

PapersReading Notes

张亮

2023 年 4 月 19 日

目录

1	HHT(Hilbert-Huang Transform)	3
1.1	Introduction	3
1.2	EMD(Empirical Mode Decomposition)	3
1.3	Hilbert Transform	4
1.4	Additional from me	7
1.5	Conclusion	9
1.6	Reference	9
2	MC&Quasi-MC	9
2.1	Standard MC	9
2.2	Quasi MC	11
2.3	Reference	11
3	Multivariate empirical mode decomposition	11
3.1	Abstract & Introduction	11
3.2	Existing multivariate extensions of EMD	11
3.3	The proposed n-variate EMD algorithm	13
3.4	Simulations	13
3.5	Additional from me	13
3.6	Reference from me	15

1 HHT(Hilbert-Huang Transform)

HHT 是经验模态分解 (EMD) 和希尔伯特谱分析 (HSA) 的结果。

1.1 Introduction

一般来说, 要分析变幅正余弦信号的瞬时频率是一件很困难的事, 因为对于周期信号而言, 要求得频率需要一个周期内的完整波形才能将之实现。而变幅信号很难确定一个完成的波形且不同幅值的信号很可能不在同一种周期内, 计算瞬时频率其实是一件很困难的事情。

对于瞬时频率来说, 会有如下的操作方法, 首先将信号通过 Hilbert 变换将信号转换为另一个与原始信号相位相差 $\frac{\pi}{2}$ 的数据, 通过 Gabor 的 Analysis Signal methods 其实可以发现, 这两个信号都属于复平面在实轴-时间轴和虚轴-时间轴平面的投影。这样就能对一般的变幅正余弦信号求瞬时频率了。

而对于信号本身来说, 自然界里的信号不全是所谓的正余弦信号这种平稳的信号, 可能会存在很多复杂的信号, 所以就有了黄锬将非平稳信号分解成多个简单信号的叠加 (这就是 EMD 了)。这样就能分析波形的诸如瞬时频率等其他的一些特征了。

希尔伯特-黄变换其实是一种将信号分解为所谓的固有模态函数 (intrinsic mode functions, IMF), 是一种被用来处理非线性、非平稳信号的方法。

疑问点: HSA(希尔伯特谱分析)见章节末的补充

HHT 的第一步就是将原始信号分解为 IMF(Intrinsic mode functions), 将信号分解成不同的 IMF 会有不同的结果, First IMF 是最震荡 (高频) 的部分,

1.2 EMD(Empirical Mode Decomposition)

EMD 算法是在时域角度进行分析的, 具有自适应性、数据驱动性和高效性。由于分解是基于数据的局部特征时间尺度, 因此它可以应用于非线性和非平稳过程。

•Intinsic mode functions(IMF)

1. IMF 在整个数据集上的极值点和过零点的数据只能相等或者相差为 1。
2. 在任意点, 局部极大值的包络和局部极小值的包络的均值为 0。

它代表了与简单调和函数相对应的一种普遍简单的振荡模式。根据定义, IMF 是任何具有相同数目的极值点和零点交叉点的函数, 其包络线对称于零。这一定义保证了 IMF 的希尔伯特转换表现良好。

疑问: 包络 (envelope) 是什么?

包络是由信号的极大值点 (或极小值点) 通过插值方法形成的一条曲线, 通常来说, 上包络线 (极大值形成) 和下包络线 (极小值形成) 上的数值点的数量相等 (或者称为对称)。

一般的 EMD 算法如下所示:

算法 1 Standard EMD algorithm

1. 找出振荡波型的极值集合 $x'(k)$;
2. 对极大值 (极小值也一样) 使用三次样条插值的方法获取振荡波形的上包络 $e_{max}(k)$ (极小值是下包络 $e_{min}(k)$);
3. 计算均值 $m(k) = \frac{1}{2}[e_{max}(k) + e_{min}(k)]$
4. 做差: $s(k) = x'(k) - m(k)$ 得一个振荡波形信号, 若该信号满足 EMD 的停止原则, 则将该信号 $s(k)$ 作为一个 IMF 分量 $d_i(k)$; 否则进入下一步;
5. 将 $s(k)$ 作为新的 $x'(k)$ 再次进行新一轮的计算。

可以通过指定获得的 IMF 分量数或其他的原则停止 IMF 的继续生成, 最后会得到一个残差项 $r(k)$, 则会有一下式子

$$x(k) = \sum_{i=1}^N c_i(k) + r(k)$$

常见的几种 EMD 停止原则：

1. EMD 分解层数达到预设值：EMD 分解会将原始信号分解成多个本征模态函数 (IMF)，每个 IMF 表示一个时频分量。为了避免过度分解，可以预先设定分解层数，当分解层数达到预设值时停止分解。
2. 每个 IMF 的剩余部分与噪声比例达到预设阈值：EMD 前期的 IMF 代表高频信息，后期的 IMF 代表低频信息。当 IMF 的剩余部分与噪声比例达到预设阈值时，我们认为该 IMF 已经捕获到了信号中的主要特征，可以停止分解。
3. 相邻两个 IMF 的频带宽度比例达到预设范围：EMD 分解后得到的每个 IMF 在时频平面上呈现出不同的频带宽度，通常随着 IMF 的序号增加而逐渐减小。当相邻两个 IMF 的频带宽度比例达到预设范围时，我们认为已经提取到了主要的时频信息，可以停止分解。
4. SIFT 算法的信号强度不再变化：EMD 分解后得到的每个 IMF 可以看作一个时频分量，SIFT 算法利用局部极值点来提取主要的时频信息。当信号在某个 IMF 上的局部极值点数量不再变化时，我们认为该 IMF 已经捕获到了信号中的主要特征，可以停止分解。

