疎信号に対する高速フーリエ変換

M2013SS006 溝口　竜二

指導教員：杉浦洋

1　はじめに

高速フーリエ変換 (FFT) は基本的な数値アルゴリズムであり，n 次元信号の離散フーリエ変換 (DFT) は計算量 $O(n^2)$ から $O(n \log n)$ と高速で処理され，画像圧縮，信号処理などの多くの応用分野で用いられている．信号が特殊な条件を満たすときには，計算量をさらに減らすことが出来る．今回はその条件として，信号のスパース性に着目する．多くの応用分野で，信号のフーリエ係数のほとんどは 0 またはそれに等しいものである．このような信号の性質をスパース性という．信号サイズが n，非零周波数の K のとき，通常の計算では，これら K 個の周波数成分を計算するのに，$O(Kn)$ の計算量を必要とする．これより，計算量が小さいアルゴリズムを subliner time アルゴリズムといい，疎信号 FFT(SFFT) と呼ぶ．最初の subliner time のアルゴリズムは，Hadamarld 変換に関してだが，[3] で提示された．すぐ後に，フーリエ変換に対しての subliner time アルゴリズムが発見された [4][5][6][7][8].

本研究では，MIT の Hassanieh らに提案された疎信号 FFT(SFFT) のアルゴリズムを取り上げる [1]．この SFFT では，矩形関数とガウス関数を畳み込むことで得られるフラット窓関数をフィルタを用いる．このフラット窓関数は，大きなフーリエ係数はパス領域内で区切られ，その周波数を特定する必要がある．そのサンプリングしたデータに対して低次元 FFT(高速フーリエ変換) を行うことで，疎信号の重要でない部分とそうでない部分を容易に選り分けられるようにした．重要なデータを抽出すると，元信号の積和合同置換された位置データを抽出することができる．また，積和合同置換を行うことの利点は一つ信号データを指定した回数の異なる実行結果を得られることである．そこで得られた実行結果から，実行回数が半分以上出現した位置データの箇所だけを推定することで，元信号の計算量を減らし，元信号の近似解を推定する理論である.その領域だけで，すべての周波数を間接的に推定するアルゴリズムとして実行して，再現可能性や計算量の少ない再現アルゴリズムの有用性を確認する [2].

2　変換関係と座標関係の表記

入力信号を $x \in \mathbb{C}^n$ とし，その記号と定義式を合わせて書く．$\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$ とし，$x = (x_j)_{j=0}^{n-1}$ は周期 n の無限列 $(x_j)_{j=-\infty}^{\infty}$ の部分列と見なす．$x_j = x_{\mathrm{mod}(j,\, n)} (j \in \mathbb{Z})$ である．$\mathrm{mod}(j, n) \in \mathbb{Z}_n$ は j を n で割った余りである．ベクトル x の台は $supp(x) = \{j \in \mathbb{Z}_n | x_j \neq 0\}$ と定義する．

2.1　離散フーリエ変換 (DFT)

離散的時間の信号の離散的周波数スペクトルは離散フーリエ変換によって，計算量 $O(n \log n)$ で求めることができる．入力信号 x のフーリエスペクトルは以下のように定義される [1].

$$\hat{x_i} = \frac{1}{\sqrt{n}} \sum_{j \in \mathbb{Z}_n} x_j \omega^{ij} \tag{1}$$

離散フーリエ逆変換もまた，以下のように定義される．

$$x_i = \frac{1}{\sqrt{n}} \sum_{j \in \mathbb{Z}_n} \hat{x_j} \omega^{-ij} \tag{2}$$

