を適用するというアイデアである. 新しいコスト関数は、拡張ラグランジュ法により、以下のように与えられる.

$$L_{\text{aug.}}(\mathbf{x}, \mathbf{z}; \mathbf{h}[t]) = f(\mathbf{z}) + g(\mathbf{x}) + (\mathbf{h}[t])^T(\mathbf{x} - \mathbf{z}) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (121)$$

これもやはり平方完成して利用する方が見やすいので変形しておこう.

$$L_{\text{aug.}}(\mathbf{x}, \mathbf{z}; \mathbf{h}[t]) = f(\mathbf{z}) + g(\mathbf{x}) + \frac{\mu}{2} \left\| \mathbf{x} - \mathbf{z} + \frac{\mathbf{h}[t]}{\mu} \right\|_2^2 \quad (122)$$

この L_2 ノルムで新たに導入した変数 \mathbf{z} と元の変数 \mathbf{x} がうまく調整合っていると直観的には理解できよう. この拡張ラグランジュ法を利用した変形により、 $f(\mathbf{z})$ 及び $g(\mathbf{x})$ のそれぞれの関数を分離して解くことが ADMM の工夫である. ADMM という名前の通り、以下のように交互に更新する.

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{\mu}{2} \left\| \mathbf{x} - \mathbf{z}[t] + \frac{\mathbf{h}[t]}{\mu} \right\|_2^2 \right\} \quad (123)$$

$$\mathbf{z}[t+1] = \operatorname{argmin}_{\mathbf{z}} \left\{ f(\mathbf{z}) + \frac{\mu}{2} \left\| \mathbf{x}[t+1] - \mathbf{z} + \frac{\mathbf{h}[t]}{\mu} \right\|_2^2 \right\} \quad (124)$$

そしてラグランジュ未定乗数については、

$$\mathbf{h}[t+1] = \mathbf{h}[t] + \mu(\mathbf{x} - \mathbf{z}) \quad (125)$$

として更新する.

ADMM を LASSO 型の最小化問題に適用してみよう. まず $\mathbf{u}[t] = \mathbf{h}[t]/\mu$ とおき、変数の煩雑さを減らしておく. 次に $f(\mathbf{x}) = \|\mathbf{x}\|_1$ および $g(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2/2\lambda$ であるから、解くべき最小化問題はそれぞれ

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z}[t] + \mathbf{u}[t]\|_2^2 \right\} \quad (126)$$

$$\mathbf{z}[t+1] = \operatorname{argmin}_{\mathbf{z}} \left\{ \|\mathbf{z}\|_1 + \frac{\mu}{2} \|\mathbf{x}[t+1] - \mathbf{z} + \mathbf{u}[t]\|_2^2 \right\} \quad (127)$$

である. \mathbf{x} についての最小化問題は 2 次関数であるから平方完成または単純な微分が $\mathbf{0}$ となる条件により以下の通り計算される.

$$\mathbf{x}[t+1] = \left(\mu I + \frac{1}{\lambda} A^T A \right)^{-1} \left(\frac{1}{\lambda} A^T \mathbf{y} + \mu(\mathbf{z} - \mathbf{u}) \right). \quad (128)$$

一方 \mathbf{z} については、 L_1 ノルムおよび L_2 ノルムの分離性を利用して軟判定しきい値関数による解が直ちに求まる.

$$\mathbf{z}[t+1] = S_{1/\mu}(\mathbf{x}[t+1] + \mathbf{u}[t]). \quad (129)$$

つまり何が凄いて、FISTA のように暫定解の近傍付近で、という制限はなく、しかし単なる代入の繰り返しだけで最適解を求めることができることだ.

これらの結果を統合して、LASSO 型の最小化問題を解くアルゴリズムを以下のように組むことができる。

ADMM for LASSO

1. $t = 0$ とする。初期化する。(例えば $\mathbf{x}[0] = A^T \mathbf{x}$, $\mathbf{z}[0] = \mathbf{x}[0]$, $\mathbf{h}[0] = \mathbf{0}$)

2. $\mathbf{x}[t]$ の更新

$$\mathbf{x}[t+1] = \left(\mu I + \frac{1}{\lambda} A^T A \right)^{-1} \left(\frac{1}{\lambda} A^T \mathbf{y} + \mu(\mathbf{z} - \mathbf{u}) \right). \quad (130)$$

3. $\mathbf{z}[t]$ の更新

$$\mathbf{z}[t+1] = S_{1/\mu}(\mathbf{x}[t+1] + \mathbf{u}[t]). \quad (131)$$

4. $\mathbf{u}[t]$ の更新

$$\mathbf{u}[t+1] = \mathbf{u}[t] + (\mathbf{x}[t] - \mathbf{z}[t]) \quad (132)$$

5. 終了基準を満たすまでステップ 2-4 を繰り返す。

罰金項の係数 μ を変化させないという方針のため、ステップ 2 の逆行列の計算は予め行っておけばよい。この逆行列の計算は基本的には LASSO 型の最小化問題を解く際に避けられていた。計算量が膨大な分類 $O(N^3)$ に入るためだ。これまでに様々なアルゴリズムが提案されてきたが、その歴史は如何にして逆行列の計算を避けるかにかかっており、反復の内部で逆行列の計算がないように工夫をしてきた。ADMM はその意味で禁じ手を使っている。しかし 1 回だけ前処理的に用意しておくだけである。ADMM は非常に収束が早い。逆行列の計算と ADMM の反復計算にかかる合計時間が、LASSO 型の最小化問題を解く FISTA 等の方法よりも圧倒的に短いと許されている。

特にスパース性が期待できる基底がある変換を通して見つかる場合、すなわち L_1 ノルムの項が $\|B\mathbf{x}\|_1$ となるときの ADMM は効果が絶大である。先ほどの LASSO 型の最小化問題を解いたのと同様に、敢えて制約付きの最適化問題に置き換えよう。

$$\min_{\mathbf{x}, \mathbf{z}} \left\{ \|\mathbf{z}\|_1 + \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 \right\} \quad \text{s.t.} \quad B\mathbf{x} - \mathbf{z} = \mathbf{0} \quad (133)$$

拡張ラグランジュ法の手続きに従い、以下の新しいコスト関数を最小化すれば良い。

$$L_{\text{aug}}(\mathbf{x}, \mathbf{z}; \mathbf{h}, \mu[t]) = \|\mathbf{z}\|_1 + \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \frac{\mu}{2} \|B\mathbf{x} - \mathbf{z} + \mathbf{u}[t]\|_2^2 \quad (134)$$

解くべき最小化問題はそれぞれ

$$\mathbf{x}[t+1] = \operatorname{argmin}_{\mathbf{x}} \left\{ \frac{1}{2\lambda} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \frac{\mu}{2} \|B\mathbf{x} - \mathbf{z}[t] + \mathbf{u}[t]\|_2^2 \right\} \quad (135)$$

$$\mathbf{z}[t+1] = \operatorname{argmin}_{\mathbf{z}} \left\{ \|\mathbf{z}\|_1 + \frac{\mu}{2} \|B\mathbf{x}[t+1] - \mathbf{z} + \mathbf{u}[t]\|_2^2 \right\} \quad (136)$$

である。先ほどの例と同様に解くことができる。前者は単純な微分、または平方完成により、後者は分離性を \mathbf{z} に注目して行えば直ちに以下の最適解を得ることができる。

$$\mathbf{x}[t+1] = \left(\mu B^T B + \frac{1}{\lambda} A^T A \right)^{-1} \left(\frac{1}{\lambda} A^T \mathbf{y} + \mu B^T (\mathbf{z} - \mathbf{u}) \right) \quad (137)$$

$$\mathbf{z}[t+1] = S_{1/\mu}(B\mathbf{x}[t+1] + \mathbf{u}[t]). \quad (138)$$

これは他の方法に比べて極めて有利な性質である。実は L_1 ノルムの中身がある線形変換 B をされたものであるとすると、FISTA の適用は直ちに行えるわけではない。その変換専用の方法を必要とすることがしばしば発生する。その最たる例が、全変動 (Total Variation) においてスパース性が期待できる問題である。画像等の応用例でよく利用されるもので、隣接するピクセル間での差を取る演算である。定義は様々なので論文等を読むときには注意してほしい。この場合、逆変換は非自明 (多くの場合に B は正則ではな

い) であり、その変換専用の解法を必要とする。全変動に対しては、メジャライザー最小化と組み合わせた Fast Gradient Projection (FGP) が提案されている [12]。また複数の基底でスパース性が期待できる場合、 L_1 ノルムの項を複数もつ最小化問題を解くことがある。この場合についても FISTA は苦手としていたが、最近 Fast Composit Splitting Algorithm (FCSA) の登場によりその難点は克服している [13]。それでもなお ADMM はそういった難点を未定乗数と罰金項の追加だけで柔軟に対応するため、特段の苦勞なく利用できる意味で圧倒的に有利である。また解析的に解いた最適解を逐次代入するだけで解けるというところも圧倒的な利用数の獲得に一役買っているだろう。

最後に基底追跡型の最適化問題に対する ADMM を紹介して、この章を終えることにしよう。基底追跡型の最適化問題では、 L_1 ノルムの最小化だけではなく。制約条件 $\mathbf{y} = \mathbf{A}\mathbf{x}$ を満たす必要がある。思わずこの制約条件に対して、拡張ラグランジュ法を適用してしまいそうになるが、まずはラグランジュ未定乗数法を利用して、以下のコスト関数を考えることにする。

$$L(\mathbf{x}) = \|\mathbf{x}\|_1 + \boldsymbol{\nu}^T(\mathbf{y} - \mathbf{A}\mathbf{x}) \quad (139)$$

ここで $\boldsymbol{\nu}$ はラグランジュ未定乗数である。ラグランジュ未定乗数法の精神は、双対問題へのマップである。そのときには微分をして、最適解における未定乗数と元の変数の関係を明らかにする必要がある。しかし、ここで \mathbf{x} の位置が悩ましいことに気付くだろう。絶対値関数がある。そこで、ADMM の手続きを施す。

$$L_{\text{aug}}(\mathbf{x}, \mathbf{z}; \mathbf{h}[t]) = \|\mathbf{z}\|_1 + \boldsymbol{\nu}^T(\mathbf{y} - \mathbf{A}\mathbf{x}) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{u}[t]\|_2^2 \quad (140)$$

こうしておけば、 \mathbf{x} について微分を行うことができる。まず \mathbf{x} について微分を行い、最適解を求めようとする。

$$\nabla_{\mathbf{x}} L_{\text{aug}}(\mathbf{x}, \mathbf{z}; \mathbf{h}[t]) = -A^T \boldsymbol{\nu} + \mu(\mathbf{x} - \mathbf{z} + \mathbf{u}[t]) = \mathbf{0} \quad (141)$$

より、

$$\mathbf{x} = \frac{1}{\mu} A^T \boldsymbol{\nu} + (\mathbf{z} - \mathbf{u}[t]). \quad (142)$$

この解をコスト関数に代入して、 $\boldsymbol{\nu}$ に関する最適化問題に置き換える。

$$L_{\text{aug}}(\boldsymbol{\nu}, \mathbf{z}; \mathbf{h}[t]) = \|\mathbf{z}\|_1 + \boldsymbol{\nu}^T(\mathbf{y} + A(\mathbf{z} - \mathbf{u})) - \frac{1}{2\mu} \|A^T \boldsymbol{\nu}\|_2^2. \quad (143)$$

ここで $\boldsymbol{\nu}$ について 2 次関数であるから容易に最大化を行うことができ、最適解は、

$$\boldsymbol{\nu} = \mu (AA^T)^{-1} (\mathbf{y} + A(\mathbf{z} - \mathbf{u})) \quad (144)$$

である。これを再び \mathbf{x} の最適解に代入すると、

$$\mathbf{x} = A^T (AA^T)^{-1} \mathbf{y} + \left(I - A^T (AA^T)^{-1} A \right) (\mathbf{z} - \mathbf{u}[t]). \quad (145)$$

を得る。この解は $\mathbf{y} = \mathbf{A}\mathbf{x}$ を常に満たしている。つまりアルゴリズムのどの時点で止めても、制約条件が満たされているという顕著な性質を持っている。残るは \mathbf{z} と \mathbf{u} の計算であるが、これはこれまでの例と全く同様にして計算できるので心配は要らない。結果として基底追跡型の最適化問題を解くために ADMM を適用したアルゴリズムは、

$$\mathbf{x}[t+1] = A^T (AA^T)^{-1} \mathbf{y} + \left(I - A^T (AA^T)^{-1} A \right) (\mathbf{z}[t] - \mathbf{u}[t]) \quad (146)$$

$$\mathbf{z}[t+1] = S_{1/\mu}(\mathbf{x}[t+1] + \mathbf{u}[t]) \quad (147)$$

$$\mathbf{u}[t+1] = \mathbf{u}[t] + (\mathbf{x}[t+1] - \mathbf{z}[t+1]) \quad (148)$$

を繰り返して代入して行けば良い。

4 圧縮センシング実装編

それでは理論はよくわかった、とにかく実装してみたいという人のために、実際に圧縮センシングの問題を解くためのプログラムを組むことをやってみよう。基本的には matlab を使ったコードを以下に紹介するが、同様の操作を行えば基本的には動く。matlab を使用した理由は、行列やベクトルが入り混じった計算部分や操作などについて、数式表現のまま感覚的にコーディング出来ることにある。同様のコーディングが可能なものとして Octave や Python などフリーに利用できるものがある。随時そのコードも公開していきたい。

まずは実験のために、人工的に問題設定をする部分を作成しよう。

```
% 再構成したい信号の次元
N = 100;

% 観測数
M = 50;

% 正解の原信号の次元
K = 10;

% 観測行列 A の設定
A = randn(M,N);

% 原信号 x0 の設定
x0 = [randn(K,1); zeros(N-K,1)];
x0 = x0(randperm(N));

% 観測信号 y の設定
y = A*x0;
```

ここまでで、 $N = 100$ 次元の中で、 $K = 10$ 個しか非零成分がない原信号 x_0 を作っている。その原信号の非零成分は標準正規分布に従う乱数 ($\text{randn}(K,1)$ が $K \times 1$ 次元の乱数を生成する部分) で生成した。観測行列は簡単のため $\text{randn}(M,N)$ により、標準正規分布で各成分をランダムに生成した。もちろんご自身の問題設定に合わせて変えて構わない。 $y = Ax_0$ として、原信号から観測行列を経て、観測信号 y を取得した状況を考える。この後では原信号は使わない。あくまで推定して、 x_0 を当てに行く。

4.1 ISTA の実装

さて、それでは手始めに ISTA を実装してみよう。初期条件として、 x は $x = A^T y$ から始めるとしよう。

```
% 初期条件
x = A'*y;
```

LASSO 型の最適化問題を解くので、 L_1 ノルムの係数 λ を適当に設定する。

```
% L1 ノルムの係数の設定
lambda = 1.0;
```

またメジャライザー最小化を用いるために、リップシッツ定数 L を考えなければならない。問題によっては事前に手で計算することができるが、汎用的に使えるように、ここではラインサーチ法を用いて探索することで L を適応的に変える方法を取る。初期値は $L = 1$ としておく。

```
% リップシッツ定数の初期条件
L0 = 1;
```

さて準備は整った。ISTA は繰り返し計算で、徐々に正解へと近づく方法だ。そのためプログラムとしては、for 文を使えば良い。とりあえず 100 回程度の計算をすることにしよう。

```
% 100 回反復計算
T=100;
for t = 1:T
```

```

% 前回の結果を kx に格納
kx = x;

% 前回の結果におけるコスト関数の計算 (L2ノルム部分のみ)
ktemp = 0.5*norm(y-A*x,2)^2;

% 適応的に変化させるリプシッツ定数をリセット
L = L0;

(以降登場するコードを記入)

end

```

繰り返し型の計算をする場合には、前の計算結果を利用したり比較することが多いので、for 文の中身部分の最初の時点で前回の計算結果を格納しておく。norm(*,*) はノルムを計算する便利な関数である。norm(*,2) であれば L_2 ノルム、norm(*,1) であれば L_1 ノルムを計算することができる。用意されていない言語であっても、定義に従って計算する関数を用意すれば良い。さて ISTA の中心部分を書いてみよう。

```

% 勾配計算
grad_x = -A'*(y-A*x);

% 更新候補点へ移動
MMx = x - grad_x/L;

```

ISTA の基本は勾配法であるから、まずは勾配を計算したものを用意する。問題が変われば、この勾配部分を変更すれば良い。Mmx は仮に勾配法で進んだ場合の候補位置である。メジャライザー最小化によれば、コスト関数のうち $\|y - Ax\|_2^2 / 2$ の部分を上から押さえるメジャライザーとなる二次関数を用意すれば良い。

```

%コスト関数の一部を計算
temp = 0.5*norm(y-A*MMx,2)^2;

%メジャライザーを計算
MMtemp = ktemp + grad_x'*(MMx-kx)+0.5*L*norm(MMx-kx,2)^2;

```

もしもこの時にコスト関数とメジャライザーを比較してコスト関数が上回ることがあってはいけな。その時はリプシッツ定数を変えることで、そうならないように変更する必要がある。それはメジャライザーが上回るまで繰り返す必要があるため下記のように続ける。

```

while MMtemp < temp
    %リプシッツ定数を大きくしてみる
    L = L*1.1;

    %更新候補位置再計算
    MMx = x - grad_x./L;

    %コスト関数の一部を再計算
    temp = 0.5*norm(y-A*MMx,2)^2;
    %メジャライザーを再計算
    MMtemp = ktemp + grad_x'*(MMx-kx)+0.5*L*norm(MMx-kx,2)^2;

end