可以使用 Python 自带的 Pyemd 包将数据信号进行 MED 分解 (以 EEG 信号为例)，因为这个数据信号是来自 Bonn 大学的癫痫脑电信号图，不知道如何进行数据切片，所以仅做了一个原始信号的 EMD 分解和随机切片 (0: 250) 的 EMD 分解。基于的停止原则是预设最大 EMD 分解层数。

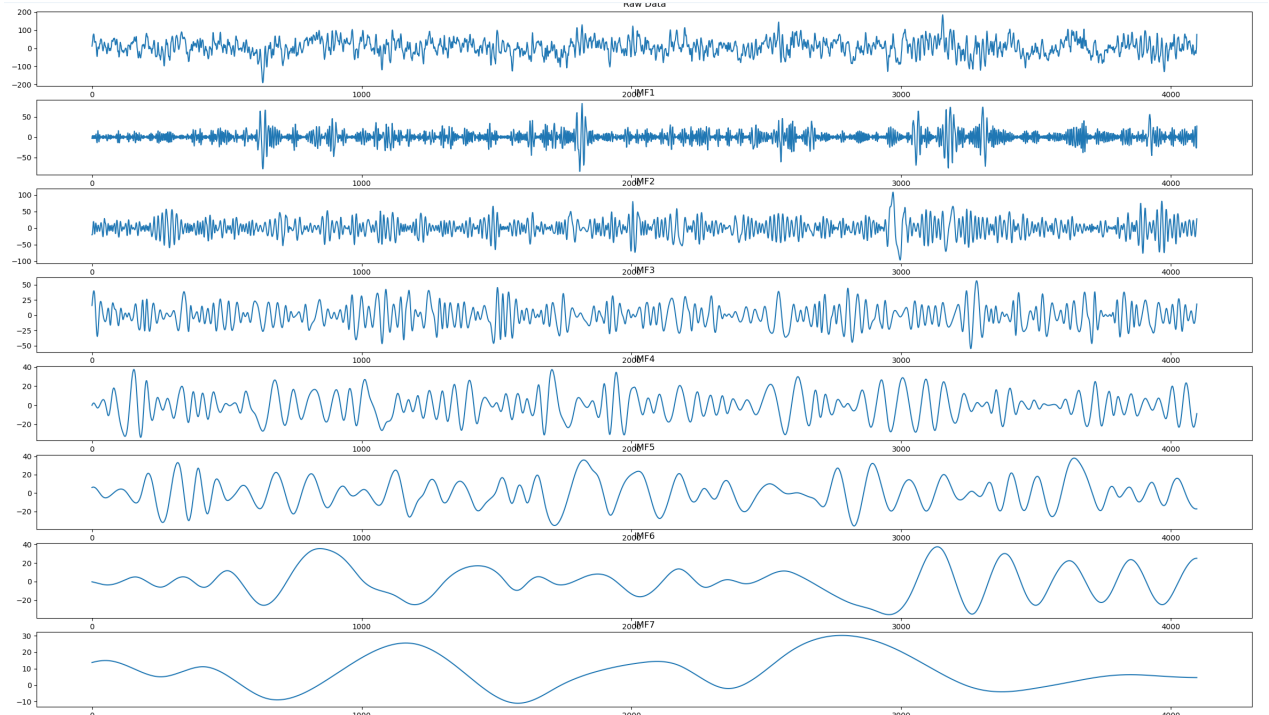


图 1: 原始 EEG 信号 EMD 分解图

1.3 Hilbert Transform

因为本人自身局限性，对于公式的理解始终差点意思，所以只能从应用的角度来分析以下 HT 变换及 HT 变换和 FT 变换之间的关系。

对于一般的正余弦波形去求解信号的频率或幅值是一件相对简单的事情，因为正余弦实信号具有周期性，只需要用周期 T 除以 2π 即可。

但是，对于更一般的实信号 (周期不确定的信号)，要计算局部频率，至少需要一个完整周期的正弦波形才行，而对于一般信号 (在周期内振荡速率在明显改变)，其瞬时频率是很难计算的。