$\omega = e^{2\pi i/n}$ は 1 の原始 n 乗根で，$i = \sqrt{-1}$ である．

3　積和合同置換

n を法として，奇数 σ，整数 $\tau \in \mathbb{Z}_n$ が与えられたとき，n 次元ベクトル x に対する，積和合同置換 $P_{\sigma,\tau}$ は次のように定義される．

$$(P_{\sigma,\tau} x)_i = x_{\sigma i + \tau}. \tag{3}$$

置換のフーリエスペクトルは以下のように表せる．

$$(\widehat{P_{\sigma,\tau} x})_{\sigma i} = \hat{x}_i \omega^{-\tau i}. \tag{4}$$

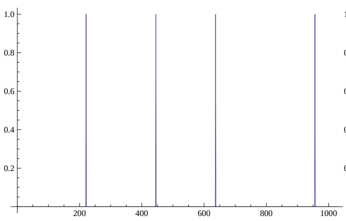

図 1　変換前 (周波数領域)

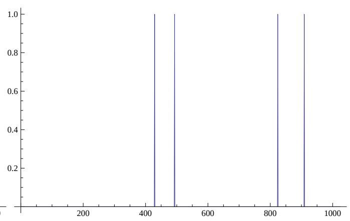

図 2　変換後 (周波数領域)

4　標準窓関数

本研究では，標準窓関数から派生するフラット窓関数を用いる [1]．デジタル信号処理では，通過域 ϵ，リップルの振幅の幅 δ，窓の幅 w をパラメータとして持つ (ϵ, δ, w) 標準窓関数 $F \in \mathbb{Z}^n$ を以下のように定義する．
台 $supp(F) \subseteq [-\frac{\omega}{2}, \frac{\omega}{2}]$ をもち，対称 $(F_{-j} = F_j)$ で，

- $\hat{F}_0 = 1$
- $\hat{F}_i > 0 \qquad i \in [-\epsilon n, \epsilon n]$
- $|\hat{F}_i| < \delta \qquad i \notin [-\epsilon n, \epsilon n]$

5 フラット窓関数

阻止域 ϵ，通過域 ϵ'，リップルの振幅の幅 δ，窓の幅 w をパラメータとして持つ $(\epsilon,\epsilon',\delta,w)$ フラット窓関数 $\mathrm{G} \in \mathbb{Z}^n$ を以下のように定義する．台 $supp(G) \subseteq [-\frac{\omega}{2},\frac{\omega}{2}]$ をもち，対称 $(G_{-j}=G_j)$ で，

- $\hat{G}_i \in [1-\delta, 1+\delta] \qquad i \in [-\epsilon' n, \epsilon' n]$
- $|\hat{G}_i| < \delta \qquad i \notin [-\epsilon n, \epsilon n]$

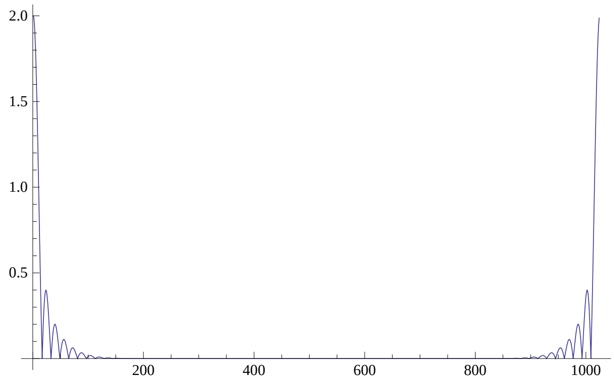

図 3 フラット窓関数 (時間領域)

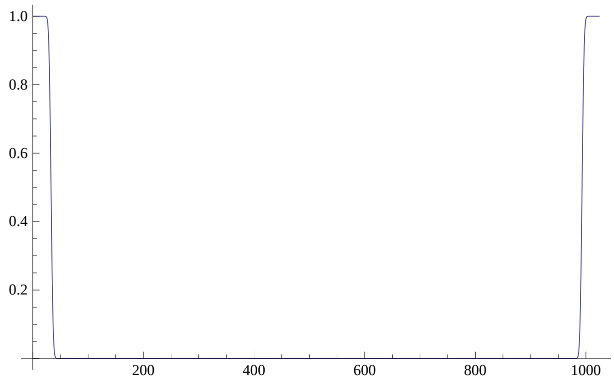

図 4 フラット窓関数 (周波数領域)