```

ここで差し込んだ while 文の中身で実行していることがラインサーチ法と呼ばれるものである。L を単調に大きくしていくために 1.1 倍している。この比率を変えても構わない。大きくすればざっくりとした探索になり、細かくすれば丁寧であるが時間がかかるようになる。そして最後に軟判定しきい値関数を適用すれば ISTA の実装が完了する。

```

%軟判定しきい値関数の適用
x = SoftThr(MMx,lambda/L);

```

ここで SoftThr は以下のように定義した。

```

function y = SoftThr(x,lambda)
%とりあえず全て0にセット

```

```

y = zeros(size(x));
%しきい値を超える集合をtempに格納
temp = find(x > lambda);
%しきい値を超えるものだけlambda分割る
y(temp) = x(temp) - lambda;
%しきい値を下回る集合をtempに格納
temp = find(x < -lambda);
%しきい値を下回るものだけlambda分増やす
y(temp) = x(temp) + lambda;

```

軟判定しきい値関数の定義通りのものがこれで組み上がる。以上で ISTA が実装できる。FISTA もこれを基本にして、 $\beta[t]$ の更新を for 文の中に入れることで実装できる。その実行結果を図 15 に掲載する。

問:FISTA を実装してみよ。

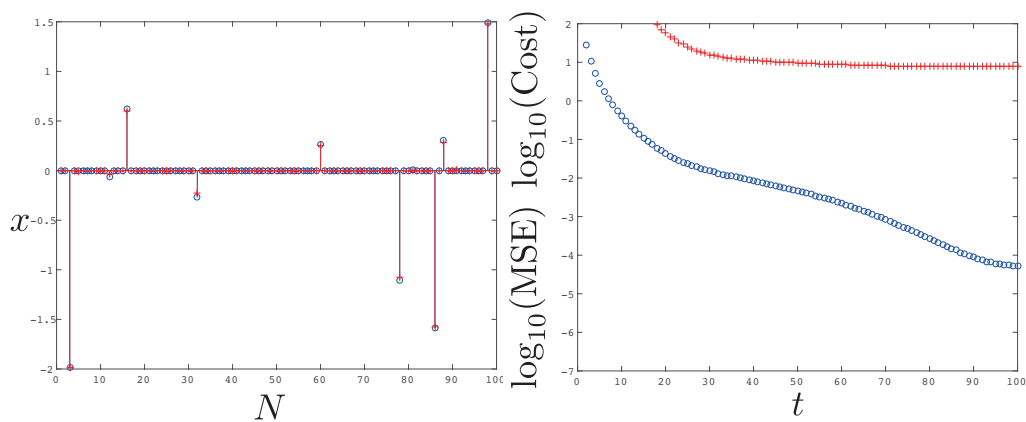


図 14: 左：原信号 (青丸) と推定信号 (赤十字) の比較. 右： $\log_{10}(\text{MSE})$ (青丸)、 $\log_{10}(\text{Cost})$ (赤十字) の減衰の様子.

4.2 ADMM の実装

さらに進んで FISTA よりも柔軟性があり、高速に最適解に近づくことが知られている ADMM を実装してみよう。こっちの方がずっとシンプルで実装にはてっとり早い。まず登場する変数が増えるので、以下のように初期条件を設定する。

```

% 初期条件
x = zeros(N,1);
z = zeros(N,1);
u = zeros(N,1);

```

BP 型の最適化問題を解くので、 λ を設定する必要はない。代わりに罰金項に相当する μ を設定する必要がある。この係数により、だいぶ振る舞いが変わることには注意したい。

```

% 罰金項の係数の設定
mu = 1.0

```

また何度も同じ行列が登場するので先にその部分を計算しておき、メモリに格納しておくでしょう。

```

A1 = A' * inv(A * A');
A2 = eye(N,N) - A1 * A;

```

さてそれでは ADMM も反復計算であるから for 文を利用しよう。

```

% 50回反復計算
T=50;
for t = 1:T

%ADMM更新式
x = A1*y + A2*(z-u);
z = SoftThr(x+u,1/mu);
u = u + x - z;

```

これだけで、ISTA・FISTA に比べて圧倒的にコードはシンプルである。しかも結果を見てみると驚きの性能である。図??に 50 ステップ実行した後の結果を載せる。半分ほどのステップであるにもかかわらず、やや振動しながらほぼ原信号と完全一致する結果を得ることに成功している。ISTA や FISTA ではメジャラ

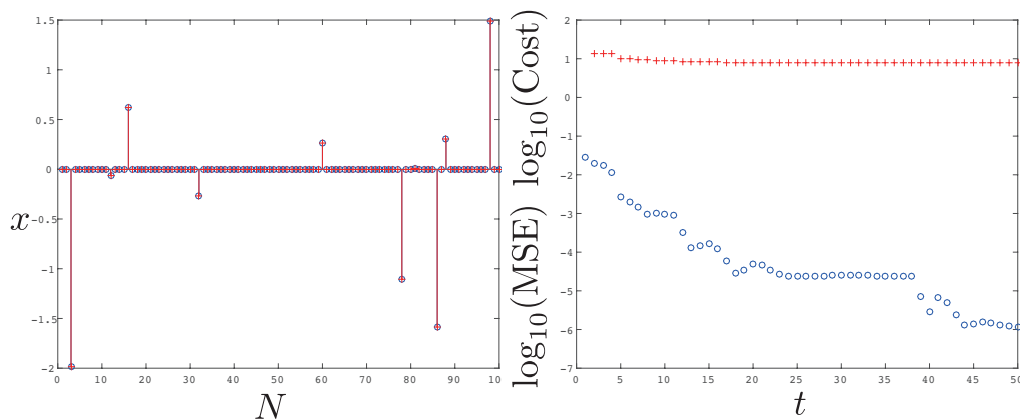


図 15: 左：原信号 (青丸) と推定信号 (赤十字) の比較. 右： $\log_{10}(\text{MSE})$ (青丸)、 $\log_{10}(\text{Cost})$ (赤十字) の減衰の様子.

イザ最小化を行っているため、着実にコスト関数を下げるといった性質がある。一方で ADMM は交互に更新を行う影響で、必ずしもコスト関数下がるというわけではない。しかし全体としては急激にコスト関数を下げること、ISTA や FISTA よりも速くコスト関数の小さい解を得ることができる。また正則化項を増やしても、変数の増加は生じるが、柔軟に対応することが可能であり、今や基本的には圧縮センシングにおいて再構成を行う場合には ADMM による方法や派生したアルゴリズムが実装されていることが多い。しかし発展目覚ましい分野であるので他の方法論も生まれて置き換わっていくことの繰り返しである。ただし基本的な精神は同一のもので多いため、ここで学んだことは役に立つことも多いだろう。

5 圧縮センシングと情報統計力学

ある条件下のもとでは、 L_1 ノルム最小化により、スパースな解を得て、しかもその解が入力ベクトルと一致する。このことが圧縮センシングの驚異的な力を示している。それではその条件を少し変わった方法で明らかにしてみよう。

5.1 スピン系の統計力学

ここで統計力学の処方箋について、簡単に紹介する。あるエネルギー関数 (ハミルトニアン) のもとでの平衡状態を調べるためには自由エネルギーを計算すればよい。ここで全結合型のイジング模型のエネルギー関数を用意する。

$$E(\mathbf{x}) = -\frac{J}{N} \sum_{i < j} x_i x_j \quad (149)$$

x_i は各点におかれたイジングスピンで $x_i = \pm 1$ をとる. 相互作用の強度を J として強磁性的な相互作用 ($J > 0$) を考える. カノニカル分布の平衡状態にあるとき、スピンの状態はつぎの確率分布に従う.

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left(-\frac{E(\mathbf{x})}{T}\right) \quad (150)$$

ここで Z は規格化定数であり分配関数である. 自由エネルギーを計算するためにはこの分配関数を計算すればよい. それでは分配関数の計算を実行してみよう.

$$Z = \sum_{\mathbf{x}} \exp\left(\frac{J}{NT} \sum_{i < j} x_i x_j\right) \quad (151)$$

よく利用される以下の恒等式を用いて分配関数を書き換えよう. 指数の肩に現れている和はいわゆる和の 2 乗を計算したときのクロスターム部分であるから、

$$\frac{1}{N^2} \sum_{i < j} x_i x_j = \frac{1}{2} \left\{ \left(\frac{1}{N} \sum_{i=1}^N x_i \right)^2 - \frac{1}{N^2} \sum_{i=1}^N x_i^2 \right\} \approx \frac{1}{2} \left(\frac{1}{N} \sum_{i=1}^N x_i \right)^2 + O\left(\frac{1}{N}\right) \quad (152)$$

とかける. この表示を用いると分配関数は、

$$Z = \sum_{\mathbf{x}} \exp\left\{ \frac{NJ}{2T} \left(\frac{1}{N} \sum_{i=1}^N x_i \right)^2 \right\} \quad (153)$$

ここで現れる $\sum_{i=1}^N x_i/N$ はスピンの揃い具合を表している磁化である. このように微視的自由度の平均的性質を推し量る物理量を**秩序パラメータ**と呼ぶ. そこで秩序パラメータである磁化

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad (154)$$

に注目して、特定の m を与えるものについて \mathbf{x} の和を取り、そして m の積分を実行することにしよう.

$$Z = \sum_{\mathbf{x}} \int dm \delta\left(m - \frac{1}{N} \sum_{i=1}^N x_i\right) \exp\left(\frac{NJ}{2T} m^2\right) \quad (155)$$

この操作は以下の恒等式を分配関数の中身に挿入したと考えてもよい.

$$1 = \int dm \delta\left(m - \frac{1}{N} \sum_{i=1}^N x_i\right) \quad (156)$$

ここで少し意味合いを考えよう. ある磁化 m を持つ微視的状态 \mathbf{x} の組み合わせについて全て和を取るというのは、状態数の数え上げに他ならない. その対数を取ったものをエントロピーと呼ぶ. 本来状態数はエネルギーを引数としてもつが、この場合エネルギーに相当する部分は $NJm^2/2$ であるからエネルギーの代わりに m を用いても意味は変わらない. そこで以下のように文字をおく.

$$-\frac{1}{T} e(m) = \frac{J}{2T} m^2 \quad (157)$$

$$s(m) = \frac{1}{N} \log \sum_{\mathbf{x}} \delta\left(m - \frac{1}{N} \sum_{i=1}^N x_i\right) \quad (158)$$

このとき、分配関数は

$$Z = \int dm \exp\left\{ N \left(-\frac{e(m)}{T} + s(m) \right) \right\} \quad (159)$$

という表示を持つ. ここで統計力学の前提である $N \rightarrow \infty$ (熱力学極限) を考慮すると、積分の主要な寄与は鞍点からのみ決まるので

$$Z = \exp\left\{ N \left(-\frac{e(m^*)}{T} + s(m^*) \right) \right\} \quad (160)$$

となる。ここで m^* は

$$m^* = \arg \max_m \left\{ -\frac{e(m)}{T} + s(m) \right\} \quad (161)$$

から決まる最大値をとるときの m である。分配関数の対数を取り、 N で割ることで1 スピンあたりの自由エネルギーを求めると、

$$-f = \frac{T}{N} \log Z = \max_m \{-e(m) + Ts(m)\} \quad (162)$$

という熱力学でよく知られた変分原理を再現する。(ヘルムホルツの自由エネルギーはエネルギーからエントロピーの効果を引いたものの最小化で与えられる。) それでは計算の話に戻ろう。残る計算するべき量はエントロピーである。デルタ関数の積分表示を用いて、以下のように書き換える。

$$\delta \left(m - \frac{1}{N} \sum_{i=1}^N x_i \right) = \int d\tilde{m} \exp \left\{ \tilde{m} \left(Nm - \sum_{i=1}^N x_i \right) \right\} \quad (163)$$

この表示によりエントロピーは、

$$s(m) = \frac{1}{N} \log \left\{ \int d\tilde{m} \exp(Nm\tilde{m}) \prod_{i=1}^N \sum_{x_i} \exp(\tilde{m}x_i) \right\} \quad (164)$$

と変形できる。ここで

$$\sum_{\mathbf{x}} \prod_{i=1}^N f(x_i) = \prod_{i=1}^N \sum_{x_i} f(x_i) \quad (165)$$

という関係を用いた、 x_i についての和をとると

$$s(m) = \frac{1}{N} \log \left\{ \int d\tilde{m} \exp(Nm\tilde{m} + N \log 2 \cosh \tilde{m}) \right\} \quad (166)$$

ここでも同様に $N \rightarrow \infty$ を考慮して鞍点法を適用すると、

$$s(m) = m\tilde{m}^* + \log 2 \cosh \tilde{m}^* \quad (167)$$

を得る。ここで \tilde{m}^* は

$$\tilde{m}^* = \arg \max_{\tilde{m}} \{m\tilde{m} + \log 2 \cosh \tilde{m}\} \quad (168)$$

である。全ての結果をまとめると、自由エネルギーは次の m と \tilde{m} についての最小化問題を解けばよいことがわかる。

$$f = \min_{m, \tilde{m}} \{Jm^2 + Tm\tilde{m} + T \log (2 \cosh \tilde{m})\} \quad (169)$$

この最小化問題を解くと、 $\tilde{m} = -\tanh m$ が成立するので、 m についての自己無撞着方程式が得られる。

$$m = \tanh \left(\frac{J}{T} m \right) \quad (170)$$

この自己無撞着方程式を反復代入により固定点を求めることで解くことができる。

$$m[t+1] = \tanh \left(\frac{J}{T} m[t] \right) \quad (171)$$

温度と結合の変化に応じて、全結合相互作用をするイジング模型の相転移を議論することができる。

[問: $K = J/T$ として、 K を変化させたときの磁化の振る舞いについて数値的に調べよ。]

5.2 基底追跡の性能評価

ベイズの定理により、最大事後確率という観点で、LASSO 型の最適化問題を定式化することができた。同様に $\lambda \rightarrow 0$ とすると、

$$P(\mathbf{x}|\mathbf{A}, \mathbf{x}_0) \propto \delta(\mathbf{y} - \mathbf{A}\mathbf{x}_0) \exp(-\kappa \|\mathbf{x}\|_1) \quad (172)$$

という事後確率を用いて、基底追跡の最適化問題を最大事後確率という観点で議論できる。この最適化問題が入力ベクトルを正しく推定できるかどうかを議論しよう。事後確率に β 乗した重みによる平均を

$$\langle \cdots \rangle_{\mathbf{x}|\mathbf{A}, \mathbf{y}}^\beta = \int d\mathbf{x} \cdots \times P^\beta(\mathbf{x}|\mathbf{A}, \mathbf{y}) \quad (173)$$

とすると、推定結果が典型的に入力ベクトルと一致するかどうかは、以下の最小二乗誤差の事後平均を議論すれば良い。

$$\text{MSE} = \left\langle \frac{1}{N} \|\mathbf{x} - \mathbf{x}_0\|_2^2 \right\rangle_{\mathbf{x}|\mathbf{A}, \mathbf{y}}^\beta \quad (174)$$

$\beta = 1$ とすれば事後平均そのものを議論することができるし、事後確率の最大化による推定結果を議論するためには $\beta \rightarrow \infty$ を議論すればよい。このような期待値を計算するには統計力学の処方箋が有用である。事後確率（の β 乗）をカノニカル分布と考えれば、先ほどのスピン系の統計力学と同様に自由エネルギーの評価を通じて、期待値がどのように振る舞うかが計算できる。

さてそこでノイズなしの観測を行った場合に、基底追跡型の最適化問題を解くことにより得られる推定解が元の入力ベクトルとどれだけ一致するのかという問題を考えよう。前提条件として下記の設定を置こう。

- 観測行列 A は各成分、平均 0、分散 $1/N$ に従うランダム変数であるとする。
- 入力ベクトル \mathbf{x}_0 の各成分は、 $\rho = K/N$ 程度の非零要素があり、非零要素は正規分布に従うと仮定する。

$$P_0(x) = (1 - \rho)\delta(x) + \rho \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \quad (175)$$

- 出力ベクトルは $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ によって与えられる。

熱力学的諸量の計算 基底追跡型の最適化問題に相当する事後確率は、式 (172) で与えられるので、次の分配関数を評価することを考えよう。

$$Z(A, \mathbf{x}_0) = \lim_{\lambda \rightarrow +0} \int d\mathbf{x} \exp\left(-\frac{1}{2\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 - \beta \|\mathbf{x}\|_1\right) \quad (176)$$

ここで

$$\delta(\mathbf{y} - \mathbf{A}\mathbf{x}) = \lim_{\lambda \rightarrow +0} \exp\left(-\frac{1}{2\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2\right) \quad (177)$$

とデルタ関数を極限で表現する。ちょうどこれは再構成アルゴリズムの章でいうところの罰金法を利用していることに相当する。次に注目したいのは観測行列 A と出力情報 \mathbf{x}_0 である。分配関数、すなわち自由エネルギーがその実現値に依存している。しかしながら $N \rightarrow \infty$ の熱力学的極限では、1 自由度辺りの自由エネルギーはある特定の典型的な値を取り、その値が平均に収束するという性質が知られている。**自己平均性** と呼ばれる性質である。そこで典型的な自由エネルギーの評価を行うために、 A と \mathbf{x}_0 についての平均操作を行う。

$$-f = \left[\frac{1}{N} \log Z(A, \mathbf{x}_0) \right]_{A, \mathbf{x}_0} \quad (178)$$