而希尔伯特变换 (Hilbert Transform) 即可以解决这个问题，同时，由于原信号 (实信号) 和解析信号同频，所以可以求出相对应的瞬时频率。

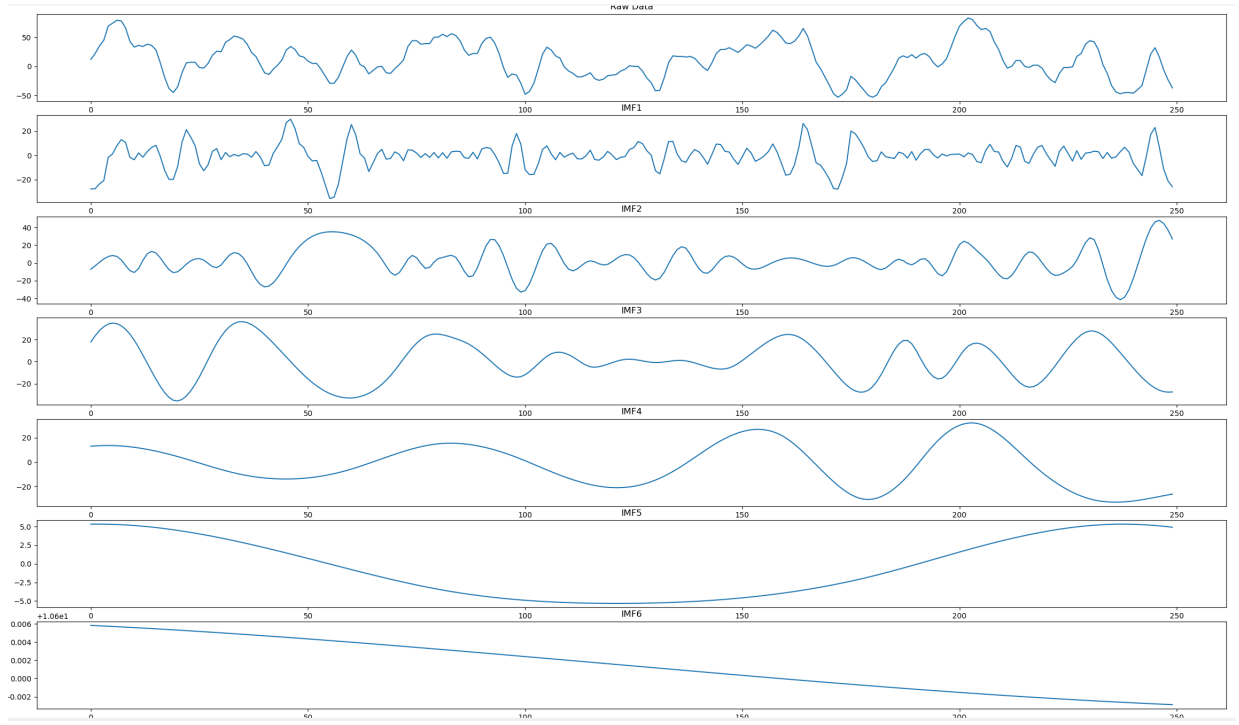


图 2: 切片 EEG 信号 EMD 分解图

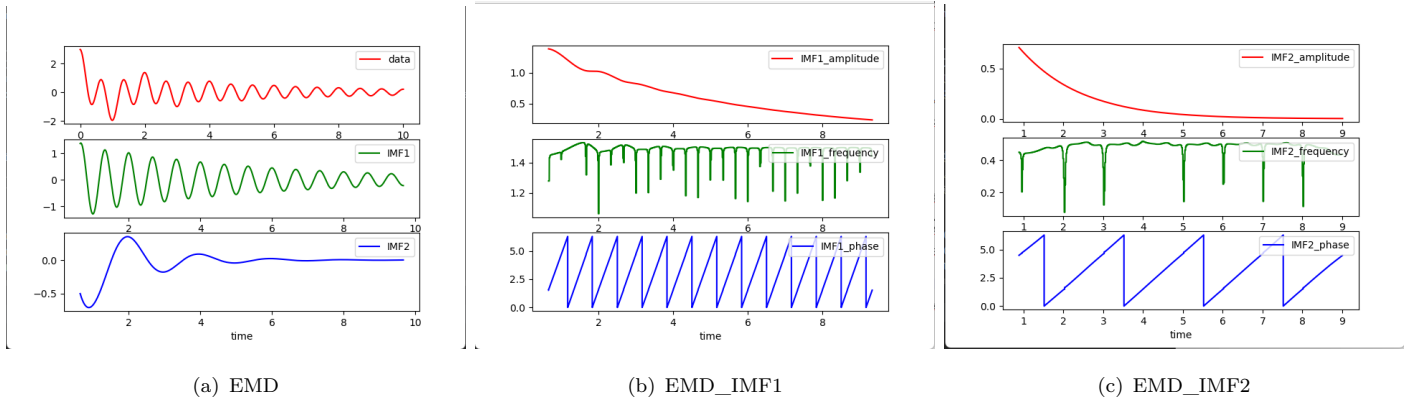


图 3: EMD 分解及幅值图

希尔伯特变换其实是能够求出实信号所对应的复信号，并且两个信号间相差相位为 $\frac{\pi}{2}$ ，最后将两种信号合并，成为最终的解析信号。

如果定义原始的实信号为 $x(t)$ ，那么所找到的虚信号的虚部即为 $y(t)$ ，最终经过希尔伯特变换获得的解析信号为

$$a(t) = x(t) + jy(t) = A(t)e^{j\theta(t)}$$

。其中， $y(t)$ 是 $x(t)$ 经过希尔伯特变换所得到的信号，后半部分等式由欧拉公式得到。即

$$y(t) = H(x(t))$$

其中， H 表示希尔伯特变换。

那么幅值的瞬时表达式为：

$$A(t) = \sqrt{x^2(t) + y^2(t)}$$

相位 $\theta(t)$

$$\theta(t) = \arctan \frac{y(t)}{x(t)}$$

那么，瞬时频率即如下：

$$\omega(t) = \frac{d\theta(t)}{dt}$$

而 HHT 具有以下特征：

1. HHT 反映的信号频谱特征随时间变化的规律。
2. HHT 可以对局部特征进行反映，这点主要得益于 EMD 的作用。EMD 可以自适应地进行时频局部化分析，有效提取原信号的特征信息。

下面是关于 HT 变换的公式定义及相关性质。

• Hilbert Transform

函数 $u(t)$ 的希尔伯特变换定义为

$$H(u)(t) = \frac{1}{\pi} p \cdot v \cdot \int_{-\infty}^{+\infty} \frac{u(\tau)}{t - \tau} d\tau$$