6 低次元 FFT

ベクトルを $x \in \mathbb{C}^n$ に対し，n の約数である自然数パラメータを B としたとき，$l \in \mathbb{Z}_B$ で，

$$q_l = \sum_{j=0}^{n/B-1} x_{l+Bj}$$

で定義される，B 次元ベクトル $q = \{q_0, q_1, \cdots, q_{B-1}\} \in \mathbb{C}^B$ のフーリエ変換を $\hat{q} = \{\hat{q_0}, \hat{q_1}, \cdots, \hat{q}_{B-1}\} \in \mathbb{C}^B$ とする．このとき，

$$\sqrt{\frac{B}{n}}\hat{q}_l = \hat{x}_{l(n/B)} \qquad (l \in \mathbb{Z}_B).$$

7 SFFT(Sparse Fast Fourier Transform)

SFFT は innner loop と outer loop の二つのアルゴリズムで構成され，以下のように記述する．

7.1 Inner Loop

任意の δ, $m = O(B \log \frac{n}{\delta})$ に対して，$(\frac{1}{B}, \frac{2}{B}, \delta, m)$ フラット窓関数を G とし，Location Loop は，パラメータ d が与えられ，各非ゼロ係数を高い確率で含む $\frac{dKn}{B}$ の集合 $I \subset [n]$ を出力し，Estimation Loop は集合 $I \subset [n]$ が与えられ，$\hat{x}_I$ は高い確率で推定する．

1. 奇数 $\sigma \in \mathbb{Z}_n$ と任意の整数 τ をランダムに選択する．
2. 積和合同置換
入力ベクトル x を置換する $x \mapsto y \quad (i \in \mathbb{Z}_n)$
$y_i = (P_{\sigma,\tau}x)_i = x_{\sigma i+\tau}, \quad \hat{y}_{\sigma i} = \widehat{(P_{\sigma,\tau}x)}_{\sigma i} = \hat{x}_i \omega^{-\tau i}$
3. フィルタの適用
フラット窓関数 G を用いて，置換されたベクトルを以下のように計算する．
$z_i = G_i \cdot y_i = G_i x_{\sigma i+\tau}$，$supp(z) \subseteq supp(G)$
4. サブサンプリング
置換されたフィルタをかけたベクトル z を B 個のビンの集合に切り取る．$L \ll \frac{n}{B}$ として，
$w_l = \sum_{l=-L}^{L} z_{l+Bj} \cong \sum_{j=0}^{n/B-1} z_{l+Bj} \quad (l \in \mathbb{Z}_B)$
5. 低次元 FFT：$\hat{w}_l = (\hat{w})_{l=0}^{B-1}$ を計算．
$\hat{w}_l \cong \sqrt{\frac{B}{n}} \hat{z}_{l(\frac{n}{B})}$ となる
6. 関数とオフセット関数
ハッシュ関数を定義する．
$h_\sigma \ : \ \mathbb{Z}_n \to \mathbb{Z}_B$, $h_\sigma(i) = round(\frac{\sigma i B}{n})$
オフセット関数を定義する．
$o_\sigma \ : \ \mathbb{Z}_n \to [-\frac{n}{2B}, \frac{n}{2B}]$, $o_\sigma(i) = \sigma i - \frac{n}{B} h_\sigma(i)$
7. Location Loop
$(\rho)_{i=0}^{n-1}$ の順列とし，$\hat{w}_i$ の成分を絶対値の大きい順に並べたものを．
$$|\hat{w}_{\rho(0)}| \geq |\hat{w}_{\rho(1)}| \cdots \geq |\hat{w}_{\rho(n-1)}|$$
とする．このとき，J を以下のように定義する．
$J = \{\rho(0), \cdots, \rho(K')\} \quad (K' = dK - 1)$
J に属する番号の，ビンに属する座標 i の集合を，
$$I = \{i \in \mathbb{Z}_n \mid h_\sigma(i) \in J\}$$
とする．
8. Estimation Loop
$i \in I$ に対して，$\hat{x}_i$ を以下のように推定する．
$$\hat{x}'_i = \frac{\hat{w}_{h_{\sigma(i)}} \omega^{\tau i}}{\hat{G}_{O_{\sigma(i)}}}$$