ここで A と \mathbf{x}_0 の実現値に関する平均を（相互作用や磁場がやはりランダム変数の問題である）スピングラス理論の用語を借りて、配位平均と呼ぶ。この対数の外からの平均操作は非常に難しいため、レプリカ法と呼ばれる数学的恒等式と解析接続を利用した計算を展開する。

レプリカ法

対数の外からの平均操作のために以下の恒等式を用いる。

$$[\log Z(A, \mathbf{x}_0)]_{A, \mathbf{x}_0} = \lim_{n \rightarrow 0} \frac{[Z^n(A, \mathbf{x}_0)]_{A, \mathbf{x}_0} - 1}{n} \quad (179)$$

ここで分配関数の冪が現れるが、一旦 n が実数であることを忘れて、自然数であると仮定して**同じ系のコピー**が存在するものとして計算を進める。最終的に n に関する式を得たときに実数であることを思い出して解析接続を行う。

それではレプリカ法にもとづき、分配関数の冪乗の平均を計算してみよう。

$$[Z^n(A, \mathbf{x}_0)]_{A, \mathbf{x}_0} = \lim_{\lambda \rightarrow +0} \left[\int d\mathbf{x}^a \exp \left(-\frac{1}{2\lambda} \sum_{a=1}^n \|\mathbf{y} - A\mathbf{x}^a\|_2^2 - \beta \sum_{a=1}^n \|\mathbf{x}^a\|_1 \right) \right]_{A, \mathbf{x}_0} \quad (180)$$

冪乗をとった影響で n 個のコピーをもつシステムの統計力学に帰着した。さてまずは A についての平均であるが、 A が登場する項について注目すると、 $\mathbf{t}^a = A\mathbf{x}^0 - A\mathbf{x}^a$ という M 次元のベクトルの部分に現れるのみである。また A がガウス分布に従うことから \mathbf{t}^a も多変量正規分布に従う。この量の平均を調べると A に関する仮定より $\mathbf{0}$ であり、共分散を調べると

$$(\mathbf{t}^a)^T \mathbf{t}^b = \frac{1}{N} ((\mathbf{x}^0)^T \mathbf{x}_0 - (\mathbf{x}^0)^T \mathbf{x}^a - (\mathbf{x}^0)^T \mathbf{x}^b + (\mathbf{x}^a)^T \mathbf{x}^b) \quad (181)$$

となることがわかる。それぞれ

$$q^{ab} = \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \quad (182)$$

と定義する。これはスピン系の統計力学で利用した磁化 $m = \sum_{i=1}^N x_i / N$ と同じように微視的状态の組み合わせからなる量の経験平均で秩序パラメータを定義している。その秩序パラメータを固定して、微視的状态について先に和を取り、あとで秩序パラメータを変化させるというのが統計力学の処方箋にもとづくアプローチであった。そこで分配関数の内部にやはり同様に、

$$1 = \prod_{a,b} \int dq^{ab} \delta \left(q^{ab} - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) \quad (183)$$

なる恒等式を代入して、 \mathbf{x}^a についての積分、 \mathbf{x}^0 についての平均をまとめてエントロピーとして定義しておこう。

$$s(\{q^{ab}\}) = \frac{1}{N} \log \left\{ \int \prod_{a=0}^n d\mathbf{x}^a \left[\exp \left(-\beta \sum_{a=1}^n \|\mathbf{x}^a\|_1 \right) \prod_{a,b} \delta \left(q^{ab} - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) \right] \right\}_{\mathbf{x}_0} \quad (184)$$

分配関数の計算は今の段階で、

$$[Z(A, \mathbf{y})]_{A, \mathbf{w}, \mathbf{x}_0} = \lim_{\lambda \rightarrow +0} \int dq^{ab} \left[\exp \left(-\frac{1}{2\lambda} \sum_{a=1}^n \|\mathbf{t}^a\|_2^2 \right) \right]_{\mathbf{t}^a} \exp(Ns(\{q^{ab}\})) \quad (185)$$

と変形させることに成功した。ここで \mathbf{t}^a についての平均は、先ほど考察したように多変量正規分布に従うので、次の確率分布に従い計算をする。

$$P(\mathbf{t}^a | \mathcal{Q}) = \left(\sqrt{\frac{\det(\mathcal{Q}^{-1})}{(2\pi)^n}} \right)^N \exp \left(-\frac{1}{2} \sum_{a,b} (\mathbf{t}^a)^T (\mathcal{Q}^{-1})^{ab} \mathbf{t}^b \right) \quad (186)$$

ここで行列 \mathcal{Q} が共分散行列であり、 $(\mathcal{Q})^{ab} = q^{ab}$ を指す。注意したいのが \mathcal{Q} は $n \times n$ 行列であることだ。添字 a, b について和をとっており、 \mathbf{t}^a は N 次元のベクトルである。そのため行列 \mathcal{Q} による2次形式につい

てのガウス積分が同様に N 回登場するという格好である。分配関数に現れるエントロピー以外の項をまとめて内部エネルギーを得ることができる。

$$-e(\{q^{ab}\}) = \frac{1}{N} \log \left[\exp \left(-\frac{1}{2\lambda} \sum_{a=1}^n \|\mathbf{t}^a\|_2^2 \right) \right]_{\mathbf{t}^a} \quad (187)$$

こうすることでスピン系の統計力学と同様に、

$$[Z(A, \mathbf{y})]_{A, \mathbf{w}, \mathbf{x}_0} = \lim_{\lambda \rightarrow +0} \int dq^{ab} \exp(-Ne(\{q^{ab}\}) + Ns(\{q^{ab}\})) \quad (188)$$

分配関数の評価は鞍点評価に落ちる。

レプリカ対称解の仮定と内部エネルギーの評価 まず内部エネルギーの計算をしてみよう。対数の内部に注目すると、残る計算は、

$$\int d\mathbf{t}^a \left(\sqrt{\frac{\det(Q^{-1})}{(2\pi)^n}} \right)^N \exp \left\{ -\frac{1}{2} \sum_{a,b} (\mathbf{t}^a)^T \left(\frac{1}{\lambda} \delta_{ab} + (Q^{-1})^{ab} \right) \mathbf{t}^b \right\} \quad (189)$$

というガウス積分を行えばよい。

ガウス積分

ガウス積分の公式

$$\int dx \sqrt{\frac{a}{2\pi}} \exp \left(-\frac{a}{2} x^2 + bx \right) = \exp \left(\frac{b^2}{2a} \right) \quad (190)$$

及びその N 次元への一般化

$$\int d\mathbf{x} \sqrt{\frac{\det(A)}{(2\pi)^N}} \exp \left(-\frac{1}{2} \mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} \right) = \exp \left(\frac{1}{2} \mathbf{b}^T A^{-1} \mathbf{b} \right) \quad (191)$$

を用いる。以降頻繁にガウス積分が登場するので

$$\int Dx = \int \frac{dx}{\sqrt{2\pi}} \exp \left(\frac{1}{2} x^2 \right) \quad (192)$$

と書く。

実際に \mathbf{t}^a についてガウス積分を実行すると、

$$-e(\{q^{ab}\}) = -\frac{\alpha}{2} \log \det \left(I + \frac{1}{\lambda} Q \right) \quad (193)$$

を得る。ここで $\alpha = M/N$ であり、 \mathbf{t}^a の次元が M であったことに注意してもらいたい。有名な公式 $\log \det(A) = \text{Tr} \log \Lambda$ (Λ は A の対角化によって得られる対角行列) を用いれば良いことがわかる。つまり問題は固有値問題に帰着した。しかしながら共分散行列 Q についてどんな特徴があるだろうか。計算を押し進めるために以下の考察にもとづき共分散行列の構造を仮設する。添字 0 は特別であるとして、 a については同じ系のコピーに過ぎないのだから、添字の入れ替えについて対称であると仮定することには無理がないだろう。そこで以下のようなレプリカ対称解をおく。

$$q_{0a} = m \quad (a > 0) \quad (194)$$

$$q_{aa} = Q \quad (a > 0) \quad (195)$$

$$q^{ab} = q \quad (a \neq b) \quad (196)$$

と置くことにする。 $q^{00} = \rho$ は定義より定まっている。これを**レプリカ対称性**の仮定と呼ぶ。(レプリカ対称性の破れとは、この対称解があるパラメータ領域では不安定化することを指す。) このとき共分散行列は以

下の構造を持つ.

$$Q = \begin{pmatrix} \rho - 2m + Q & \rho - 2m + q & \cdots & \rho - 2m + q \\ \rho - 2m + q & \rho - 2m + Q & \cdots & \rho - 2m + q \\ \vdots & & \ddots & \vdots \\ \rho - 2m + q & \rho - 2m + q & \cdots & \rho - 2m + Q \end{pmatrix} = (Q - q)I_n + (\rho - 2m + q)1_n \quad (197)$$

ここで I_n が $n \times n$ の単位行列、 1_n が $n \times n$ 全成分 1 の行列である. よって $I + Q/\lambda$ の固有値を求めると、1 個の $1 + (Q - q)/\lambda + n(\rho - 2m + q)/\lambda$ と $n - 1$ 個の $1 + (Q - q)/\lambda$ という固有値を持つことが分かる.

[問:固有値を確認せよ.]

よって以下の最終的な表式を得る. (ここで n が非常に小さいということを使っている.)

$$-e(\rho, Q, m, q) = -n \frac{\alpha}{2} \frac{\rho - 2m + q}{\lambda + (Q - q)} - \frac{n}{2} \log \left(1 + \frac{1}{\lambda} (Q - q) \right) \quad (198)$$

エントロピーの評価 スピン系の統計力学の場合と全く同様にしてデルタ関数のフーリエ積分表示を行うことで実行できる. まずレプリカ対称解を仮定したので出てくるデルタ関数は 3 つのタイプがある.

$$\delta \left(Q - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^a \right) = \int d\tilde{Q} \exp \left\{ \frac{\tilde{Q}}{2} (NQ - (\mathbf{x}^a)^T \mathbf{x}^a) \right\} \quad (199)$$

$$\delta \left(q - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) = \int d\tilde{q} \exp \left\{ -\frac{\tilde{q}}{2} (Nq - (\mathbf{x}^a)^T \mathbf{x}^b) \right\} \quad (200)$$

$$\delta \left(m - \frac{1}{N} \mathbf{x}_0^T \mathbf{x}^a \right) = \int d\tilde{m} \exp \left\{ -\tilde{m} (Nm - \mathbf{x}_0^T \mathbf{x}^a) \right\} \quad (201)$$

それぞれ積分変数の符号を変えたり係数を変えているのは後々の便利のためである. これらの積がエントロピーの対数の内部に現れるので、その部分にまず注目してみよう.

$$\begin{aligned} \prod_{a,b} \delta \left(q^{ab} - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) &= \exp \left(N \frac{n}{2} \tilde{Q} Q - N \frac{n(n-1)}{2} \tilde{q} q - N n \tilde{m} m \right) \\ &\times \prod_{a=1}^n \exp \left(-\frac{1}{2} \tilde{Q} (\mathbf{x}^a)^T \mathbf{x}^a + \tilde{m} \mathbf{x}_0^T \mathbf{x}^a \right) \prod_{a \neq b} \exp \left(\frac{1}{2} \tilde{q} (\mathbf{x}^a)^T \mathbf{x}^b \right) \end{aligned} \quad (202)$$

最後の項は見覚えがある. レプリカの添字についてのクロスタームであることに気づくと、

$$\prod_{a \neq b} \exp \left(\frac{1}{2} \tilde{q} (\mathbf{x}^a)^T \mathbf{x}^b \right) = \exp \left(\frac{\tilde{q}}{2} \left\{ \left(\sum_{a=1}^n \mathbf{x}^a \right)^2 - \sum_{a=1}^n (\mathbf{x}^a)^T \mathbf{x}^a \right\} \right) \quad (203)$$

さらにガウス積分を逆に利用したハバード・ストラトノビッチ変換を利用すれば、

$$\int D\mathbf{z} \prod_{a=1}^n \exp \left(\sqrt{\tilde{q}} \mathbf{z}^T \mathbf{x}^a - \frac{\tilde{q}}{2} (\mathbf{x}^a)^T \mathbf{x}^a \right) \quad (204)$$

を得る.

ハバード・ストラトノビッチ変換

ガウス積分の公式を逆に利用して、指数関数の肩の部分にある項を 1 次に減らすことができる.

$$\int D\mathbf{z} \exp \left(\sqrt{a} \mathbf{z}^T \mathbf{x} \right) = \exp \left(\frac{a}{2} \mathbf{x}^T \mathbf{x} \right) \quad (205)$$

代わりにガウス積分が増えることになるが、 \mathbf{x} が何かの和であるとか入り組んでいる場合に、1 次の項にすることで解きほぐすことが可能となるメリットがある.

最終的にエントロピーの項に現れる対数の内部にあるデルタ関数の積は、

$$\prod_{a,b} \delta \left(q^{ab} - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) = \exp \left(N \frac{n}{2} \tilde{Q} Q - N \frac{n(n-1)}{2} \tilde{q} q - N n \tilde{m} m \right) \\ \times \prod_{a=1}^n \int D\mathbf{z} \exp \left(-\frac{1}{2} (\tilde{Q} + \tilde{q}) (\mathbf{x}^a)^T \mathbf{x}^a + (\sqrt{\tilde{q}} \mathbf{z} + \tilde{m} \mathbf{x}_0)^T \mathbf{x}^a \right) \quad (206)$$

という形を持つ。ここで \mathbf{x}^a についての積分を考えると、 n 個の積は全く同等のものがあるので単純に積分の結果を n 乗してもかまわない。また \mathbf{x}^a の N 個の成分についても全く同等であるので積分の結果を N 乗してかまわない。最終的にエントロピーに関する積分部分は、

$$\int \prod_{a=1}^n dx^a \exp(-\beta \|\mathbf{x}^a\|_1) \prod_{a,b} \delta \left(q^{ab} - \frac{1}{N} (\mathbf{x}^a)^T \mathbf{x}^b \right) = \exp \left(N \frac{n}{2} \tilde{Q} Q - N \frac{n(n-1)}{2} \tilde{q} q - N n \tilde{m} m \right) \\ \times \exp \left\{ N n \log \phi(x_0, z; \{Q\}, \{\tilde{Q}\}) \right\} \quad (207)$$

ここでまとめて

$$\phi(x_0, z; \tilde{Q}, \tilde{q}, \tilde{m}) = \int dx \exp \left(-\frac{1}{2} (\tilde{Q} + \tilde{q}) x^2 + (\sqrt{\tilde{q}} z + \tilde{m} x_0) x - \beta |x| \right) \quad (208)$$

とおいた。指数の肩にすべて N がかかっているため、 $\tilde{Q}, \tilde{q}, \tilde{m}$ による鞍点評価をすれば良い。結局エントロピーは、 $\tilde{Q}, \tilde{q}, \tilde{m}$ による鞍点を用いて、

$$s(\rho, Q, m, q) = \max_{\tilde{Q}} \left\{ \frac{n}{2} \tilde{Q} Q - \frac{n(n-1)}{2} \tilde{q} q - n \tilde{m} m + n \left[\int_{\mathbf{x}_0} D\mathbf{z} \log \phi(x_0, z; \tilde{Q}, \tilde{q}, \tilde{m}) \right] \right\} \quad (209)$$

という格好となる。内部エネルギーもエントロピーも n についての1次の項があるため、レプリカ法の処方箋に乗っ取って、 n の1次の寄与を見れば確かに有益な情報が引き出せそう。残る問題は、 $\phi(x_0, z; \tilde{Q}, \tilde{q}, \tilde{m})$ の評価である。これは L_1 ノルム、つまり絶対値関数を含む積分であるので難しい。しかし $\beta \rightarrow \infty$ の極限をとることで、積分をせずに鞍点評価を行うことでこの問題点を回避することができる。

$\beta \rightarrow \infty$ の極限 やや天下りであるが、 β を有限に留めたままで計算を実行したのちに $\beta \rightarrow \infty$ としたときの以下の問題点

- $Q - q \sim O(1/\beta)$ で Q と q が近づく。
- $\tilde{Q} + \tilde{q} \sim O(\beta)$ 及び $\tilde{m} \sim O(\beta)$ 、 $\tilde{q} \sim O(\beta^2)$ で発散していく。

を解消するために、 $\beta(Q - q) \rightarrow \chi$ 、 $\tilde{Q} + \tilde{q} \rightarrow \beta \tilde{Q}$ 、 $\tilde{q} \rightarrow \beta^2 \tilde{\chi}$ 、 $\tilde{m} \rightarrow \beta \tilde{m}$ と変数変換を行う。内部エネルギーについては、 $\lambda \rightarrow +0$ も合わせてとると、

$$-e(\rho, Q, m, q) = -n \frac{\alpha \beta \rho - 2m + Q}{2 \chi} + O(1) \quad (210)$$

となる。一方エントロピーについては

$$s(\rho, Q, m, q) = n \beta \max_{\tilde{Q}} \left\{ \frac{1}{2} \tilde{Q} Q - \frac{1}{2} \tilde{\chi} \chi - \tilde{m} m - \left[\int_{\mathbf{x}_0} D\mathbf{z} \min_x \left\{ \frac{\tilde{Q}}{2} x^2 - (\sqrt{\tilde{\chi}} z + \tilde{m} x_0) x + |x| \right\} \right] \right\} \quad (211)$$

x_0 についての積分を実行して、 $\sqrt{\tilde{\chi}} z + \tilde{m} x_0 = \sqrt{\tilde{\chi} + \tilde{m} t}$ という変数変換を行うことにより、

$$s(\rho, Q, m, q) = n \beta \max_{\tilde{Q}} \left\{ \frac{1}{2} \tilde{Q} Q - \frac{1}{2} \tilde{\chi} \chi - \tilde{m} m - (1 - \rho) \int D\mathbf{z} \Phi(z; \tilde{Q}, \tilde{q}, 0) - \rho \int D\mathbf{t} \Phi(t; \tilde{Q}, \tilde{q}, \tilde{m}) \right\}$$