其中， $p \cdot v \cdot$ 表示柯西主值 (the Cauchy principal value)，也就是瑕积分的积分值。也可以用 P 替换。

上式也可以转换成

$$H(u)(t) = \frac{2}{\pi} \lim_{\epsilon \rightarrow 0} \int_{\epsilon}^{\infty} \frac{u(t + \tau) - u(t - \tau)}{2\tau} d\tau$$

从定义上来看，某函数的希尔伯特变换就是该函数和函数 $h = \frac{1}{\pi t}$ 的卷积

希尔伯特变换可以看作是在单位圆上将信号旋转 $\frac{\pi}{2}$ 所致，所以，希尔伯特变换存在逆变换即为 H^3 。可以这样理解，希尔伯特变换将信号转换成了解析信号的形式输出，而解析信号的额模即为包络线。

- 希尔伯特变换和傅里叶变换在频率域上，对函数 f 进行希尔伯特变换，实质为在 f 的傅里叶变换结果 $F(\omega)$ 前乘一个系数 $H(\omega) = -j \operatorname{sgn}(f)$

即相当于把 $F(\omega)$ 的所有正频率向后移动 $\frac{\pi}{2}$ 相位，所有负频率向前移动 $\frac{\pi}{2}$ 相位，即如下：

$$\begin{aligned} H(t) &= \hat{f}(t), \\ &= f(t) * \frac{1}{\pi t}, \\ &= F(f)[-j \operatorname{sgn}(f)], \end{aligned} \tag{1}$$

根据卷积定理，时域上的卷积等于频域上的相乘。

$$F\{f(t) * g(t)\} = F\{f(t)\} \cdot F\{g(t)\}$$

$$F(f)[-j \operatorname{sgn}(f)] = \begin{cases} -jF(f), & f \geq 0 \\ jF(f), & \text{otherwise} \end{cases}$$

其中 H 表示希尔伯特变换， F 表示傅里叶变换。

• Hilbert spectral analysis

希尔伯特谱分析其实是基于两个式子由

$$\omega = \frac{d\theta(t)}{dt}$$

这个式子已经在前面完成了推导。

和希尔伯特变换可以推导出

$$X(t) = \sum_{j=1}^n a_j(t) \exp(i \int \omega_j(t) dt)$$

这样， $X(t)$ 其实就是一个关于时间和频率的函数，那么就可以在时频图上去分析信号了，这就是希尔伯特谱分析了，这也是希尔伯特·黄变换的一个重要部分。

1.4 Addtional from me

补充 1 EMD 的数据驱动性和自适应性

- 数据驱动性和自适应性:

EMD 算法 (Empirical Mode Decomposition) 被称为自适应的和数据驱动算法, 是因为它能够自适应地对信号进行分解, 并且不需要预先指定信号的基函数。

EMD 算法是一种数据驱动的信号分解方法, 它可以将信号分解成若干个本征模态函数 (Intrinsic Mode Functions, IMFs), 每个 IMF 都代表了信号中的一种局部特征。EMD 算法不需要预先指定基函数, 而是通过局部极值点的连接和插值, 构造出 IMF 函数。这使得 EMD 算法能够适应各种信号的形态和变化, 具有非常好的自适应性。

因此, EMD 算法被认为是一种自适应的和数据驱动算法, 能够高效地对信号进行分解和处理。EMD 算法已被广泛应用于信号处理、图像处理、语音处理、生物医学信号处理等领域, 例如信号去噪、特征提取、信号压缩等方面。

- 插值方法和 Embedding:

- 插值方法:

插值方法是一种常用的信号处理技术, 它可以用于在一些离散数据点之间估计缺失的数据点的值。插值方法将现有的数据点视为已知值, 并在这些数据点之间插入新的数据点, 以估计缺失数据点的值。插值方法通常需要定义一个插值函数, 该函数可以根据已知数据点之间的关系来估计缺失数据点的值。插值方法通常适用于连续的信号数据, 例如时间序列数据、图像数据等。

- Embedding 方法

Embedding 方法是一种将高维数据映射到低维空间的技术, 它可以用于提取数据中的结构信息。Embedding 方法通常通过构造数据的特征向量来实现, 这些特征向量可以反映数据中的局部结构和相对位置关系。Embedding 方法可以用于各种类型的数据, 例如文本数据、图像数据、语音数据等, 并且可以提供有关数据的降维表示, 以便进行分类、聚类、相似性计算等任务。

- 插值方法 (Interpolation)

从某种意义上来说, 插值方法其实是一种变形的拟合, 不过这种拟合是一种及其过度的拟合, 需要拟合到所有数据的特征, 然后从拟合函数上找到除样本点外的一些数据点 (或者称为未知点)。

一般的插值方法包括但不仅限于以下几种:

- 线性插值:

线性插值是用一系列首尾相连的线段依次连接相邻各点, 每条线段内的点的高度作为插值获得的高度值。利用插值点与其他数据点 (区间内) 的斜率值与区间端点的梯度值进行插值;

一般有如下形式, 设 x_1, x_2 为插值区间, x_k 为插值点, 有

$$\frac{x_1 - x_2}{y_1 - y_2} = \frac{x_1 - x_k}{y_1 - y_k}$$

- 多项式插值 (拉格朗日插值)