7.2 Outer Loop(個人選抜)

パラメータ ϵ, δ, $B = O(\sqrt{\frac{nK}{\epsilon \log \frac{n}{\delta}}})$, $d = O(\frac{1}{\epsilon})$ をもち，inner loop の実行回数を $L = O(\log n)$ とする．

1. Location loop の実行
L 回の Location loop を実行し，座標 $I_1, \ldots, I_L$ の L 個の集合を返す．
2. カウント
座標 i をを見つけるごとに出現数 s_i をカウントする．
$s_i = |\{r | i \in I_r\}|$

3. 座標の保持
   Location Loop で，$s_i \geq L/2$ の半分発生した座標 i を保持する．

$$I' = \{i \in I_1 \cup \cdots \cup I_L \mid s_i \geq \frac{L}{2}\}.$$

4. フーリエ係数 $\hat{x}'_i (i \in I')$ の推定
   $i \in I'$ に対し，適当な inner loop で，Estimation Loop を 1 回実行し，フーリエ係数 $\hat{x}'_i$ の推定を行う．

8 SFFT の実行結果

8.1 Inner Loop の実行結果

本研究では，$n = 1024$ の入力信号で，非零フーリエ係数 $(K = 4)$ をもつ入力信号の離散フーリエ変換 $\hat{x}$ の成分 $(221, 445, 637, 956) = 1$ で，その他は 0 というデータを作成し，$B = 16$ として，サブサンプリングと低次元 FFT を行い，SFFT を実行した．実行結果より，入力信号の離散フーリエ変換を高い確率で推定することができたが，アルゴリズムを実行していく中で，誤差が生じる結果の事例が生じた．Inner Loop は，location loop の回数を 1 回実行した結果をもとに，既知の位置データで，estimation loop を行い，入力信号を推定するアルゴリズムとする．

8.2 Estimation Loop の成功例

非零フーリエ係数が，一つしか含まれないビンの Estimation Loop は成功する．次の例では，４つの非零フーリエ係数すべての推定が成功する．

$$\mathrm{ha} = \{11, 2, 13, 7\}$$
$$\mathrm{WF} = \{1., 0.914928, 0.914928, 1.\}$$
$$\mathrm{o} = \{5, 29, 29, 4\}$$
$$\mathrm{gF} = \{1., 0.914928, 0.914928, 1.\}$$
$$\mathrm{estimatedx[[ii]]} = \{1., 1., 1., 1.\}$$