を得る。ここで

$$\Phi(z; \tilde{Q}, \tilde{q}, \tilde{m}) = \min_x \left\{ \frac{\tilde{Q}}{2} x^2 - \sqrt{\tilde{\chi} + \tilde{m}^2} z x + |x| \right\} \quad (212)$$

とおいた。この最小化問題は実は簡単に解くことができ、

$$\Phi(z; \tilde{Q}, \tilde{q}, \tilde{m}) = -\frac{1}{2\tilde{Q}} \left(\left| \sqrt{\tilde{\chi} + \tilde{m}^2} z \right| - 1 \right)^2 \Theta \left(\left| \sqrt{\tilde{\chi} + \tilde{m}^2} z \right| - 1 \right) \quad (213)$$

である。ここで

$$\Theta(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (214)$$

である。

[問:最小化問題を実際に解いてみよ。]

$\Phi(z; \tilde{Q}, \tilde{q}, \tilde{m})$ の z に関するガウス積分は丁寧に場合分けと部分積分を行えば実行できる。全ての結果をまとめると、1自由度あたりの自由エネルギー $-\beta f = \frac{1}{N} [\log Z]_{A, \mathbf{x}_0}$ をみると、

$$-f = \max_{\tilde{Q}, \tilde{q}} \left\{ \frac{\alpha}{2\tilde{\chi}} (\rho - 2m + Q) + \frac{1}{2} (Q\tilde{Q} - \tilde{\chi}) - m\tilde{m} + \frac{(1-\rho)}{\tilde{Q}} G(\tilde{\chi} + \tilde{m}^2) + \frac{\rho}{\tilde{Q}} G(\tilde{\chi}) \right\} \quad (215)$$

という表式を得る。ここで

$$H(a) = \int_a^\infty Dz \quad (216)$$

$$G(a) = (a+1)H\left(\frac{1}{\sqrt{a}}\right) - \sqrt{\frac{a}{2\pi}} \exp\left(-\frac{1}{2a}\right) \quad (217)$$

と定義した。あとは Q 、 \tilde{Q} についての鞍点を調べれば良いだけである。それぞれ偏微分することで以下の鞍点方程式を得る。

$$\tilde{Q} = \frac{\alpha}{\tilde{\chi}} \quad (218)$$

$$\tilde{\chi} = \frac{\alpha(\rho - 2m + Q)}{\tilde{\chi}^2} \quad (219)$$

$$\tilde{m} = \frac{\alpha}{\tilde{\chi}} \quad (220)$$

$$Q = \frac{2\rho}{\tilde{Q}^2} G(\tilde{\chi} + \tilde{m}^2) + \frac{2(1-\rho)}{\tilde{Q}^2} G(\tilde{\chi}) \quad (221)$$

$$\chi = \frac{2\rho}{\tilde{Q}} H\left(\frac{1}{\sqrt{\tilde{\chi} + \tilde{m}^2}}\right) + \frac{2(1-\rho)}{\tilde{Q}} H\left(\frac{1}{\sqrt{\tilde{\chi}}}\right) \quad (222)$$

$$m = 2\rho \frac{\tilde{m}}{\tilde{Q}} H\left(\frac{1}{\sqrt{\tilde{\chi} + \tilde{m}^2}}\right) \quad (223)$$

これを適当な初期条件のもと、反復代入を行うことで固定点を探す。パラメータ α と ρ について変化させると次の MSE が急激に変化するところが出現する。

$$\text{MSE} = \left[\left\langle \frac{1}{N} \|\mathbf{x} - \mathbf{x}_0\|_2^2 \right\rangle_{\mathbf{x}|A, \mathbf{x}_0}^{\beta \rightarrow \infty} \right]_{A, \mathbf{x}_0} = \rho - 2m + Q \quad (224)$$

その振る舞いにより基底追跡の相境界が明らかとなる (図 16)。このようにして統計力学的な処方箋により、基底追跡や LASSO 型の最適化問題の性能評価など、圧縮センシングの問題を解く際に利用される最適化問題の性質を明らかにすることができる。観測行列をガウス分布にしたがうランダム行列としたが、直交行列をランダムに選んだものの性能評価など実際に使われる圧縮センシングの問題に近い状況についても実行することができる。信号の特性や、ノイズが混入した場合など拡張も様々であり、習得するとよい技術である。

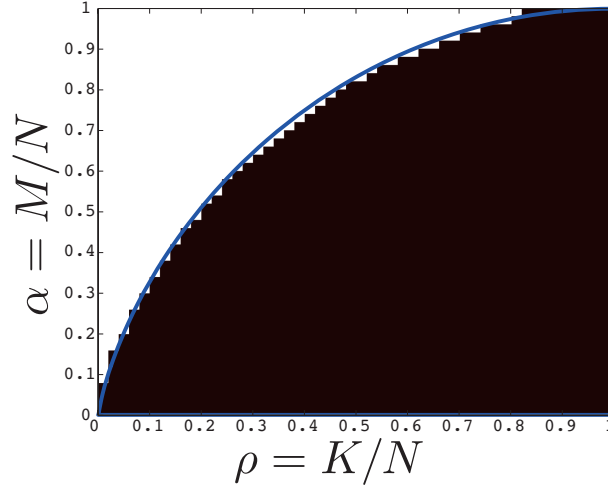


図 16: 基底追跡のレプリカ解析の結果. MSE が 0.001 を境目として、黒が MSE が大きい領域 (失敗相)、白が MSE が小さい領域 (成功相). 曲線は図 6 のもの.

5.3 近似メッセージ伝搬法

後に続くボルツマン機械学習でも登場する統計力学由来の推定アルゴリズムについて紹介する. 手法としての導出は後に譲るとして、ここでは圧縮センシングに利用した場合について紹介する.

統計力学はある観測行列、そして原信号をもってきたときに、推定できるかどうかの典型的な性能 (鞍点の意味で尤もらしい状態) を評価している. その意味で平均的な性質についてのみ焦点を当てている. ここで紹介するのは、それとは真逆で具体的にある観測行列、原信号をもってきたときに、それを推定するためのアルゴリズムである. しかも先に登場したメジャライザー最小化を基本とした ISTA に似た形のアルゴリズムが導出できるため、最適化問題と統計力学の関係性の意味でも非常に興味深い内容を含む.

まずメッセージ伝搬法 (信念伝搬法) について、導出は後に譲るとして、天下り的に導入する. 確率変数間の関係をファクターノードと変数ノードによってわけてグラフ上で表現する. このときファクターノードを μ とギリシャ文字で表し、変数ノードをアルファベットで表現する. ファクターノード μ についている変数ノード k は、 $k \in \partial\mu$ と表示する. 一方変数ノード k についているファクターノード μ は、 $\mu \in \partial k$ と表記する. 変数ノードは確率変数の要素を表しており、ファクターノードは、確率変数間の関係を記述する.

そのようなグラフ表現をすると確率分布関数が以下のように分けられる.

$$P(\mathbf{x}) = \prod_{k=1}^N f_k(x_k) \prod_{\mu} f_{\mu}(\mathbf{x}_{\partial\mu}) \quad (225)$$

このときに各確率変数の期待値、相関を計算するのに有用な周辺分布は以下のメッセージによる表現を持つ.

$$P_k(x_k) = f_k(x_k) \prod_{\mu \in \partial k} \tilde{M}_{\mu \rightarrow k}(x_k) \quad (226)$$

$$P_{\mu}(\mathbf{x}_{\partial\mu}) = f_{\mu}(\mathbf{x}_{\partial\mu}) \prod_{k \in \partial\mu} M_{k \rightarrow \mu}(x_k) \quad (227)$$

ここで $\tilde{M}_{\mu \rightarrow k}$ と $M_{k \rightarrow \mu}$ はメッセージと呼ばれる. このメッセージは以下の反復方程式を経由して構成される.

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int d\mathbf{x}_{/k} f_{\mu}(\mathbf{x}_{\partial\mu}) \prod_{l \neq k \in \partial\mu} M_{l \rightarrow \mu}^{[t]}(x_l) \quad (228)$$

$$M_{k \rightarrow \mu}^{[t]}(x_k) \propto f_k(x_k) \prod_{\nu \neq \mu \in \partial k} \tilde{M}_{\nu \rightarrow k}^{[t-1]}(x_k) \quad (229)$$

これはボルツマン機械学習の項で述べられるように、確率分布関数の強い複数の確率変数間の相関関係を断ち切る近似を行うことで得られる方法である。

さて我々の目的とする圧縮センシングの原信号推定に利用してみよう。まずは基底追跡型の最適化問題について考えよう。観測行列 A 、観測信号 \mathbf{y} が与えられたときの事後確率はベイズの定理を利用して、以下のように与えられる。

$$P^\beta(\mathbf{x}|A, \mathbf{y}) \propto \prod_{k=1}^N \exp(-\beta|x_k|) \prod_{\mu=1}^M \delta(y_\mu - \mathbf{a}_\mu^T \mathbf{x}) \quad (230)$$

ファクターノード μ にたいして、変数ノードは全て繋がっている。また \mathbf{a}_μ は観測行列の行ベクトルを抜き取ったものである。 $\beta \rightarrow \infty$ の極限において、 L_1 ノルム最小化を利用して、スパースな解を得るという格好をしているのはこれまで通りである。制約条件を表すデルタ関数部分を罰金法にあるように、制約条件から決まるガウス分布で表現することも可能である。そこで一般に以下の形で与えられる確率分布関数を以降取り扱うことにする。

$$P^\beta(\mathbf{x}|A, \mathbf{y}) \propto \prod_{k=1}^N \exp(-\beta|x_k|) \prod_{\mu=1}^M f(\mathbf{a}_\mu^T \mathbf{x} | y_\mu) \quad (231)$$

そのままメッセージ伝搬法を利用しても良いが、更に変数の数が多い ($N \rightarrow \infty$) という問題の性質を利用して、近似を施すことにより計算量の更に低いアルゴリズムを構成することができる。これを**近似メッセージ伝搬法** (Approximate Message Passing:AMP) と呼ぶ。

それでは近似メッセージ伝搬法の導出を行う。 $N \rightarrow \infty$ の極限での計算であるから、先ほどの統計力学による解析手法が参考になる。まずメッセージの更新式 (228) の計算を進める。以下の恒等式を利用する。

$$1 = \int du_\mu \delta(u_\mu - \mathbf{a}_\mu^T \mathbf{x}) \quad (232)$$

この恒等式をメッセージ更新式 (228) に挿入しよう。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int du_\mu \int d\mathbf{x}_{/k} f(u_\mu | y_\mu) \delta(u_\mu - \mathbf{a}_\mu^T \mathbf{x}) \prod_{l \neq k \in \partial \mu} M_{l \rightarrow \mu}^{[t]}(x_l) \quad (233)$$

さらにデルタ関数をフーリエ積分表示 (後の便宜のため虚数単位を用いた表示) をする。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int du_\mu \int d\tilde{u}_\mu \int d\mathbf{x}_{/k} f(u_\mu | y_\mu) \exp\{i\tilde{u}_\mu (u_\mu - \mathbf{a}_\mu^T \mathbf{x})\} \prod_{l \neq k \in \partial \mu} M_{l \rightarrow \mu}^{[t]}(x_l) \quad (234)$$

積分される変数に注目して、 x_l についての積分の積として表示する。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int du_\mu \int d\tilde{u}_\mu f(u_\mu | y_\mu) \exp\{i\tilde{u}_\mu (u_\mu - a_{\mu k} x_k)\} \prod_{l \neq k \in \partial \mu} \int dx_l \exp(-i\tilde{u}_\mu a_{\mu l} x_l) M_{l \rightarrow \mu}^{[t]}(x_l) \quad (235)$$

ここで観測行列は慣例に従い、平均 0、分散 $1/M$ のガウス分布に従うものを仮定する。そのとき $a_{\mu l}$ が分散 $1/M$ 程度の微小な値を持つことをから、 x_l に関する積分の中身を展開する。 [t] その際に以下の平均と分散を定義する。

$$m_{k \rightarrow \mu}[t] \equiv \int dx_k x_k M_{k \rightarrow \mu}^{[t]}(x_k) \quad (236)$$

$$V_{k \rightarrow \mu}[t] \equiv \int dx_k x_k^2 M_{k \rightarrow \mu}^{[t]}(x_k) - (m_{k \rightarrow \mu}[t])^2 \quad (237)$$

これにより x_l に関する積分を $a_{\mu k}$ について 2 次まで展開したのちに実行したつもりになり、再び指数関数の肩に引き上げる。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int du_\mu \int d\tilde{u}_\mu f(u_\mu | y_\mu) \exp\{i\tilde{u}_\mu (u_\mu - a_{\mu k} x_k)\} \prod_{l \neq k \in \partial \mu} \exp\left(-i\tilde{u}_\mu a_{\mu l} m_{l \rightarrow \mu}[t] - \frac{\tilde{u}_\mu^2}{2} a_{\mu l}^2 V_{l \rightarrow \mu}[t]\right) \quad (238)$$

ここで \tilde{u}_μ についてガウス分布関数となっていることに注目して、積分を実行する。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto \int du_\mu f(u_\mu | y_\mu) \exp \left\{ -\frac{1}{2V_\mu} (u_\mu - m_\mu[t] + a_{\mu k}(x_k - m_{k \rightarrow \mu}[t]))^2 \right\}$$

ここで

$$m_\mu[t] \equiv \sum_k a_{\mu k} m_{k \rightarrow \mu}[t] \quad (239)$$

$$V_\mu[t] \equiv \sum_k a_{\mu k}^2 V_{k \rightarrow \mu}[t] \quad (240)$$

と定義した。途中 $\sum_{l \neq k} m_{l \rightarrow \mu}[t] = m_\mu[t] - m_{k \rightarrow \mu}[t]$ 、 $\sum_{l \neq k} V_{l \rightarrow \mu}[t] = V_\mu[t] - V_{k \rightarrow \mu}[t]$ として丁寧に主要に寄与する項を考慮しながら計算する。さらに $a_{\mu k}$ が小さいので、更に展開をする。

$$\begin{aligned} \tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) &\propto \int du_\mu f(u_\mu | y_\mu) \exp \left\{ -\frac{1}{2V_\mu[t][t]} (u_\mu - m_\mu[t])^2 \right\} \\ &\quad \times \left\{ 1 + a_{\mu k}(x_k - m_{k \rightarrow \mu}[t]) \frac{(u_\mu - m_\mu[t])}{V_\mu[t]} - \frac{a_{\mu k}^2}{2} (x_k - m_{k \rightarrow \mu}[t])^2 \left\{ \frac{1}{V_\mu[t]} - \frac{(u_\mu - m_\mu[t])^2}{(V_\mu[t])^2} \right\} \right\} \end{aligned} \quad (241)$$

ここで u_μ についての積分を実行する。そこで以下のモーメントを定義する。

$$c_\mu^{(k)} = \int du_\mu f(u_\mu | y_\mu) \exp \left\{ -\frac{1}{2V_\mu[t]} (u_\mu - m_\mu[t])^2 \right\} \left(\frac{u_\mu - m_\mu[t]}{V_\mu[t]} \right)^k \quad (242)$$

このモーメントを利用してメッセージは以下のようにガウス分布関数となることがわかる。

$$\tilde{M}_{\mu \rightarrow k}^{[t]}(x_k) \propto c_\mu^{(0)} \exp \left\{ m_{\mu \rightarrow k}[t] x_k - \frac{1}{2} V_{\mu \rightarrow k}[t] x_k^2 \right\}$$

ここでメッセージによる平均と分散にかかわる x_k の 1 次、2 次の係数は

$$m_{\mu \rightarrow k}[t] = a_{\mu k} g_0(m_\mu[t] | V_\mu[t]) - a_{\mu k}^2 g_1(m_\mu[t] | V_\mu[t]) m_{k \rightarrow \mu}[t] \quad (243)$$

$$V_{\mu \rightarrow k}[t] = -a_{\mu k}^2 g_1(m_\mu[t] | V_\mu[t]) \quad (244)$$

であり、 $g_r(m_\mu | V_\mu)$ は以下のように定義した。

$$g_r(m_\mu | V_\mu) = \frac{\partial^r}{\partial m_\mu^r} \log c_\mu^{(0)} \quad (245)$$

この量は $f(u_\mu | y_\mu)$ を (242) 式のようにガウス積分することで得られる。つまりメッセージ更新式 (228) については評価が終わっている。この中に登場する m_μ 、 V_μ の評価のためにもう片方のメッセージ更新式 (229) に目を向ける。ここでメッセージによるモーメント

$$d_{k \rightarrow \mu}^{(r)} = \int dx_k x_k^r M_{k \rightarrow \mu}^{[t-1]}(x_k) \quad (246)$$

を計算することを考えてみよう。 $r = 1$ のとき $m_{k \rightarrow \mu}[t-1]$ 、 $r = 2$ を利用して $V_{k \rightarrow \mu}[t-1]$ を得ることができ、ここでメッセージ更新式の定義 (229) をみると、評価するべきは以下の積分であることがわかる。

$$d_{k \rightarrow \mu}^{(r)} = \int dx_k x_k^r \exp \left(-\beta |x_k| + \sum_{\nu \neq \mu} m_{\nu \rightarrow k}[t-1] x_k - \frac{1}{2} V_{\nu \rightarrow k}[t-1] x_k^2 \right) \quad (247)$$