对 n 个样本点过拟合出一个最高次可能为 $n-1$ 的多项式, 一次性拟合, 就比如下图:

使用插值后得到的一个 5 阶多项式为

$$f(x) = -0.0001521x^6 - 0.003130x^5 + 0.07321x^4 - 0.3577x^3 + 0.2255x^2 + 0.9038x$$

而这种方法在加入新的数据点时, 会使得函数更新一次, 并且可能会导致龙格现象。

所以可以所用样条插值的方法, 例如使用三次样条插值:

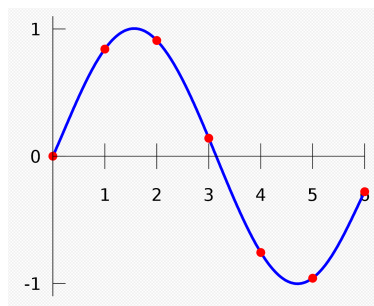


图 4: n 次多项式拟合图

◦ 样条曲线插值

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & \text{if } x \in [0, 1], \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1296, & \text{if } x \in [1, 2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & \text{if } x \in [2, 3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & \text{if } x \in [3, 4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & \text{if } x \in [4, 5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.9282, & \text{if } x \in [5, 6]. \end{cases} \quad (2)$$

• Embedding

Embedding 时离散实例连续化的映射。

A 4-dimensional embedding

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4

图 5: 将离散数据 Embedding 成四维图示例

Embedding 对于推荐系统中深度学习主要应用可以简单总结如下:

- 作为 Embedding 层嵌入到深度模型中, 实现将高维稀疏特征到低维稠密特征的转换 (如 Wide&Deep、DeepFM 等模型);
- 作为预训练的 Embedding 特征向量, 与其他特征向量拼接后, 一同作为深度学习模型输入进行训练 (如 FNN);
- 在召回层中, 通过计算用户和物品的 Embedding 向量相似度, 作为召回策略 (如 Youtube 推荐模型等);
- 实时计算用户和物品的 Embedding 向量, 并将其作为实时特征输入到深度学习模型中 (如 Airbnb 的 embedding 应用)。

几种比较主流的 Embedding 技术:

- 经典的矩阵分解方法: 这里主要是介绍 SVD 方法;
- 基于内容的 embedding 方法: 这部分主要涉及到 NLP 相关的文本 embedding 方法, 包括静态向量 embedding (如 word2vec、GloVe 和 FastText) 和动态向量 embedding (如 ELMo、GPT 和 BERT) ;

- 基于 *Graph* 的 *embedding* 方法：这部分主要是介绍图数据的 *embedding* 方法，包括浅层图模型（如 *DeepWalk*、*Node2vec* 和 *Metapath2vec*）和深度图模型（如基于谱的 *GCN* 和基于空间的 *GraphSAGE*）

补充 2 功率谱 功率谱是一种将一个信号分解成不同频率的成分的方法，因此它提供了关于信号在各个频率上的强度信息。功率谱的单位通常为功率密度 (*power density*)，即每单位频率内的功率，通常用瓦特/赫兹 (W/Hz) 或分贝/赫兹 (dB/Hz) 表示。

在连续时间域中，可以对功率密度进行积分来计算功率，即

$$P = \int_{f_{min}}^{f_{max}} S(f) df$$

其中, $S(f)$ 表示对应的功率密度函数, P 表示功率, f_{min} 和 f_{max} 分别表示最小频率和最大频率。

功率密度函数是指在频域上，每个单位频率上的信号功率。它可以表示为信号的傅里叶变换 (*FFT*) 的模的平方。而在连续时间域上，信号在每个频率 f 上的功率密度函数 $S(f)$ ，如下表是：

$$S(f) = \frac{|F(f)|^2}{2\pi}$$

其中, $F(f)$ 表示傅里叶变换 (频域)。

1.5 Conclusion

目前来说，HHT 可能用于 spike 检测方面的方法可能有以下：

1. **Energy Ratio**，这种方法来自于一篇 15 年的 paper，通过对比三种不同棘波和无棘波的 IMFs 的 Energy Ratio 参数，可以发现一个简单的特点：IMF1 在无棘波时的 Energy Ratio 比其他三种棘波的 Energy Ratio 都高。
2. **MEMD** 这种方法在一种专利中被使用，且获得了较好的表现。但需要细究其算法。

1.6 Reference

对于 HHT 的粗略理解：[希尔伯特-黄变换 \(HHT\) 的前世今生——一个从瞬时频率讲起的故事](#)

HHT 进一步了解：[Hilbert-Huang transform](#)

可以先看看这个，再看上面的维基百科 [希尔伯特变换详解 \(the Hilbert transform\)](#)

[数值分析：数据插值方法](#)

2 MC&Quasi-MC

2.1 Standard MC

从历史角度来分析，蒙特卡洛算法的原理只是通过随即均匀采样估算某类数据，而最早的蒙特卡洛思想出现在布丰投针计算圆周率 π 的实验中，直到二战时期，由于美国曼哈顿原子弹计划的需要，在计算中子在原子弹中的增殖和扩散过程中，需要求解高维玻尔兹曼方程 (高阶偏微分方程 (或者说是在高维空间中的积分))。

MC 方法的一大应用——求积分。

而通常来说，计算形如

$$F(x) = \int_{-\infty}^{\infty} f(x)g(x)dx$$

在 $g(x)$ 或 $f(x)$ 较为复杂的情况下，这个积分方程是难解的，甚至说可能解不出。

而蒙特卡洛算法即是为了解决这一问题的。

举个比较简单的例子：对于一个服从高斯分布的 $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ，需要计算其中样本 x 的平方 $g(x) = x^2$ 在整个分布上的期望，即