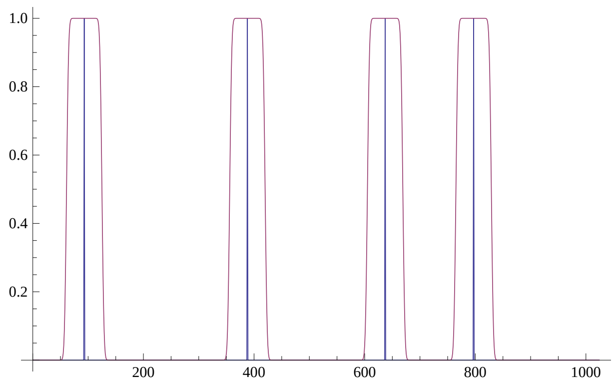

図 5 置換した信号とフラット窓関数を畳み込みした信号

8.3 Estimation Loop の失敗例

非零フーリエ係数が複数含まれるビンは失敗する．次の例では，１番目の非零フーリエ係数の推定精度が劣化する．

$$\mathrm{ha} = \{16, 15, 10, 7\}$$
$$\mathrm{WF} = \{1.00361, 0.914928, 0.914928, 0.999996\}$$
$$\mathrm{o} = \{5, 29, 29, 20\}$$
$$\mathrm{gF} = \{1., 0.914928, 0.914928, 0.999996\}$$
$$\mathrm{estimatedx[[ii]]} = \{1. - 0.0850721 I, 1., 1., 1.\}$$

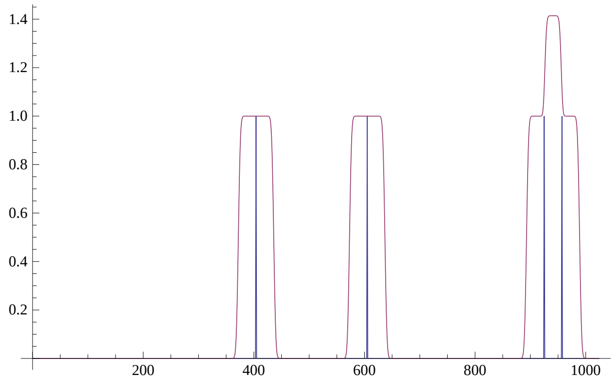

図 6 置換した信号と窓関数を畳み込みした信号

8.4 Outer Loop の結果

本研究では，location loop の実行回数 $(L = 50, 100)$ で行い，実行回数の半分以上の位置データを抽出し，その位置データをもとに，estimation loop で推定した．

8.5 ランダム実験の成功例

$L = 50$ のとき

$$\text{位置データ} = \{176, 265, 377, 824\}$$
$$\mathrm{ha} = \{1, 11, 14, 6\}$$
$$\mathrm{WF}\{1., 1., 1., 0.99696\}$$
$$\mathrm{o} = \{17, 6, 12, 25\}$$
$$\mathrm{gF} = \{1., 1., 1., 0.99696\}$$
$$\mathrm{estimatedx[[iii]]} = \{1., 1., 1., 1.\}$$

$L = 100$ のとき

$$\text{位置データ} = \{64, 171, 199, 689\}$$
$$\mathrm{ha} = \{12, 11, 10, 15\}$$
$$\mathrm{WF}\{1., 1., 0.999996, 1\}$$
$$\mathrm{o} = \{1, 8, 20, 17\}$$
$$\mathrm{gF} = \{1., 1., 0.999996, 1.\}$$
$$\mathrm{estimatedx[[iii]]} = \{1., 1., 1., 1.\}$$

8.6 ランダム実験の失敗例

推定位置データ $= \{533, 146, 337, 470, 536\}$

入力信号の位置データ $= \{147, 338, 471, 537\}$

ha $= \{11, 12, 11, 6, 11\}$

WF $= \{0.408201, 0.980173, 0.408201, 0.999999, 0.408201\}$

o $= \{24, 27, 22, 19, 9\}$

gF $= \{0.998987, 0.980186, 0.999919, 0.999999, 1.\}$

estimated x[[iii]] $= \{0.124674 + 0.389131I, 0.999986 + 0.000081384I, 0.0832468 - 0.399657I, 1., 0.0833951 + 0.399592I\}$