例によって絶対値関数があるため積分の実行は難しい。しかしながら $\beta \rightarrow \infty$ を考えるため、積分するする詐欺で、鞍点評価に逃げるができる。そこで $m_{\nu \rightarrow k}[t-1] \rightarrow \beta m_{\nu \rightarrow k}[t-1]$ 、 $V_{\nu \rightarrow k}[t-1] \rightarrow \beta V_{\nu \rightarrow k}[t-1]$

とスケールを変更する. さらに

$$m_k[t-1] = \sum_{\mu} m_{\mu \rightarrow k}[t-1] \quad (248)$$

$$V_k[t-1] = \sum_{\mu} V_{\mu \rightarrow k}[t-1] \quad (249)$$

を定義する. レプリカ法による計算のときと同様な計算にするため, 先んじて以下の量を計算する.

$$\begin{aligned} I(a|b) &= \log \left\{ \int dx \exp \left(-\beta|x| + \beta ax - \frac{1}{2}\beta bx^2 \right) \right\} \\ &\approx \frac{\beta}{2b} (|a| - 1)^2 \Theta(|a| - 1) \end{aligned} \quad (250)$$

ここから $d_{k \rightarrow \mu}^{(r)} = (\partial/\partial\beta a)^r I(a|b)$ より,

$$\begin{aligned} m_{k \rightarrow \mu}[t] &= S_{1/(V_k[t-1] - V_{\mu \rightarrow k}[t-1])} \left(\frac{m_k[t-1] - m_{\mu \rightarrow k}[t-1]}{V_k[t-1] - V_{\mu \rightarrow k}[t-1]} \right) \\ &\approx S_{1/V_k[t-1]} \left(\frac{m_k[t-1]}{V_k[t-1]} \right) - \frac{m_{\mu \rightarrow k}[t-1]}{V_k[t-1]} \Theta(|m_k[t-1]| - 1) \end{aligned} \quad (251)$$

$$\beta V_{k \rightarrow \mu}[t] = \frac{1}{V_k[t-1] - V_{\mu \rightarrow k}[t-1]} \Theta(|m_k[t-1] - m_{\mu \rightarrow k}[t-1]| - 1) \approx \frac{1}{V_k[t-1]} \Theta(|m_k[t-1]| - 1). \quad (252)$$

これで一般的に確率分布関数 $f(u_{\mu}|y_{\mu})$ の形が与えられたとき, メッセージをそれぞれガウス分布関数に近似したうえで, その1次と2次の係数についての関係式を以下のように得る.

$$m_k[t] = \frac{1}{\beta} \sum_{\mu=1}^M a_{\mu k} g_0(m_{\mu}[t]|V_{\mu}[t]) - \frac{1}{\beta} \sum_{\mu=1}^M a_{\mu k}^2 g_1(m_{\mu}[t]|V_{\mu}[t]) S_{1/V_k[t-1]} \left(\frac{m_k[t-1]}{V_k[t-1]} \right) \quad (253)$$

$$V_k[t] = -\frac{1}{\beta} \sum_{\mu=1}^M a_{\mu k}^2 g_1(m_{\mu}[t]|V_{\mu}[t]) \quad (254)$$

$$m_{\mu}[t] = \sum_k a_{\mu k} S_{1/V_k[t-1]} \left(\frac{m_k[t-1]}{V_k[t-1]} \right) - V_{\mu}[t] g_0(m_{\mu}[t-1]|V_{\mu}[t-1]) \quad (255)$$

$$\beta V_{\mu}[t] = \sum_k a_{\mu k}^2 \frac{1}{V_k[t-1]} \Theta(|m_k[t-1]| - 1). \quad (256)$$

LASSO に対する近似的メッセージ伝搬法 さて LASSO 型の最適化問題に対する近似メッセージ伝搬法のアルゴリズムを書いてみよう. それは $f(u_{\mu}|y_{\mu})$ を

$$f(u_{\mu}|y_{\mu}) = \exp \left\{ -\frac{\beta}{2\lambda} (y_{\mu} - u_{\mu})^2 \right\} \quad (257)$$

とおけばよい. 式 (245) に従い, まず $c_{\mu}^{(0)}$ を計算すると,

$$g_0(m_{\mu}|V_{\mu}) = \frac{\beta}{\lambda + \beta V_{\mu}} (y_{\mu} - m_{\mu}) \quad (258)$$

$$g_1(m_{\mu}|V_{\mu}) = -\frac{\beta}{\lambda + \beta V_{\mu}} \quad (259)$$

であるから $\beta V_\mu \rightarrow \chi_\mu$ として、

$$m_k[t] = \sum_{\mu=1}^M \frac{a_{\mu k}}{\lambda + \chi_\mu[t]} (y_\mu - m_\mu[t]) + V_k[t] S_{1/V_k[t-1]} \left(\frac{m_k[t-1]}{V_k[t-1]} \right) \quad (260)$$

$$V_k[t] = \sum_{\mu=1}^M \frac{a_{\mu k}^2}{\lambda + \chi_\mu[t]} \quad (261)$$

$$m_\mu[t] = \sum_k a_{\mu k} S_{1/V_k[t-1]} \left(\frac{m_k[t-1]}{V_k[t-1]} \right) - \chi_\mu[t] g_0(m_\mu[t-1] | \chi_\mu[t-1]) \quad (262)$$

$$\chi_\mu[t] = \sum_k a_{\mu k}^2 \frac{1}{V_k[t-1]} \Theta(|m_k[t-1]| - 1). \quad (263)$$

ここでさらに近似を行う。(いわゆる近似メッセージ伝搬法は、この近似を行ったもの。上記は正確には緩和信念伝搬法 (relaxed Belief Propagation) と呼ぶ。) 大数の法則 (非常に多くの確率変数の和からなる量は期待値に収束する) より $\sum_\mu a_{\mu k}^2 = 1$ および $a_{\mu k}^2 = 1/M$ としてもよい。ここで ISTA 等前述したアルゴリズムとの対応をより見やすくするために、 $\sum_\mu a_{\mu k}^2 = L$ 、 $a_{\mu k}^2 = L/M$ とおく。さらに $\chi_\mu[t] = \chi[t]$ および $V_k[t] = V[t]$ とする。

$$m_k[t] = \sum_{\mu=1}^M \frac{a_{\mu k}}{\lambda + \chi[t]} (y_\mu - m_\mu[t]) + V[t] S_{1/V[t-1]} \left(\frac{m_k[t-1]}{V[t-1]} \right) \quad (264)$$

$$V[t] = \frac{L}{\lambda + \chi[t]} \quad (265)$$

$$m_\mu[t] = \sum_k a_{\mu k} S_{1/V[t-1]} \left(\frac{m_k[t-1]}{V[t-1]} \right) - \frac{\chi[t]}{\lambda + \chi[t-1]} (y_\mu - m_\mu[t-1]) \quad (266)$$

$$\chi[t] = \frac{L}{M} \sum_{k=1}^N \frac{1}{V[t-1]} \Theta(|m_k[t-1]| - 1) \quad (267)$$

それぞれガウス分布関数の係数をあらわしている。そこで期待値に軟判定しきい値関数をくぐらせたものから各成分の信号を推定することにしよう。

$$x_k[t] = S_{\Lambda[t]} \left(\frac{m_k[t]}{V[t]} \right) \quad (268)$$

この各時刻の推定値に関する反復方程式であるとみて、式変形を試みる。

$$x_k[t+1] = S_{\Lambda[t]} \left(x_k[t] + \frac{1}{L} \sum_{\mu=1}^M a_{\mu k} z_\mu[t] \right) \quad (269)$$

$$z_\mu[t] = (y_\mu - \mathbf{a}_\mu^\top \mathbf{x}[t]) + \frac{K[t-1]}{M} z_\mu[t-1] \quad (270)$$

$$\Lambda[t] = \frac{\lambda}{L} + \frac{\Lambda[t-1]}{M} K[t-1] \quad (271)$$

ここで

$$K[t] = \sum_{k=1}^N \Theta \left(\left| x_k[t] + \frac{1}{L} \sum_{\mu=1}^M a_{\mu k} z_\mu[t] \right| - \Lambda[t] \right) \quad (272)$$

また $V[t] = 1/\Lambda[t]$ とした。上記2つの反復方程式を称して、圧縮センシングの再構成アルゴリズムとしての有名な AMP と呼ぶ。3つめの反復式 (271) は、圧縮センシングの再構成の可能性を表現する重要な式である。(その割にあまり目にしない。) 第2項は、 N 成分あるうち非零の成分がいくつあるのかを数えている。そのため復号される信号の非零成分の個数 $K[t]$ が $K[t] < M$ であれば、基本的には1より小さい係数が $\Lambda[t-1]$ にかかる。そのため $\Lambda[t]$ は収束することが期待される。一方で非零成分の個数が $K[t] > M$ で

あれば発散してしまうため復号ができない。\$z_\mu[t]\$ は各時刻での各行における残差 \$y_\mu - \mathbf{a}_\mu^T \mathbf{x}[t]\$ に由来する項であり、第一項までとすれば **ISTA と一致する**ことに注意したい。統計力学由来の AMP により更に非自明な補正項が登場する。\$\lambda \to 0\$ とすれば LASSO 型から基底追跡型の最小化問題に対応するが、一般には ISTA による計算は不安定なものとなる。しかしこの補正項により、ISTA の計算不安定性を弱める性質があることが知られている。その意味で AMP は ISTA より有用であり盛んに応用的な研究やその性質についての理論的研究が進められてきた。

5.4 状態発展法

AMP は ISTA をひな形として、補正項を付け加えることにより安定な動作をする。それでは果たして、その収束先はレプリカ法で議論したものと一致するかどうか？非常に気になるところである。反復法により得られた結果が、どれだけ知りたい情報と一致しているかを表す量に注目して、その時間発展を追うことにより、反復法の性能を知ることができる。

ここでは基底追跡法に対する ISTA を例にとって解析してみよう。\$A\$ を各成分、平均 0、分散 \$1/M\$ の独立同分布のガウス分布から生成される場合を考える。このときリプシッツ定数は \$L = 1\$ である。その ISTA の更新式は以下の通りである。

$$\mathbf{x}[t+1] = S_\lambda(\mathbf{x}[t] + A^T \mathbf{z}[t]) \quad (273)$$

$$\mathbf{z}[t] = \mathbf{y} - A\mathbf{x}[t] \quad (274)$$

ここで \$\mathbf{x}[t]\$ を \$\mathbf{y} = A\mathbf{x}_0[t]\$ であることを使い、正解 \$\mathbf{x}_0\$ と推定された解 \$\mathbf{x}[t]\$ との関係が明らかとなるように書き換えてみよう。

$$\mathbf{x}[t+1] = S_\lambda(\mathbf{x}_0[t] + (A^T A - I)(\mathbf{x}[t] - \mathbf{x}_0)). \quad (275)$$

ここで大数の法則により、非常に大きいシステムサイズ \$N \to \infty\$ では、\$A^T A - I\$ は対角成分、非対角成分共に分散 \$1/M\$ のガウス分布に従う。また正解と推定された解の間の平均二乗誤差 (MSE) を

$$(\sigma[t])^2 = \frac{1}{N} \|\mathbf{x}[t] - \mathbf{x}_0\|_2^2 \quad (276)$$

とおくと、平均 0 分散 1 のガウス分布に従う乱数 \$\mathbf{w}\$ で \$\mathbf{x}[t] - \mathbf{x}_0 = \sigma[t]\mathbf{w}\$ と書くことができる。これらの事実を使うと、式 (275) から MSE の更新式を導くことができる。

$$(\sigma[t+1])^2 = \Psi((\sigma[t])^2) \quad (277)$$

ここで

$$\Psi((\sigma[t])^2) = \int D\mathbf{w} \left\| S_\lambda \left(\mathbf{x}_0[t] + \sqrt{\frac{N}{M}} \sigma[t]\mathbf{w} \right) - \mathbf{x}_0 \right\|_2^2 \quad (278)$$

である。平均二乗誤差が様々な観測結果に対してどのように振る舞うのかを評価するには、経験平均を評価すれば良い。ここで更に大数の法則を利用することで、経験平均の代わりに期待値を計算することにしよう。\$\mathbf{x}[t]\$ については \$\mathbf{y}\$ すなわち \$\mathbf{x}_0\$ により完全に決定されるので、その統計性は正解 \$\mathbf{x}_0\$ にゆだねられる。そのため \$\mathbf{x}_0\$ についての期待値を取ることが望ましく、

$$\Psi((\sigma[t])^2) = \int d\mathbf{x}_0 P(\mathbf{x}_0) \int D\mathbf{w} \left\| S_\lambda \left(\mathbf{x}_0[t] + \sqrt{\frac{N}{M}} \sigma[t]\mathbf{w} \right) - \mathbf{x}_0 \right\|_2^2 \quad (279)$$

とすれば良い。ISTA の性能解析がこの非線形関数の性質で決まる単純な格好となる。この非線形関数が、平均二乗誤差 \$(\sigma[t])^2 = 0\$ 付近で安定かどうかを調べれば、誤差のない完全な再構成が可能かどうかの領域が定められる。

$$\left. \frac{d}{d\sigma^2} \Psi(\sigma^2) \right|_{\sigma^2=0} < 0 \quad (280)$$

このように平均二乗誤差の変化を来たい値の計算によって調べる方法を状態発展法 (state evolution)、または密度発展法 (density evolution) 等と呼ぶ。レプリカ法による評価よりも容易であるため、最適化問題で用いられる反復計算の性能限界を調べるために重宝される方法である。実際の計算は、定義通りに従い代入すれば良い。

6 ボルツマン機械学習

同じ猫の画像であったとしても、画像データは様々な形態をもつ。しかしながら人間が知覚しているように、様々な猫の画像を見ても、猫である。これはどういうことだろうか。人間は自然と、高次元のデータが与えられたときに、そのデータの本質的な構造を抽出しているということだ。その結果猫であると認識していると考えられる。ボルツマン機械学習は、多次元ベクトルに過ぎない高次元データが与えられたときに、そのデータを創りだしている本質部分である、**生成モデル**を明らかにする枠組みである。この生成モデルが明らかとなれば、同様の性質を有する高次元データを自在に用意することが可能であり、また生成モデルそのものの特徴から、データの背後にある本質に迫ることができる。

ビッグデータ時代と呼ばれて久しい現代において、データを取得する量は増加する一方である。しかし大量のデータを得るだけ得ておいて、何をしたら良いかわからない、といった声も一方で聞かれる。何もできないという状況を打破するひとつのツールとしてボルツマン機械学習をここでは学習してみよう。

6.1 問題設定

計測技術や信号処理技術、そして情報処理そのものの質の向上により、我々は大量のデータを取得することが可能となった。\$N\$ 次元のベクトル \$\mathbf{x}^{(d)}\$ で表される大量のデータ (\$d = 1, 2, \dots, D\$) が与えられたときに、そのデータの起源を少数の説明変数で表した生成モデルを解明する処方箋のひとつがボルツマン機械学習である。ここで大前提として、データは確率的に出力されて得られているものとする。

ボルツマン機械学習では、統計力学で基本となるカノニカル分布に従ってデータが出力されると考える。

$$P(\mathbf{x}|\mathbf{u}) = \frac{1}{Z(\mathbf{u})} \exp\{-E(\mathbf{x}|\mathbf{u})\}. \quad (281)$$

ここで \$Z(\mathbf{u})\$ は分配関数、\$\mathbf{u}\$ はデータの構造を表すエネルギー関数 \$E(\mathbf{x}|\mathbf{u})\$ を形作るパラメータである。例えば \$N\$ 次元の 2 値データがイジング模型のカノニカル分布から生成されたと仮定する場合には、

$$E(\mathbf{x}|\mathbf{u}) = \sum_{i=1}^N \sum_{j \in \partial i} J_{ij} x_i x_j + \sum_{i=1}^N h_i x_i \quad (282)$$

とする。\$x_i = \pm 1\$ であり、パラメータ \$\mathbf{u} = (J, \mathbf{h})\$ によりエネルギー関数が特徴づけられている。\$j \in \partial i\$ は、\$J_{ij} \neq 0\$ で繋がっている成分 \$i\$ に関与する \$j\$ についての和である。他にも \$N\$ 次元の連続値を持つデータがガウス分布から生成されたと仮定する場合には、

$$E(\mathbf{x}|\mathbf{u}) = \frac{1}{2} \mathbf{x}^T J \mathbf{x} + \mathbf{h}^T \mathbf{x} \quad (283)$$

とする。\$x_i\$ は連続値であり、やはりパラメータ \$\mathbf{u} = (J, \mathbf{h})\$ によりエネルギー関数が特徴づけられている。上記の仮定のもと、与えられた大量のデータの経験分布

$$P_D(\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D \delta(\mathbf{x} - \mathbf{x}^{(d)}) \quad (284)$$

に最も近いカノニカル分布を探してくることがボルツマン機械学習の目標である (図 17)。その結果、パラメータ \$\mathbf{u}\$ によりデータの経験分布を「もっともらしく」再現する確率分布を得ることができる。パラメータそのものからデータの特徴を調べることも可能である。

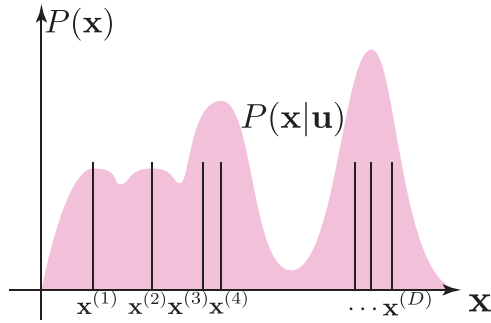


図 17: データの経験分布に合うボルツマン分布を求めるボルツマン機械学習.