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

显然，这个积分很难求出，甚至是不可能解出。

而根据大数定律，当样本数量足够多的情况下，样本的期望是在一定条件下与总体分布的期望相等的，即

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx = \frac{1}{n} \sum_{i=1}^n g(x_i)$$

那么，现在的问题即转换成怎样去抽取足够多且满足分布特点的样本了。

显然，最直接的思路是这样的，通过概率密度 $f(x)$ 求出对应的分布 $F(x) \in [0, 1]$ ，那么，肯定存在这样的关系：对于在 $(0,1)$ 区间内的均匀随机采样 $i \in (0, 1)$ ，必定存在

$$F(x) = i$$

因为 $F(x)$ 是一个单调不减函数，所以 $F(x)$ 可逆，即

$$x = F^{-1}(i)$$

但是， $F(x)$ 也不一定就能解出，就像上面的高斯分布的密度函数。

所以，我们需要找一个分布 $q(x)$ ，满足以下两点：

1. $q(x)$ 可积分。

2. $mq(x) \geq f(x)$

那么就可以在 $q(x)$ 采样了，但是，从图像上来看，采样到的样本肯定是满足 $q(x)$ 的，但不一定是满足 $f(x)$ 。所以就有了基本的接受-拒绝采样方法了。即以 $p = \frac{f(x)}{mq(x)}$ 接受该采样，表示该样本既满足 $mq(x)$ ，又满足 $f(x)$ 。

当采到足够多的样本时，就可以用了作为积分的近似了。

这是一个例子

计算积分 $\int_0^1 \sqrt{x-x^2}$

乍看之下，可以直接使用三角代换来计算实现，答案是 π 。

但如果使用 Monte Carlo 方法呢？

$$\int_0^1 \sqrt{x-x^2} = \int_0^1 dx \int_0^{f(x)} dy = P(Y \leq f(X))$$

即对 X 在 $(0,1)$ 上采样，对 Y 在 $(0,1)$ 上采样，满足式子

$$x^2 + y^2 \leq 1$$

最后，用满足条件的样本点除以总的样本点个数，即可获得相应的积分值。

以下为算法的代码实现：

```
1 import random
2 import math
3
4
5 def montecarlo(N):
6     i = 0
7     count = 0
8     while i <= N:
9         x = random.random()
10        y = random.random()
11        if pow(x,2) + pow(y,2) < 1:
12            count += 1
13            i += 1
```

```

14     pi = 4*count /N
15     print(pi)
16     print(math.pi)
17
18     if __name__ == "__main__":
19         montecarlo(80000)

```

输出如下： »3.14975

»3.141592653589793

2.2 Quasi MC

蒙特卡罗方法和拟蒙特卡罗方法的主要区别在于采样方法不同。蒙特卡罗方法使用随机采样来估计积分或求解问题，而拟蒙特卡罗方法使用一组确定性的点来代替随机点，从而减少采样误差。此外，拟蒙特卡罗方法还可以使用低差异序列或高斯序列等更加均匀的点分布来提高精度。

Quasi Monte Carlo 方法同一般的 Standard 方法来说，根据采样随机序列 (low discrepancy series) 的不同，可能会有不同的结果，但所得结果较一般的结果更均匀且收敛速度更快。如图 21 所示。

下图是关于 MonteCarlo 和 Quasi-MC 之间生成样本的样本空间差异图。

2.3 Reference

蒙特卡洛 (Monte Carlo, MCMC) 方法的原理和应用

徐亦达机器学习：Markov Chain Monte Carlo 马尔科夫蒙特卡洛 (MCMC)

这篇 Paper 上有关于 MonteCarlo 可行性的证明及 MC 和 QMC 之间的关系：Monte Carlo and quasi-Monte Carlo methods

3 Multivariate empirical mode decomposition

3.1 Abstract & Introduction

主要介绍了 EMD 的数据驱动性和在非线性、非平稳信号上的利好，但存在以下两点不足：

1. 存在模板匹配的问题，这个需要仔细阅读论文 (因为 1.Huang(2008) 提出过一种关于模态混合的缺点；2.Looney 和 Mandic(2009) 提出了一种模板匹配的问题) 需要特别注意一下两者之间的区别。
2. 多元信号求包络的问题尚未得到解决，所以提出了一种使用 1. 均匀角坐标和 2. 拟蒙特卡洛方法 (quasi-Monte Carlo methods) 求解问题。

3.2 Existing multivariate extensions of EMD

EMD 算法需要满足的条件：

1. 具有尽可能对称的上下包络；
2. 过零点的数目和极值点的数目相等或者相差为 1。

一般的 EMD 算法过程见 HHT 中的 MED 部分。

二维和三维的 EMD 模态分解的扩展是基于 MED 的**旋转不变性**，这个是在一篇论文中提到的，将多信号进行旋转处理后再进行 EMD 分解。过程较为复杂，需要花点时间去理解。主要是两篇 IEEE 的论文 *Altaf, M. U., Gautama, T., Tanaka, T. & Mandic, D. P. 2007 Rotation invariant complex empirical mode decomposition.* 和 *Rilling, G., Flandrin, P., Goncalves, P. & Lilly, J. M. 2007 Bivariate empirical mode decomposition.*