8.7 誤って選ばれる周波数の期待値

L 回の Location loop のうち，$\frac{L}{2}$ 回以上の出現回数があったとき，非零周波数の位置データとして選択される．この際に，固有値が 0 の周波数が誤って選ばれるときがある．その個数の平均値を計算すると，誤って選ばれる周波数の期待値を，n：信号の大きさ，m：出現個数，B：ビンの個数，k：非ゼロフーリエ係数の個数，L：location loop の出現回数，p：k がビンに含まれる確率とし，以下のような期待値で推定した．

$$E = \sum_{n=\frac{L}{2}}^{L} (1-p)^{(L-m)} p^m \binom{L}{m} \cdot n \tag{5}$$

式 (5) を，$L = 50, L = 100$ のとき，半分の値 $(25, 50)$ が生じる期待値は，0.0823796,0.000172405 となり，$L = 50$ のときより $L = 100$ のとき，極端に半分の値の誤って選ばれる周波数の期待値が下がる．この結果，$L = 100$ 回の実行結果でかなりの期待値で選択されないことがわかる．

8.8 location loop での見解

Inner Loop は，非零フーリエ係数の位置データを特定することができる．この結果より，location loop は非零周波数がかかる K 個のビンが重要であることがわかる．信号の離散フーリエ変換がフラット窓関数を畳み込む際に，非零成分の周波数を含むビンが重なる部分が生じ，位置データに誤差が生じる．このように，location loop は１回では，失敗する可能性があるため，複数回する必要がある．

実行回数が $L = 50$ のとき，Outer loop の出現回数 s_i が実行回数の半分以上の位置データを推定する際に，誤って選ばれる周波数を推定してしまう場合が生じる．実行回数が $L = 100$ とき，Outer loop を数十回実行しても，半分の 50 回に至ることはなかった．

8.9 estimation loop での見解

非零成分の周波数を含むビンが K 個の場合，正確に非零フーリエ係数を推定できる．また，ビンで衝突し，推定する箇所に大きな膨らみが生じた場合，位置データに誤差が生じることから，非零フーリエ係数は異なる解となり，正しく推定することが出来ないこともある．

位置データが K 個正確に抽出される場合は，入力信号の離散フーリエ変換の非零成分のフーリエ係数を推定できるが，位置データが K 個以外の場合，正しく推定される部分もあるが，精度としてはあまり良くない結果となって推定された．

9 まとめ（今後の課題）

SFFT のアルゴリズムの結果が上手くいかない事象をもとに，誤って選ばれる周波数を抽出しないように，location loop の実行回数を増やすことが重要であると判明した．極端に,location loop の実行回数を増やすと，計算量が大きくなるので，実行結果を制限する必要がある．

本研究のアルゴリズムは主に，location loop に依存している．入力信号に対して，擬似乱数をパラメータとした積和合同置換を実行しており，location loop 内の組織的な理論が曖昧になっており，結果依存での location loop を組むことで，理論的に追求する必要がある．フラット窓関数と低次元 FFT は，計算量 $(B \log B)$ で出来る利点もあるので，組織的に構築した位置データをもとに行う事で、より効率のよい高速フーリエ変換のアルゴリズムが出来るのではないかと考えられる．また，先行研究である MIT の研究者である Hassanieh らは，新たな改良した SFFT のアルゴリズムを提唱しているので，新しい理論を追加するのも本研究のさらなる課題でもある．

参考文献

[1] H. Hassanieh,P. Indyk, D. Katabi,and E. Simple and Practical Algorithm for Sparse Fourier Transform. SODA,2012

[2] 和田山正,「圧縮センシングにおける完全再現十分条件について」, 日本神経回路学科誌，Vol. 17，No. 2(2010),63-69

[3] E. Kushilevitz and Y. Mansour. Learning decision-trees using the fourier spectrum. STOC, 1991

[4] A. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. STOC, 2002.

[5] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding.FOCS, pages146, 2003

[6] A. Gilbert, M. Muthukrishnan, and M. Strauss.Improved time bounds for near-optimal space fourier representations.SPIE Conference, Wavelets, 2005.

[7] M. A. Iwen. Combinatorial sublinear-time fourier algorithms.Foundations of Computational Mathematics, 10:303 338, 2010.

[8] A. Akavia. Deterministic sparse fourier approximation via fooling arithmetic progressions.COLT, pages381393, 2010.