6.2 最尤法

さてそうすると 2 つの異なる確率分布を持ってきたときに、それらが近いか遠いかを調べるための計量が必要だ。最も一般的に用いられるのがカルバック・ライブラー (KL) 情報量である。

$$D_{\text{KL}}(P|Q) = \int dx P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (285)$$

この KL 情報量の意味で、データの経験分布に最も近い確率分布を与えるパラメータ \mathbf{u} を求めてみよう。 $Q(x)$ を未知のパラメータを持つ $P(x|\mathbf{u})$ に、 $P(x)$ をデータの経験分布 $P_D(x)$ としよう。このとき KL 情報量は、

$$D_{\text{KL}}(P_D|P_{\mathbf{u}}) = \int dx P_D(x) \log \left(\frac{P_D(x)}{P(x|\mathbf{u})} \right) \quad (286)$$

となる。パラメータ \mathbf{u} が関与する部分に注目すると、以下の最大化問題と等価であることが分かる。

$$\mathbf{u}^* = \arg \max_{\mathbf{u}} L(\mathbf{u}). \quad (287)$$

ここで $L(\mathbf{u})$ は対数尤度関数 (の経験平均) と呼び、以下のように定義される。

$$L(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \log P(\mathbf{x} = \mathbf{x}^{(d)}|\mathbf{u}). \quad (288)$$

ベイズ推定の説明に現れた尤度関数が再び登場した。ボルツマン機械学習は、この対数尤度関数の最大化 (最尤法) を行うことで、得られたデータに適合する「もっともらしい」パラメータの推定を行うとも言い換えることができる。少し計算してみると明らかとなるが、対数尤度関数は、

$$L(\mathbf{u}) = -\frac{1}{D} \sum_{d=1}^D E(\mathbf{x} = \mathbf{x}^{(d)}|\mathbf{u}) - \log Z(\mathbf{u}). \quad (289)$$

という形を持つ。第一項がエネルギー関数 (の経験平均) の符号を変えたものであり、第二項が分配関数の対数であるから、自由エネルギーである。統計力学の言葉でいえば、自由エネルギーとエネルギー (温度は 1 として) の差であるエントロピーが最大となるようなエネルギー関数の形を求めることがボルツマン機械学習の中心的な問題であることが分かる。

それでは最尤法を実行するために、お得意の勾配法を試してみよう。

$$\mathbf{u}[t+1] = \mathbf{u}[t] + \eta \frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} \quad (290)$$

ここで η は学習係数と呼ばれる量で、小さければ小さいほど正確であるが計算時間の長大化に繋がるのでほどよい値をとることが要求される。対数尤度関数の微分が必要となるので、パラメータ \mathbf{u} について対数

尤度関数の微分を取ってみる.

$$\frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = -\frac{1}{D} \sum_{d=1}^D \frac{\partial E(\mathbf{x} = \mathbf{x}^{(d)}|\mathbf{u})}{\partial \mathbf{u}} + \left\langle \frac{\partial E(\mathbf{x}|\mathbf{u})}{\partial \mathbf{u}} \right\rangle_{\mathbf{u}}. \quad (291)$$

第1項はエネルギー関数の形を知っていれば評価は容易である. データに関する経験平均をとるだけだ. 一方第2項は熱平均の計算 $\langle \dots \rangle_{\mathbf{u}} = \sum_{\mathbf{x}} \dots \times P(\mathbf{x}|\mathbf{u})$ が必要となる. ただ一度きりの計算であれば, やる気にもなるだろう. しかし勾配法の手続きは, 適当な初期条件 $\mathbf{u}[0]$ のもと, 以下の反復計算を行うため何度も何度も勾配の計算, 熱平均の計算が必要であることが分かる (図 18). これがボルツマン機械学習の運

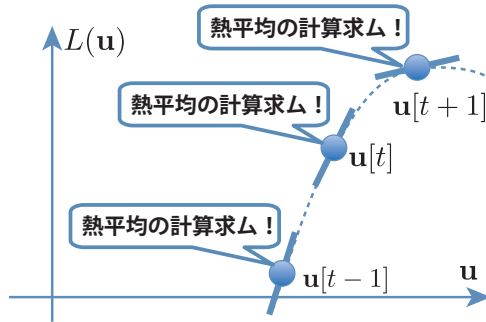


図 18: 最尤法の様子

命を決めた. 正確な実行のために必要な計算量が非常に膨大となり, 現実的な学習法ではないと, かつてその黎明期では判断された. しかしながら 2000 年代に入ってから, データ数の大規模化と新しい計算手法の提案を契機に現実的な計算手法であると受け入れられて今日に至る. その代表的なものを紹介していこう.

6.3 ボルツマン機械学習の学習方法

ボルツマンの名前を冠する通り, 統計力学の手法による学習方法がいくつか提案されている. それぞれに得手不得手があるので, 利用するにはそこに注意してほしい. 話を具体的にするため, エネルギー関数をイジング模型のものにした場合 (282) を考えよう. この場合必要となる熱期待値は,

$$\left\langle \frac{\partial E(\mathbf{x}|\mathbf{u})}{\partial J_{ij}} \right\rangle_{\mathbf{u}} = \langle x_i x_j \rangle_{\mathbf{u}} \quad (292)$$

$$\left\langle \frac{\partial E(\mathbf{x}|\mathbf{u})}{\partial h_i} \right\rangle_{\mathbf{u}} = \langle x_i \rangle_{\mathbf{u}} \quad (293)$$

である. つまり統計力学の言葉で言えば相関関数と局所磁化である.

平均場近似 (Mean Field Approximation) 熱平均または分配関数を計算するには計算量困難を伴うので, いっそのこと近似で諦めるとしよう. 熱平均や分配関数の計算で困難さの原因となるのは確率変数間の相関である. そこでその変数間の相関がない独立な分布で近似するとしよう.

$$P(\mathbf{x}|\mathbf{u}) \approx \prod_{i=1}^N P_i(x_i|\mathbf{u}) \quad (294)$$

ただしその近似はできる限りボルツマン分布を利用した場合に近いものにしたい. そこで KL 情報量を利用して, ボルツマン分布にできるだけ近くなるように $P_i(x_i|\mathbf{u})$ の関数形を決めよう.

$$P_i^{\text{MF}}(x_i|\mathbf{u}) = \arg \min_{P_i(x_i|\mathbf{u})} \left\{ \int d\mathbf{x} \prod_{i=1}^N P_i(x_i|\mathbf{u}) \log \left(\frac{\prod_{i=1}^N P_i(x_i|\mathbf{u})}{P(\mathbf{x}|\mathbf{u})} \right) \right\} \quad (295)$$

ここで $\int dx_i P_i(x_i|\mathbf{u}) = 1$ という等式制約を課す。この等式制約を考慮してラグランジュ未定乗数を導入して新たなコスト関数を導入すると、

$$P_i^{\text{MF}}(x_i|\mathbf{u}) = \arg \min_{P_i(x_i|\mathbf{u})} \left\{ \int dx \prod_{i=1}^N P_i(x_i|\mathbf{u}) \log \left(\frac{\prod_{i=1}^N P_i(x_i|\mathbf{u})}{P(\mathbf{x}|\mathbf{u})} \right) + \sum_{i=1}^N \lambda_i P_i(x_i|\mathbf{u}) \right\} \quad (296)$$

となる。このコスト関数を定義に従い、少し変形してみる。

$$\sum_{i=1}^N \int dx_i P_i(x_i|\mathbf{u}) \log P_i(x_i|\mathbf{u}) + \int dx \prod_{i=1}^N P_i(x_i|\mathbf{u}) E_i(x_i|\mathbf{x}_{/i}, \mathbf{u}) + \log Z(\mathbf{u}) + \sum_{i=1}^N \lambda_i P_i(x_i|\mathbf{u}) \quad (297)$$

ここで $E_i(x_i|\mathbf{x}_{/i}, \mathbf{u})$ はエネルギー関数を以下のように和の形で分解したものである。 $/i$ は添字 i 以外のものを表している。

$$E(\mathbf{x}|\mathbf{u}) = \sum_{i=1}^N E_i(x_i|\mathbf{x}_{/i}, \mathbf{u}) \quad (298)$$

具体例を書いてみると、イジング模型のエネルギー関数の場合は

$$E_i(x_i|\mathbf{x}_{/i}, \mathbf{u}) = \sum_{j \in \partial i} J_{ij} x_i x_j + h_i x_i \quad (299)$$

である。さて $P_i(x_i|\mathbf{u})$ について変分をとると、

$$0 = \log P_i(x_i|\mathbf{u}) + 1 + E_i(x_i|\mathbf{m}_{/i}, \mathbf{u}) + \lambda_i \quad (300)$$

なる条件を得る。ここで $m_i = \int dx_i x_i P_i(x_i|\mathbf{u})$ であり、まさに求めたい局所磁化である。等式制約について注意しながら $P_i(x_i|\mathbf{u})$ を求めると、

$$P_i(x_i|\mathbf{u}) = \frac{1}{Z_i^{\text{MF}}(\mathbf{u})} \exp(-E_i(x_i|\mathbf{m}_{/i}, \mathbf{u})) \quad (301)$$

を得る。ここで $Z_i^{\text{MF}}(\mathbf{u})$ は規格化定数である。

イジング模型の平均場近似

具体的に $P_i(x_i|\mathbf{u})$ の場合について計算すると、

$$P_i(x_i|\mathbf{u}) = \frac{\exp\left(-\sum_{j \in \partial i} J_{ij} x_i m_j + h_i x_i\right)}{2 \cosh\left(\sum_{j \in \partial i} J_{ij} m_j + h_i\right)} \quad (302)$$

である。局所磁化 m_i について自己無撞着方程式を得ることができる。

$$m_i = \tanh\left(\sum_{j \in \partial i} J_{ij} m_j + h_i\right) \quad (303)$$

相関関数については、線形応答

$$\frac{\partial}{\partial h_j} \langle x_i \rangle_{\mathbf{u}} = \langle x_i x_j \rangle_{\mathbf{u}} - \langle x_i \rangle_{\mathbf{u}} \langle x_j \rangle_{\mathbf{u}} \quad (304)$$

を利用する [14].

問:熱平均の定義とイジング模型のエネルギー関数にもとづいて上記の線形応答の関係式を示せ

まず自己無撞着方程式の両辺に対して \tanh^{-1} をとる。

$$\tanh^{-1} m_j = \sum_{k \in \partial j} J_{jk} m_k + h_j \quad (305)$$

ここから

$$h_j = \tanh^{-1} m_j - \sum_{k \in \partial j} J_{jk} m_k \quad (306)$$

であるので、

$$\frac{dm_i}{dh_j} = \left(\frac{dh_j}{dm_i} \right)^{-1} = \left(\frac{\delta_{ij}}{1 - m_j^2} - J_{ij} \right)^{-1} \quad (307)$$

を得る。ここで $J_{ji} = J_{ij}$ を利用した。これらの結果を利用して、勾配法に必要な熱平均値 $\langle x_i \rangle_{\mathbf{u}}$ および $\langle x_i x_j \rangle_{\mathbf{u}}$ を近似的に得る。平均場近似のよいところはその計算の速さにある。精度を多少犠牲にしても高速に計算ができる。但し最近平均場近似の様々な矛盾を解消することで、平均場近似の精度向上を試みる研究が一気に進み、相当な精度の良さを達成することに成功している [16, 17]。また平均場近似では独立な確率分布 $P_i(x_i | \mathbf{u}, \mathbf{x}_{/i})$ による変分を用いたが、さらに2変数による確率分布からの変分による信念伝搬法 (Belief Propagation)、その一般化によるクラスター変分法など多数の進展があり、いわゆる統計力学におけるベータ近似との関係もあり豊富な研究成果がある。

疑似最尤法 (Pseudo Likelihood Estimation) 平均場近似と似た精神を持っている簡単な仕組みでありながら、データ数 D が非常に大きい場合に漸近的に最尤法と一致する解を得る手法である [18]。平均場近似と同様に、エネルギー関数が変数毎の和に分解できる ($E(\mathbf{x} | \mathbf{u}) = \sum_{i=1}^N E_i(x_i | \mathbf{x}_{/i}, \mathbf{u})$) 場合に、「得られたデータ」を利用した近似を分配関数についておこなう。

$$Z(\mathbf{u}) \approx \prod_{i=1}^N \sum_{x_i} \exp \left\{ -E_i(x_i | \mathbf{x}_{/i} = \mathbf{x}_{/i}^{(d)}, \mathbf{u}) \right\}. \quad (308)$$

イジング模型の場合には、

$$Z(\mathbf{u}) \approx 2^N \prod_{i=1}^N \cosh \left(\sum_{j \in \partial i} J_{ij} x_j^{(d)} + h_i \right). \quad (309)$$

とする。この分配関数を利用した尤度関数を疑似尤度関数と呼ぶ。この結果にもとづき、

$$\langle x_i \rangle_{\mathbf{u}} \approx \tanh \left(\sum_{j \in \partial i} J_{ij} x_j^{(d)} + h_i \right) \quad (310)$$

$$\langle x_i x_j \rangle_{\mathbf{u}} \approx x_i^{(d)} \tanh \left(\sum_{j \in \partial i} J_{ij} x_j^{(d)} + h_i \right) + x_j^{(d)} \tanh \left(\sum_{i \in \partial j} J_{ji} x_i^{(d)} + h_j \right) \quad (311)$$

ここで $J_{ij} = J_{ji}$ であることを用いているため、相関についてはふたつの項が現れる。どうも先ほどの平均場近似と非常に似ているが、疑似最尤法は m_i ではなく $x_i^{(d)}$ 、つまり「得られたデータ」を利用しているところが大きく異なる。

マルコフ連鎖モンテカルロ法 統計力学を知る読者は、熱平均の計算にマルコフ連鎖モンテカルロ法 (MCMC 法) を採用したらよいのではないかと考えたかもしれない。マルコフ連鎖モンテカルロ法は、所望の確率分布に収束するマスター方程式を計算機上でシミュレートすることにより、期待値を計算する手法である [19]。イジング模型のマルコフ連鎖モンテカルロ法について簡単に触れておこう。イジング模型の確率分布の時間発展を考えるために離散時間マスター方程式を考える。

$$P_{[t+1]}(\mathbf{y}) = \sum_{\mathbf{x}} W_{\mathbf{u}}(\mathbf{y} | \mathbf{x}) P_{[t]}(\mathbf{x}) \quad (312)$$

ここで $W_{\mathbf{u}}(\mathbf{y} | \mathbf{x})$ が状態 \mathbf{x} から \mathbf{y} へと遷移する確率であり、 $P_{[t]}(\mathbf{x})$ が現時刻で得られている確率分布である。この確率過程の収束先の定常分布がボルツマン分布であることを要求する。そのためには釣り合い条件が成立すれば良い。

釣り合い条件

所望の定常分布 $P_{ss}(\mathbf{x})$ を持つための遷移確率の条件は、

$$\sum_{\mathbf{x}} W_{\mathbf{u}}(\mathbf{y}|\mathbf{x})P_{ss}(\mathbf{x}|\mathbf{u}) = P_{ss}(\mathbf{y}|\mathbf{u}) \quad (313)$$

を満たせばよい。

問題はこの釣り合い条件を満たす遷移確率を、できるだけ簡単に用意したいということだ。そのための処方箋として、詳細釣り合い条件を更に課することが多い。

詳細釣り合い条件

以下の詳細釣り合い条件を満たす遷移確率は

$$\frac{W_{\mathbf{u}}(\mathbf{y}|\mathbf{x})}{W_{\mathbf{u}}(\mathbf{x}|\mathbf{y})} = \frac{P_{ss}(\mathbf{y}|\mathbf{u})}{P_{ss}(\mathbf{x}|\mathbf{u})} \quad (314)$$

釣り合い条件を満たす。

ここで注意したいのは、釣り合い条件を満たすものを見つけるために更に制限の厳しい詳細釣り合い条件を課したことである。所望の定常分布に行き着くためには釣り合い条件を満たせば良い。(細かいことを言えば任意の2状態へ有限のステップで遷移する確率が存在するというエルゴード条件も必要。) その十分条件として詳細釣り合いがある。この詳細釣り合い条件を満たすとき、平衡系と呼び定常状態は特別な名前が冠される、平衡状態である。最近ではその詳細釣り合い条件を破ったマルコフ連鎖モンテカルロ法がいくつか提案されており、その高速な緩和が注目されているが、これは別の機会に譲ろう。いくつかの文献を挙げるに留める [20, 21, 22].