旋转不变性 (rotation invariant) 保证了每个 IMF 具有普相同数目的幅值和相位。(Cite from Altaf etc. 2007)

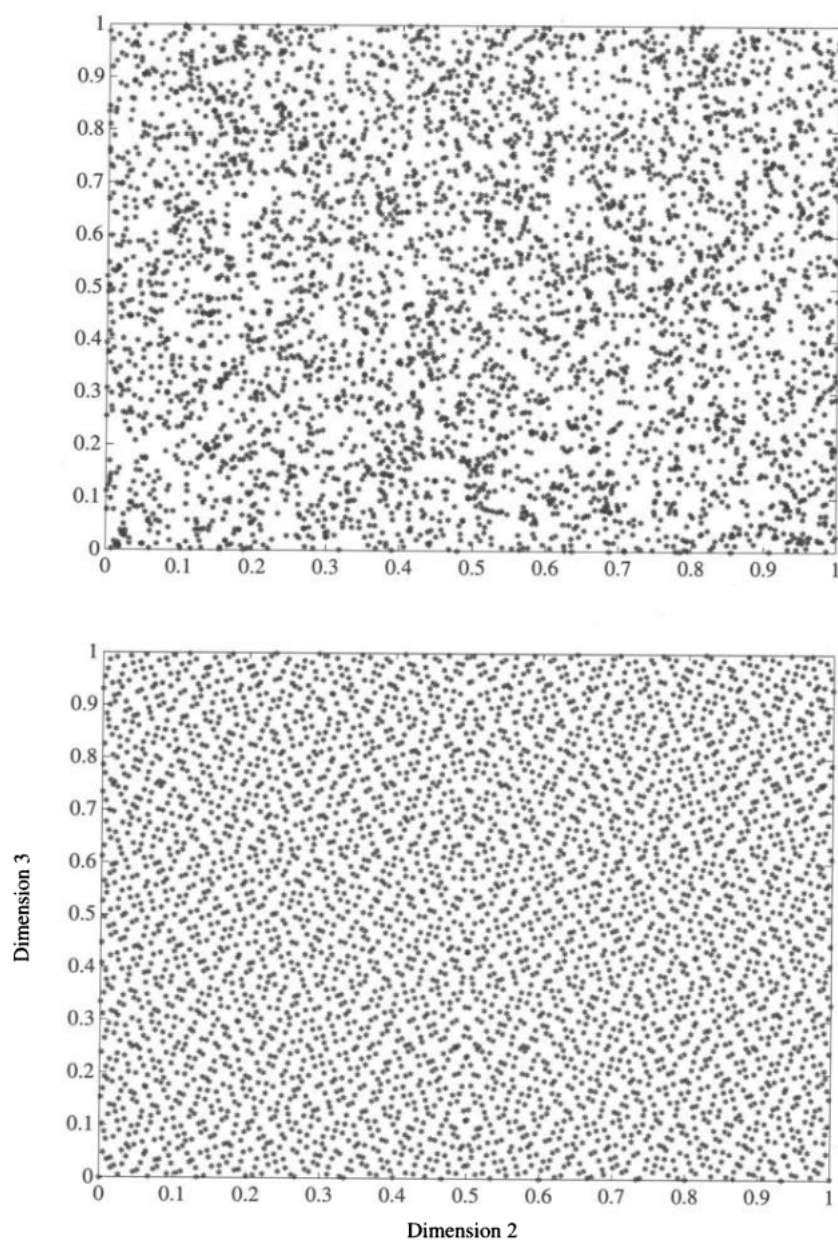


图 6: 拟蒙特卡洛 (底) 和蒙特卡洛 (顶) 差异图

3.3 The proposed n-variate EMD algorithm

疑问点 1 多元曲线的局部均值可以理解成在 n 维空间上多方向的一个积分的近似。这个近似的准确性取决于方向向量选取的均匀性。

- 关于球面

球面是一类基本的形状。例如, 1 维球面是圆圈, 2 维球面是球体的表面。一般地, 也有 n 维球面, 通常记为 S^n , 它是拓扑空间和 n 维流形的基本例子。

- 球面的定义:

设 $n \geq -1$ 是整数。 n 维球面定义为 Euclid 空间 \mathbb{R}^{n+1} 的子集

$$S^n = \{(x_1, \dots, x_{n+1}) \in \mathbb{R}^{n+1} | x_1^2 + \dots + x_{n+1}^2 = 1\}$$

它带有 \mathbb{R}^{n+1} 的子空间拓扑, 成为拓扑空间。当 $n > -1$ 时, 它也是 \mathbb{R}^{n+1} 的子流形, 这定义了它的流形结构。

例如, $S^{-1} = \emptyset$ 是空空间, S^0 是两个点构成的离散空间, S^1 是平面上的单位圆周, S^2 是空间中的单位球面。

在 $n \in \mathbb{N}_+$ 时, 从球面上挖去一个点后 (S^n / x_0) 和 \mathbb{R}^n 通过球极射影同胚。

使用均匀采样方法不能产生均匀的分布的原因: **因为极点的两点。**

使用 Quasi Monte Carlo 方法随机采样出更加均匀的序列, 这个方法较一般 Standard Monte Carlo 方法来说收敛速度更快, 且产生序列更均匀。

3.4 Simulations

使用 MEMD 进行模板匹配和旋转模板的提取。

3.5 Additional from me

补充 3 模板匹配 (Mode Alignmet) 和模式混合 (Mode Mixing)