さてイジング模型のマルコフ連鎖モンテカルロ法を実行するために詳細釣り合い条件を満たす解を探してみよう。代表的なものはある特定の1スピンをフリップするルールを採用して、以下の遷移行列によるふたつが挙げられる。ランダムに選んだある特定のスピンの x_i を x_i から $-x_i$ へと変化させる。この状態を $\bar{\mathbf{x}}$ とする。そのとき遷移確率を

$$W_{\mathbf{u}}(\bar{\mathbf{x}}|\mathbf{x}) = \min \left(1, \frac{P_{ss}(\bar{\mathbf{x}}|\mathbf{u})}{P_{ss}(\mathbf{x}|\mathbf{u})} \right) = \min \left(1, \exp \left(-2 \sum_{j \in \partial i} J_{ij} x_i x_j - 2h_i x_i \right) \right) \quad (315)$$

とする。これをメトロポリス法と呼ぶ。一方遷移確率を

$$W_{\mathbf{u}}(\bar{\mathbf{x}}|\mathbf{x}) = \frac{P_{ss}(\bar{\mathbf{x}})}{P_{ss}(\mathbf{x}|\mathbf{u}) + P_{ss}(\bar{\mathbf{x}}|\mathbf{u})} = \frac{1}{2} \left\{ 1 - \tanh \left(\sum_{j \in \partial i} J_{ij} x_i x_j + h_i x_i \right) \right\} \quad (316)$$

とする熱浴法 (グラウバーダイナミクス) がある。

[問：それぞれ詳細釣り合いを満たすことを確認せよ。]

メトロポリス法が実装が簡単なため、アルゴリズムを紹介しておく。

メトロポリス法によるマルコフ連鎖モンテカルロ法

1. $\mathbf{x}[0]$ を初期化する。例えば一様分布からサンプリングする。
2. スピンを反転するサイト i を選ぶ。
3. 反転に必要なエネルギー $\Delta E(\mathbf{x}[s]|\mathbf{u})$ を計算する。

$$\Delta E(\mathbf{x}[s]|\mathbf{u}) = -2 \sum_{j \in \partial i} J_{ij} x_i[s] x_j[s] - 2h_i x_i[s] \quad (317)$$

4. エネルギーが下がるときはスピン反転を受け入れる。そうでない場合は一様乱数 r が以下の不等式を満たすときに反転を受け入れる。

$$r \leq \exp\left(-\frac{2\Delta E(\mathbf{x}[s]|\mathbf{u})}{T}\right) \quad (318)$$

5. 終了基準を迎えるまで、ステップ 2-4 を繰り返す。

十分な時間 t_w の緩和を待った後、平衡状態に到達していると考えられるため、得られた確率分布関数からサンプリングを行うことで、以下のように期待値を計算する。

$$\langle f(\mathbf{x}) \rangle_{\mathbf{u}} \approx \frac{1}{T} \sum_{s=t_w+1}^{T+t_w} f(\mathbf{x}[s]) \quad (319)$$

t_w が十分におおきく、 T も多く取ることにより期待値の計算は非常に高精度で行うことができる。そのため精度を要求すると、マルコフ連鎖モンテカルロ法は、平衡分布への緩和に時間がかかるため計算量が膨大なものとなる。

最小確率流法 (Minimum Probability Flow) 尤度関数の代わりに新しいコスト関数を導入して、パラメータ推定を行う最近の手法を紹介しよう。最小確率流法 (Minimum Probability Flow、以下 MPF とする) は疑似最尤法よりも収束性に優れた手法として知られる [24]。この方法は、連続時間マスター方程式に基づく確率分布の時間発展に注目した方法である。そのため統計力学の観点からの研究が待たれる非常に興味深い手法である。まずマスター方程式を以下のように書き下す。

$$\frac{dP_t(\mathbf{y})}{dt} = \sum_{\mathbf{x}} W_{\mathbf{u}}(\mathbf{y}|\mathbf{x})P_t(\mathbf{x}). \quad (320)$$

ここで最小確率流法では、初期条件は与えられたデータによる経験分布とする。

$$P_0(\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D \delta(\mathbf{x} - \mathbf{x}^{(d)}) \quad (321)$$

そこから指定したパラメータ \mathbf{u} により決まるボルツマン分布への緩和を考えてみよう。このとき仮のパラメータが与えられたデータに適合するものであれば、初期分布は定常分布に非常に近いものと考えられる。そのためマスター方程式による分布の変化は非常に小さいものとなるだろう。この考察に基づいて初期分布から微小時間経過したときの KL 情報量を最小化する方法が最小確率流法である。

まずはマスター方程式の遷移行列をボルツマン機械学習で導入した確率分布を定常分布としてもつように設計する。つまり釣り合い条件を満たすようにする。さらに扱いが容易になるように詳細釣り合いを課した 1 スピンフリップによる遷移行列として、以下の遷移行列を採用する。

$$W_{\mathbf{u}}(\bar{\mathbf{x}}|\mathbf{x}) = \exp\left\{-\frac{1}{2}(E(\bar{\mathbf{x}}|\mathbf{u}) - E_j(\mathbf{x}|\mathbf{u}))\right\}. \quad (322)$$

$\bar{\mathbf{x}}$ は先ほどと同様に \mathbf{x} から 1 スピンだけフリップした状態を示す。最小確率流法の処方箋に従い、以下のように初期分布と微小時間後の分布との KL 情報量を考えてみる。

$$D_{\text{KL}}(P_0|P_t) \approx D_{\text{KL}}(P_0(\mathbf{x})|P_0(\mathbf{x})) + dt \frac{d}{dt} D_{\text{KL}}(P_0(\mathbf{x})|P_t(\mathbf{x}))|_{t=0}. \quad (323)$$

マスター方程式を思い出して計算をすることで以下の量を得る。

$$D_{\text{KL}}(P_0(\mathbf{x})|P_t(\mathbf{x})) \approx \frac{dt}{D} \sum_{d=1}^D \sum_{\mathbf{y} \in \partial \mathbf{x}^{(d)}} \exp\left\{-\frac{1}{2}(E(\mathbf{y}|\mathbf{u}) - E(\mathbf{x}^{(d)}|\mathbf{u}))\right\}. \quad (324)$$

[問:確認せよ.]

ここで和の添字 $\mathbf{y} \in \partial \mathbf{x}^{(d)}$ は $\mathbf{x}^{(d)}$ から 1 スピンフリップで移れる状態のみで和を取ることを意味する。先ほどの議論に基づいて、この量が小さいときほど、初期分布は定常分布に近いものと考えられる。定常分布はパラメータによって決まるため、この量が最小となるようなパラメータを計算することで推定値を得る。これが最小確率流法である。注意したいのはこの方法は実際にマルコフ連鎖モンテカルロ法を用いてマスター方程式による分布の変化を追う必要はない。あくまで式 (324) を与えられたデータに基づき計算して最小化をすればよい。そのため最尤法のように勾配法に基づく解法を行うことができる。対数尤度関数の代わりに以下のコスト関数を定義する。

$$L_{\text{MPF}}(\mathbf{u}) \equiv -\frac{1}{D} \sum_{d=1}^D \sum_{\mathbf{y} \in \partial \mathbf{x}^{(d)}} \exp \left\{ -\frac{1}{2} \left(E(\mathbf{y}|\mathbf{u}) - E(\mathbf{x}^{(d)}|\mathbf{u}) \right) \right\}. \quad (325)$$

このコスト関数をパラメータ \mathbf{u} について微分をとると、

$$\frac{\partial L_{\text{MPF}}(\mathbf{u})}{\partial \mathbf{u}} = \frac{1}{2D} \sum_{d=1}^D \sum_{\mathbf{y} \in \partial \mathbf{x}^{(d)}} \left(\frac{\partial E(\mathbf{y}|\mathbf{u})}{\partial \mathbf{u}} - \frac{\partial E(\mathbf{x}^{(d)}|\mathbf{u})}{\partial \mathbf{u}} \right) \exp \left\{ -\frac{1}{2} \left(E(\mathbf{y}|\mathbf{u}) - E(\mathbf{x}^{(d)}|\mathbf{u}) \right) \right\} \quad (326)$$

を得る。それでは実際にイジング模型の場合について、コスト関数 (325) の最小化してみよう。1 スピンフリップであるという条件を考えると、どこかひとつの i において、 $y_i = -x_i^{(d)}$ となるので、

$$E(\mathbf{y}|\mathbf{u}) - E(\mathbf{x}^{(d)}|\mathbf{u}) = 2x_i^{(d)} \left(h_i + \sum_{j \in \partial i} J_{ij} x_j^{(d)} \right) \quad (327)$$

であることが分かる。一旦固定した i を後で動かして和を取れば同等のものが得られる。

$$L_{\text{MPF}}(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \sum_{i=1}^N \exp \left\{ - \left(h_i + \sum_{j \in \partial i} J_{ij} x_j^{(d)} \right) x_i^{(d)} \right\}. \quad (328)$$

このことから各勾配は、

$$\frac{\partial L_{\text{MPF}}(\mathbf{u})}{\partial J_{ij}} = \frac{1}{D} \sum_{d=1}^D x_i^{(d)} x_j^{(d)} \exp \left\{ - \left(h_i + \sum_{j \in \partial i} J_{ij} x_j^{(d)} \right) x_i^{(d)} \right\} \quad (329)$$

$$\frac{\partial L_{\text{MPF}}(\mathbf{u})}{\partial h_i} = \frac{1}{D} \sum_{d=1}^D x_i^{(d)} \exp \left\{ - \left(h_i + \sum_{j \in \partial i} J_{ij} x_j^{(d)} \right) x_i^{(d)} \right\} \quad (330)$$

となることがわかる。この勾配にもとづき、パラメータを更新することで効率のよい学習が達成できる。

6.4 更なる複雑度への挑戦

ボルツマン機械学習は、提案当初の黎明期はその計算量の要求に応えられず、歓迎されたものではなかった。平均場近似を始めとする高速な計算手法では精度を犠牲にしてしまう。擬似最尤法ではデータの量が重要である。マルコフ連鎖モンテカルロ法では計算時間が膨大なものとなる。しかしながら、近年の計算機の性能向上とデータの取得量があいまって、ボルツマン機械学習を実行することはそう難しくなくなってきた。そうなれば、更なる複雑度への挑戦である。エネルギー関数 $E(\mathbf{x}|\mathbf{u})$ を複雑にすればするほど生成モデルの表現の幅が大きく向上する。例えば 3 体相互作用、4 体相互作用といったように、エネルギー関数の中身に $J_{ijklmn} \dots x_i x_j x_k x_l x_m x_n \dots$ といった複数の積からなる項を用意することが考えられる。一方で、マルコフ連鎖モンテカルロ法によるカノニカル分布への収束が遅くなり、平均場近似による精度の悪化が顕著になる。また擬似最尤法等データの量による精度向上も期待できなくなる。やはり難しくすればする程、

大変なものになってしまうのだ。その計算量と複雑度のギリギリを攻めることを考える。その結実のひとつが深層学習の基礎となる隠れ変数有りボルツマン機械学習である。

その隠れ変数有りボルツマン機械学習について、単純で有名な例を通して理解していこう。隠れ変数有りボルツマン機械学習では、データを表現する可視変数 \mathbf{v} と、そのデータと結合して、しかしデータの表現に直接は関わらない隠れ変数 \mathbf{h} を導入する。これらの間にあるエネルギー関数 $E(\mathbf{v}, \mathbf{h})$ を仮定して、ボルツマン機械学習を行うことを考える。典型的な例は、混合ガウスモデルである。可視変数 \mathbf{v} は平均 $\boldsymbol{\mu}_h$ 、分散 σ_h^2 のガウス分布に従うとして、隠れ変数 h は $1 \sim K$ 個の添字を表すとして、それぞれ重み w_h をもつ場合、以下の確率分布に従う。

$$P(\mathbf{v}, \mathbf{h} | \mathbf{u}) = P(\mathbf{v} | \mathbf{h}, \mathbf{u}) P(\mathbf{h} | \mathbf{u}) = p_h \sqrt{\frac{1}{2\pi\sigma_h^2}} \exp\left\{-\frac{1}{2\sigma_h^2} (\mathbf{v} - \boldsymbol{\mu}_h)^2\right\} \quad (331)$$

ここで $p_h = w_h / \sum_{h=1}^K w_h$ である。またパラメータ \mathbf{u} は、 $(\boldsymbol{\mu}_h, \sigma_h, p_h)$ のセットを表す。エネルギー関数による表記をすれば、自明な定数を除き、

$$E(\mathbf{v}, \mathbf{h} | \mathbf{u}) = \frac{1}{2\sigma_h^2} (\mathbf{v} - \boldsymbol{\mu}_h)^2 + \frac{1}{2} \log \sigma_h^2 - \log p_h \quad (332)$$

となる。また事実上観測される可視変数のみの周辺確率を計算すると、

$$P(\mathbf{v} | \mathbf{u}) = \sum_{h=1}^K p_h \sqrt{\frac{1}{2\pi\sigma_h^2}} \exp\left\{-\frac{1}{2\sigma_h^2} (\mathbf{v} - \boldsymbol{\mu}_h)^2\right\} \quad (333)$$

となり、混合ガウス分布と呼ばれるものである。典型的な隠れ変数を有する問題設定である。ここでは隠れ変数は、どのガウス分布からデータが生成されたかを示している。いわば2段階の構造を取っていることがわかる。隠れ変数により選ばれたガウス分布に従い、データを表す \mathbf{v} が出力される。隠れ変数は、我々の観測結果からは見ることで見ることができないものである。このように隠れ変数が存在するときに、観測できるデータ \mathbf{v} は不完全であるという。仮に隠れ変数についての情報も知り得る場合、そのデータ \mathbf{v}, \mathbf{h} は完全であるという。

この隠れ変数の導入により、単純なガウス分布から多少の変更で複雑なデータの生成過程を表現していることに注目してもらいたい。少しだけの変更で、ガウス分布の構造は著しく損なうことがなく生成モデルの複雑度は増している。ガウス分布ならではの計算のしやすさや、数値計算上での実装のしやすさ等はある程度保持したまま、より多彩なデータの構造を表現することができる。しかしながら、この隠れ変数がある問題では、尤度が極大の解が多く存在して、学習の効率を悪化させることがよくあることが知られている。こればかりは多彩なデータを表現するための犠牲でもあり、ある程度は仕方がないことである。

期待値最大化法 この混合ガウス分布に対する最尤法の解法として、有名なものが期待値最大化法 (Expectation-Maximization) である。仮に完全なデータの経験分布が得られたとしよう。その経験分布に一番適合する $P(\mathbf{v}, \mathbf{h} | \mathbf{u})$ を得ようという場合、KL 情報量最小化から尤もらしいパラメータが得られる。しかしながら我々が得ることのできるものは、可視変数だけである。その経験分布は可視変数の周辺分布を近似してくれるだろう。そこで以下の対数尤度関数の最大化を考える。

$$L(\mathbf{u}) = \sum_{d=1}^D \log P(\mathbf{v}^{(d)} | \mathbf{u}) \quad (334)$$

式 (333) にあるように可視変数の周辺分布には和が含まれており、そのままでは計算がしやすい格好にはない。そこで、条件付き確率を用いて以下のように周辺分布を書き換える。

$$L(\mathbf{u}) = \sum_{d=1}^D \left\{ \log P(\mathbf{h}, \mathbf{v}^{(d)} | \mathbf{u}) - \log P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{u}) \right\} \quad (335)$$

書き換えるのは勝手だが、このままでは \mathbf{h} については、確率変数のままであるので都合が悪い。何らかの確率分布で平均をとることにしよう。

$$L(\mathbf{u}) = \sum_{d=1}^D \left\{ \int d\mathbf{h} Q_d(\mathbf{h}) \log P(\mathbf{h}, \mathbf{v}^{(d)} | \mathbf{u}) - \int d\mathbf{h} Q_d(\mathbf{h}) \log P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{u}) \right\} \quad (336)$$

左辺はそもそも隠れ変数には依存しないので、これは任意の確率分布 $Q_d(\mathbf{h})$ について成立する。KL 情報量の形で書き直すことを狙うと以下の式を得る。

$$L(\mathbf{u}) = \sum_{d=1}^D \left\{ \int d\mathbf{h} Q_d(\mathbf{h}) \log \left(\frac{P(\mathbf{v}^{(d)}, \mathbf{h} | \mathbf{u})}{Q_d(\mathbf{h})} \right) - \int d\mathbf{h} Q_d(\mathbf{h}) \log \left(\frac{P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{u})}{Q_d(\mathbf{h})} \right) \right\} \quad (337)$$

KL 情報量の非負性より、右辺第二項を落とすことで、対数尤度関数の下限を得ることができる。

$$L(\mathbf{u}) \geq \sum_{d=1}^D \left\{ \int d\mathbf{h} Q_d(\mathbf{h}) \log \left(\frac{P(\mathbf{v}^{(d)}, \mathbf{h} | \mathbf{u})}{Q_d(\mathbf{h})} \right) \right\} \quad (338)$$

等号成立条件は $Q_d(\mathbf{h}) = P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{u})$ が成立するときである。ここで、 $\log P(\mathbf{h}, \mathbf{v} | \mathbf{u})$ は和が含まれず、容易に計算が実行できることに注目したい。等号成立条件が成立する場合、事後確率 $P(\mathbf{h} | \mathbf{v}, \mathbf{u})$ に関する事後平均を計算すれば良いことがわかる。一方で、事後確率の対数 $\log P(\mathbf{h} | \mathbf{v}, \mathbf{u})$ については計算が複雑なままである。そこで一歩後退して、任意の $Q_d(\mathbf{h})$ で下限となる事実を利用して、メジャライザーの逆、マイナライザーとなる次の関数を定義する。

$$Q(\mathbf{u}, \mathbf{w}) = \sum_{d=1}^D \left\{ \int d\mathbf{h} P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{w}) \log \left(\frac{P(\mathbf{v}^{(d)}, \mathbf{h} | \mathbf{u})}{P(\mathbf{h} | \mathbf{v}^{(d)}, \mathbf{w})} \right) \right\} \quad (339)$$

この関数は $\mathbf{u} = \mathbf{w}$ のとき $L(\mathbf{w})$ となり、かつ式 (338) より $L(\mathbf{u}) \geq Q(\mathbf{u}, \mathbf{w})$ である。さらに自明に成立する不等式 $Q(\mathbf{u}, \mathbf{w}) \geq L(\mathbf{w})$ が成立する。この性質は、圧縮センシングの ISTA を構成する際に利用したメジャライザーと同様である。最尤法では、最大化を目指すため、下限を持つてくるため不等号は全て逆をとっている。このようにメジャライザーと同じ構造をもつことから、逐次更新のアルゴリズムを組み立てることができる。特に $Q(\mathbf{u}, \mathbf{w})$ の関数の最大化を行う際に、 \mathbf{u} に依存している部分が、和を含まない結合確率 $P(\mathbf{v}, \mathbf{h} | \mathbf{u})$ にしか含まれないところが重要である。