- 模式混合 在不同频率尺度上出现相似模式的现象。(David Looney and Danilo P. Mandic, 2009)

补充 4 多元样条插值

样条插值在每个间隔中使用低阶多项式, 并选择多项式以使它们平滑地吻合在一起。结果函数被称为样条曲线。这种方法在前一章写插值方法中有提到。

多元插值是函数变量多余一个时所使用的插值方法, 因为变量在空间坐标系中, 所以, 这种插值方法又叫做空间插值 (spatial interpolation)。

补充 5 Halton 序列 & Hammersley 序列

- Halton 序列算法

Halton 序列是一种低差异度的无偏采样点生成算法。它是基于一个简单的观察: 为了获得一个均匀分布的样本点集合, 我们可以使用一组不同的质数作为基础, 将每个维度映射到一个序列中。

具体来说, 在 Halton 序列中, 我们选择一组互不相同的质数 x_1, \dots, x_n 作为基础, 其中 n 是数据点的维度。则在 1 维 Halton 序列中第 i 个样本为

$$r_i^x = \sum_{k=0}^s \frac{a_k}{x^{k+1}}$$

其中 m 是预先定义的整数, a_k 是一个介于 0 和 x_i 之间的整数, 用于确定该维度上的 n 个样本点中的第 k 个点。

$$i = \sum_{k=0}^s a_k x^k$$

其获取到的采样序列为 $(r_i^{x_1}, r_i^{x_2}, \dots, r_i^{x_n})$

- Hammersley 序列 这种采样的个数 n 是事先知道的, 其获取的第 i 次样本为 $(i/n, r_i^{x_1}, r_i^{x_2}, \dots, r_i^{x_{n-1}})$

两者采样结果

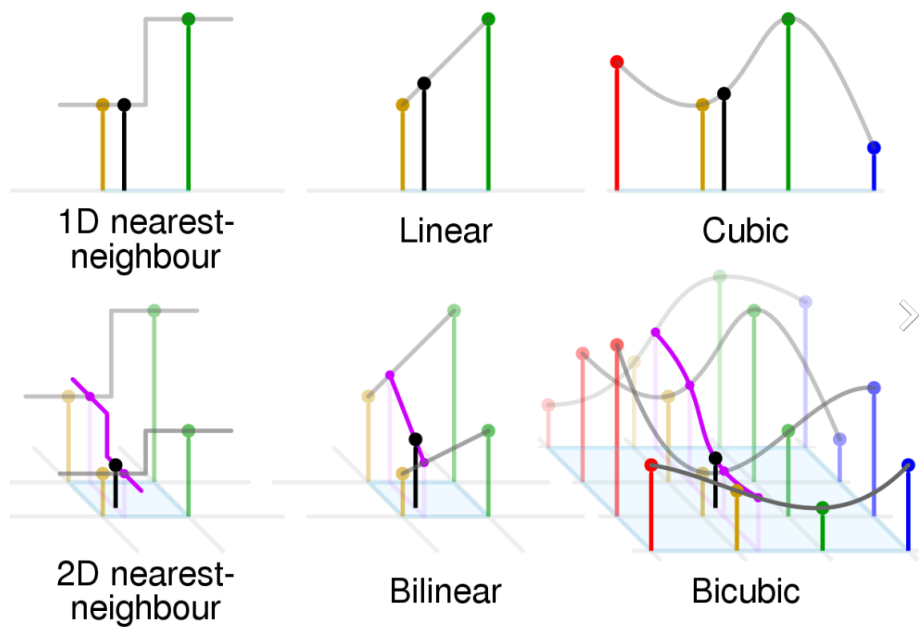


图 7: 多元插值方法示意图 (最近邻插值、线性插值、三次插值)

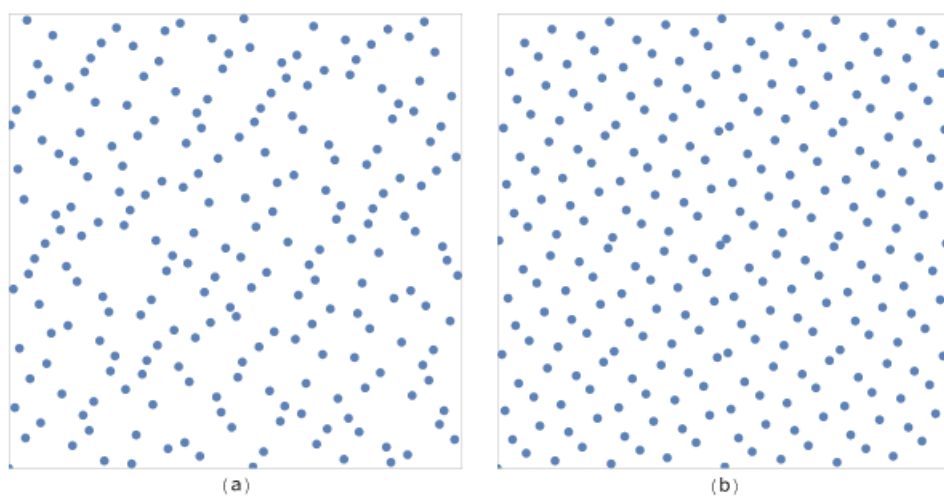


图 8: Halton 序列算法 (a) 和 Hammersley 序列算法 (b)

3.6 Reference from me

[EMD Wiki](#)

[球面](#)

[拓扑空间](#)