期待値最大化法 (Expectation-Maximization algorithm)

1. $t = 0$ とする。適当な初期条件で $\mathbf{u}[0]$ を初期化する。
2. マイナライザーである $Q(\mathbf{u}, \mathbf{w})$ を計算する。 (E-step)
3. 以下の最大化問題をとく。 (M-step)

$$\mathbf{u}[t+1] = \arg \max_{\mathbf{u}} \{Q(\mathbf{u}, \mathbf{u}[t])\}. \quad (340)$$

4. 終了基準を迎えるまで、ステップ 2-4 を繰り返す。

制限ありボルツマン機械 隠れ変数の導入により、複雑度をあげながら、しかし計算はしやすい構造をとる生成モデルを用意することができた。この性質を巧みに利用したのがいわゆる深層学習である。可視変数部分からデータが表現される一方で、多彩なデータの構造は、背後にある複雑なもので表現されるものとして、隠れ変数にゆだねるという発想である。そこで最も単純なものとして、可視変数同士、隠れ変数同士には結合を考えず、可視変数と隠れ変数だけに結合がある制限有りボルツマン機械を導入しよう。

$$P(\mathbf{v}, \mathbf{h} | \mathbf{u}) = \frac{1}{Z(\mathbf{u})} \exp \left(\sum_{i=1}^{N_v} b_i v_i + \sum_{j=1}^{N_h} c_j h_j + \sum_{i,j} W_{ij} v_i h_j \right). \quad (341)$$

ここで、 $Z(\mathbf{u})$ は規格化定数である。またパラメータは $\mathbf{u} = (W, \mathbf{b}, \mathbf{c})$ である。隠れ変数について和を取り、可視変数のみの周辺分布を見てみよう。

$$P(\mathbf{v}|\mathbf{u}) = \frac{1}{Z(\mathbf{u})} \exp \left\{ \sum_{i=1}^{N_v} b_i v_i + \sum_{j=1}^{N_h} \log 2 \cosh \left(c_j + \sum_{i=1}^{N_v} W_{ij} v_i \right) \right\}. \quad (342)$$

可視変数について実効的に相互作用が生じていることが見て取れる。つまり本来可視変数同士は生成モデルとしては結合がないものの、隠れ変数を通して結合している効果が現れている。このように可視変数同士、隠れ変数同士に結合がない制限を課したボルツマン機械を、制限ありボルツマン機械と呼ぶ。この制限ありボルツマン機械は、以下の重要な性質を持つ。同様にして、隠れ変数についての周辺分布を計算すると

$$P(\mathbf{h}|\mathbf{u}) = \frac{1}{Z(\mathbf{u})} \exp \left\{ \sum_{j=1}^{N_h} c_j h_j + \sum_{i=1}^{N_v} \log 2 \cosh \left(b_i + \sum_{j=1}^{N_h} W_{ij} h_j \right) \right\}. \quad (343)$$

これを用いて、隠れ変数を知った上での条件付き確率を計算すると、

$$P(\mathbf{v}|\mathbf{h}, \mathbf{u}) = \prod_{i=1}^{N_v} \exp \left\{ b_i v_i + \sum_{j=1}^{N_h} W_{ij} v_i h_j - \log 2 \cosh \left(b_i + \sum_{j=1}^{N_h} W_{ij} h_j \right) \right\}. \quad (344)$$

という格好を持つ。特徴的なのが、可視変数について独立であるということがわかる。つまり条件付き確率によるサンプリングが容易であるということだ。先ほど紹介したようなマルコフ連鎖モンテカルロ法を利用して、サンプリングに苦心するのは、変数同士に相関があり、確率分布に従ったサンプリングを即座に行うことができないためである。しかし相関がない場合は、マルコフ連鎖モンテカルロ法を利用する必要がない。そのためマルコフ連鎖モンテカルロ法での定常分布までの緩和を待つ必要がないということになる。このような条件付き確率における独立性を、条件付き独立と呼び、制限ありボルツマン機械の利点がここにある。緩和を待たずにサンプリングを即座に行うことができ、なおかつ生成モデルにそれなりに複雑度を持たせることができる。同様に可視変数を知った上での条件付き確率についても同様に

$$P(\mathbf{h}|\mathbf{v}, \mathbf{u}) = \prod_{j=1}^{N_h} \exp \left\{ c_j h_j + \sum_{i=1}^{N_v} W_{ij} v_i h_j - \log 2 \cosh \left(c_j + \sum_{i=1}^{N_v} W_{ij} v_i \right) \right\}. \quad (345)$$

条件付き独立性を有した格好をしている。

この生成モデルを用いて、以下の対数尤度関数の最大化を目指すことでパラメータの学習を行う。

$$L(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \log P(\mathbf{v}^{(d)}|\mathbf{u}) \quad (346)$$

ここでパラメータについての偏微分を行うと、

$$\frac{\partial}{\partial \mathbf{b}} L(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \mathbf{v}^{(d)} - \langle \mathbf{v} \rangle_{\mathbf{u}} \quad (347)$$

$$\frac{\partial}{\partial \mathbf{c}} L(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \tanh \left(\mathbf{c} + \left(\mathbf{v}^{(d)} \right)^T W \right) - \langle \mathbf{h} \rangle_{\mathbf{u}} \quad (348)$$

$$\frac{\partial}{\partial W} L(\mathbf{u}) = \frac{1}{D} \sum_{d=1}^D \mathbf{v}^{(d)} \left(\tanh \left(\mathbf{c} + \left(\mathbf{v}^{(d)} \right)^T W \right) \right)^T - \langle \mathbf{v}^{(d)} \mathbf{h}^T \rangle_{\mathbf{u}} \quad (349)$$

と各勾配を得ることができるが、再び熱期待値の計算が必要となる。

コントラストティブ・ダイバージェンス法 このような制限ありボルツマン機械の特性である条件付き独立性を利用して、提案されたものがコントラストティブ・ダイバージェンス法 (Contrastive divergence、以下 CD 法と略記する) である。隠れ変数ありの生成モデルでは、可視変数のみが与えられるので、隠れ変数についての情報をなにかの形で擬似生成する必要がある。そこに先ほどの条件付き独立性を利用する。CD 法では、可視変数と隠れ変数の“条件付き”サンプリングを交互に行うことにより、生成モデルに従った確率分布のサンプリングを行う。「得られたデータ」の経験分布を可視変数の初期条件として始める [23].

コントラストティブ・ダイバージェンス法

1. $t = 0$ とする。 $\mathbf{v}[0] = \mathbf{v}^{(d)}$ と初期化する。以下マルコフ連鎖モンテカルロ法と同様。
2. 条件付き確率 $P(\mathbf{h}^t | \mathbf{v}^t, \mathbf{u})$ にもとづき、隠れ変数 \mathbf{h}^t をサンプリングする。
3. 条件付き確率 $P(\mathbf{v}^{t+1} | \mathbf{h}^t, \mathbf{u})$ にもとづき、隠れ変数 \mathbf{v}^{t+1} をサンプリングする。
4. k ステップのサンプリングを終えたら、得られた $\mathbf{v}^{(d)}[k+1], \mathbf{h}^{(d)}[k]$ ($d = 1, 2, \dots, D$) により経験平均を取り、これを熱期待値とする。

$$\langle f(\mathbf{v}, \mathbf{h}) \rangle_{\mathbf{u}} \approx \frac{1}{D} \sum_{d=1}^D f(\mathbf{v}^{(d)}[k+1], \mathbf{h}^{(d)}[k]) \quad (350)$$

k ステップだけ行う場合の CD 法を CD- k と呼ぶ。経験的には、この k はごくわずかであっても良いことが多いとされている。もちろんこの k が大きい方が最尤法の計算が正しく行われるのだが、小さい場合についても尤度の高いパラメータを得ることができるため、高速に複雑なモデルの学習を行うことができるため重宝される。但しデータの量がものを言うことを注意しておきたい。CD 法では最後に経験平均を取るのだが、この経験平均を取るのにはデータの個数分であり D 個の和からなる。そのため大きなデータ数でないと正しい熱期待値を統計誤差の意味で計算することができないので注意したいところである。

6.5 確率勾配法

非常に大きなデータを取得することができるようになった現代においては、データが少ないから学習を行うことができないという状況は回避できる場面もあるだろう。むしろ逆にデータの数が大きすぎて、手に負えないという状況はありえる。

6.6 スパースな相関推定

それではボルツマン機械学習を使って何ができるか。多くの利用例があり得るうち、スパースな相関推定について紹介しよう。多くの要素からなるデータが得られたときに、どの要素とどの要素が強く関係しているのだろうか？それを調べたいときにボルツマン機械学習が有効となる。そのときに相互作用係数 J_{ij} はありとあらゆる組み合わせの間において考慮する必要がある。しかし一方で、本当に重要な関係性だけを抜き出したいという場合、 L_1 ノルムによる変数選択の効果を利用したボルツマン機械学習が有効である [25].

そこで尤度関数の最大化を行うときに、 L_1 ノルム項を追加する。

$$\min_{\mathbf{u}} \left\{ -L(\mathbf{u}) + \lambda \sum_{i < j} \|J_{ij}\|_1 \right\} \quad (351)$$

ここで符号を変えて最小化をすることにしよう。また $J_{ij} = J_{ji}$ という対称性を仮定する。尤度関数の代わ

りに疑似尤度関数を用いてもよいし、最小確率流法を用いて、

$$\min_{\mathbf{u}} \left\{ L_{\text{MPF}}(\mathbf{u}) + \lambda \sum_{i < j} \|J_{ij}\|_1 \right\} \quad (352)$$

としてもよい。λ > 0 は大きくすればするほど、得られる推定解のスパース性を促す。そのため非常に数少ない本質的なものだけを抽出したいというのであればλは大きくすることが望まれる。

ここで圧縮センシングの経験が活かされる。L₁ ノルムを含む最適化問題を扱わなければならない。尤度関数はやや複雑な格好をしているが、メジャライザー最小化による方法で割と簡単に最適化することができる。ここで $g(\mathbf{u}) = -L(\mathbf{u})$ または $g(\mathbf{u}) = L_{\text{MPF}}(\mathbf{u})$ であり、 $f(\mathbf{u}) = \|\mathbf{u}\|_1$ と考えればよい。メジャライザー最小化により、

$$\mathbf{u}[t+1] = \arg \min_{\mathbf{u}} \left\{ f(\mathbf{u}) + (\nabla g(\mathbf{u}))^T (\mathbf{u} - \mathbf{u}[t]) + \frac{L}{2} \|\mathbf{u} - \mathbf{u}[t]\|_2^2 \right\} \quad (353)$$

つまり必要なのは対数尤度関数の勾配であり、これまでに散々評価してきたものであり、またリプシッツ定数が必要である。圧縮センシングにおける LASSO 型の最小化問題のときには、リプシッツ定数を予め評価することができたが、ボルツマン機械学習の場合には予めの評価が可能であることは期待できそうにない。そのようにリプシッツ定数が非自明な場合には、以下のようにメジャライザーの性質 $g(\mathbf{u}) \leq q_L(\mathbf{u}, \mathbf{u}[t-1])$ に注目して適応的に変更することでメジャライザーを構成することができる。

リプシッツ定数が非自明な場合の対処法

1. 初期化する。適当な大きさのリプシッツ定数 $L[0]$ を用意する。係数 $\alpha > 1$ を用意する。
2. 仮のリプシッツ定数 $L[t]$ で暫定的にメジャライザー最小解 $\mathbf{u}[t]$ を求める。

$$\mathbf{u}[t] = \arg \min_{\mathbf{u}} \{ f(\mathbf{u}) + q_L(\mathbf{u}, \mathbf{u}[t-1]) \} \quad (354)$$

3. 利用したリプシッツ定数によるメジャライザーが $g(\mathbf{u}[t])$ を上回ることを確認する。

$$g(\mathbf{u}[t]) \leq q_L(\mathbf{u}[t], \mathbf{u}[t-1]) \quad (355)$$

もし成立していなかったら、非負の整数値 i を増やして $L[t] = \alpha^i L[t-1]$ として更新してステップ 2 に戻る。

この方法を利用すれば LASSO 型の最適化問題以外にも FISTA は柔軟に利用することができる。試しに $N = 25$ 個のノード間の $N(N-1)/2 = 300$ 個の相互作用のうち 9 割程度が零成分をもち、残りは平均 0、分散 1 のガウス分布に従う非零成分を持つ J_{ij} 、同様に平均 0、分散 1 ガウス分布に従う磁場 (バイアス) h_i を持つイジング模型のボルツマン分布から人工データとして、5000 個のデータを用意して、疑似最尤法及び最小確率流法を L_1 ノルム項つきで実行することで得た結果を載せる (図 19)。どちらの方法を採用するにせよ、うまくスパースな相関を推定することに成功している。最小確率流法のほうが、収束までにかかる反復計算の回数が少なかったことをコメントしておく。

参考文献

- [1] D. L. Donoho: Discrete & Comput. Geom., **35**, (2006) 617.
- [2] D. L. Donoho and J. Tanner: Proc. Nat. Acad. Sci., **102**, (2005) 9452.

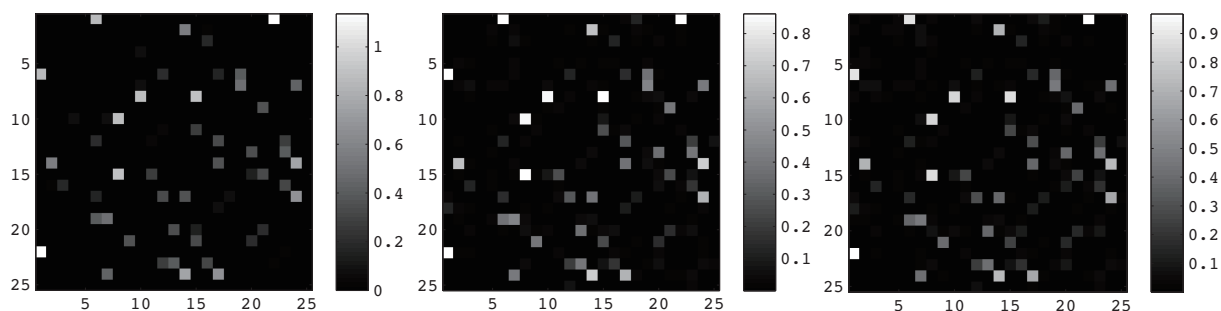


図 19: スパース相関推定. 人工データを生成した相互作用係数 (左). 疑似最尤法と L_1 ノルムによる推定結果 (中). 最小確率流法と L_1 ノルムによる推定結果 (右).

- [3] Y. Kabashima, T. Wadayama and T. Tanaka: J. Stat. Mech.: Theory Exp., (2006) L09003.
- [4] D. L. Donoho, A. Maleki and A. Montanari: Proc. Nat. Acad. Sci., **106**, (2009) 18914.
- [5] M. Lustig, D. L. Donoho and J. M. Pauly: Magn. Reson. Med., **58**, (2007) 1182.
- [6] R. Tibshirani: J. R. Statist. Soc. B, **58**, (1996) 267.
- [7] M. Honma, K. Akiyama, M. Uemura and S. Ikeda: Publ. Astron. Soc. Jpn. **66**, (2014) 95.
- [8] A. Beck and M. Teboulle: SIAM J. Imaging Sci. **2**, (2009) 183.
- [9] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly and R. G. Baraniuk: IEEE Signal Process. Mag. **25**, (2008) 83.
- [10] W. Yin, S. Osher, D. Goldfarb and J. Darbon: SIAM J. Imag. Science **1**, (2008) 143.
- [11] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein: Foundations and Trends in Machine Learning, **3** (2010) 1.
- [12] A. Beck and M. Teboulle: IEEE Trans. Image Processing **18**, (2009) 2419.
- [13] J. Huang, S. Zhang, and D. Metaxas: Lecture Note in Computer Science **6361** (2010) 135.
- [14] H. J. Kappen and F. B. Rodríguez: J. Neural Comp. **10** (1998) 1137.
- [15] J. S. Yedidia, W. T. Freeman and Y. Weiss: Mitsubishi Electric Research Laboratories, Tech. Rep. **TR2002-35**, (2002).
- [16] Mu. Yasuda and K. Tanaka: Phys. Rev. E, **87**, (2013) 012134.
- [17] J. Raymond and F. Ricci-Tersenghi: Phys. Rev. E, **87**, (2013) 052111.
- [18] J. Besag: Journal of the Royal Statistical Society. Series D (The Statistician), **24** (1975) 179.
- [19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller: J. Chem. Phys. **21**, (1953) 1087.
- [20] H. Suwa and S. Todo: Phys. Rev. Lett. **105**, (2010) 120603.
- [21] K. S. Turitsyn, M. Chertkov, and M. Vucelja: Phys. D (Amsterdam, Neth.) **240**, (2011) 410.

- [22] A. Ichiki and M. Ohzeki: Phys. Rev. E **88**, (2013) 020101.
- [23] M. Welling, and G. Hinton: Artificial Neural Networks, **2414**, (2002) 351.
- [24] J. Sohl-Dickstein, and P. B. Battaglino and M. R. DeWeese: Phys. Rev. Lett., **107**, (2011) 220601.
- [25] M. Ohzeki: J. Phys. Soc. Jpn. **84** (2015) 054801.