## 多元时间序列的分段高斯贪心算法GGS

### **模型内容**

在机器学习中，很多无监督学习算法，试图切分较大的复杂数据集成为较小的简单数据集，如主成分分析PCA，高斯混合聚类GMM，来达到对数据进行降维、降噪，包括异常点检测和风格迁移发现的目的。PCA是一种高斯协方差分析方法，能够提取数据主要的坐标成分，以此达到降维和降噪的效果。GMM利用混合高斯模型做密度分析，是一种数据样本软聚类方法。

这些方法，同样在多元时间序列分析中发挥着重要的作用，我们转化为分段高斯协方差正则最大似然问题：

* **正则化协方差问题**

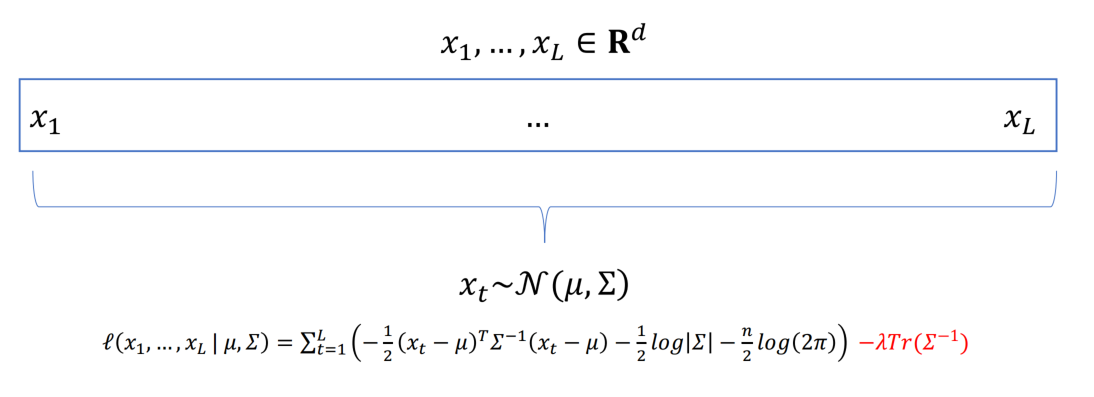
给定时间序列，并假设，数据序列的Log似然为：

|  |  |
| --- | --- |
|  | （1） |

的问题，即样本维度数量远大于样本数，PCA降维投影方法对经验协方差矩阵做SVD分解，去掉较小的特征值分量，从而获得稳定的估计量，并利用它进行投影。在高维统计中，一种经典方法是直接对做正则化限制，选取一种度量，如trace模即：

|  |  |
| --- | --- |
|  | （2） |

示意图如下：



, 其它方法，如稀疏逆约束和稀疏PCA等，都是高维统计中常见的方法。正则化方法的好处是：

* 获得高维数据的协方差逆估计的稳定值，将正则化约束转化为优化求解问题。事实上，对矩阵trace模的大小限制，等价于对一类矩阵范数的大小进行限制，包括秩、最大特征值、最大行范数、谱半径等。因而，我们的联合似然求解只在约束可行解内求得，控制了估计量的预测性方差。
* 正则化项具有的降噪和降维效果。事实上，高维统计算法，如稀疏逆估计Sparse Inverse Covariance等，提供了具有其它统计特性的正则化选项，具有特征筛选和降噪的效果。而在本文中，超参数控制了trace模对的限制，进而在优化过程中平衡对似然率的提升（模型拟合、参数优化是否充分）和的自由度（对预测方差的限制）。超参数直接控制了模型选择，我们将在后文中介绍如何学习超参数。
* 分段高斯问题

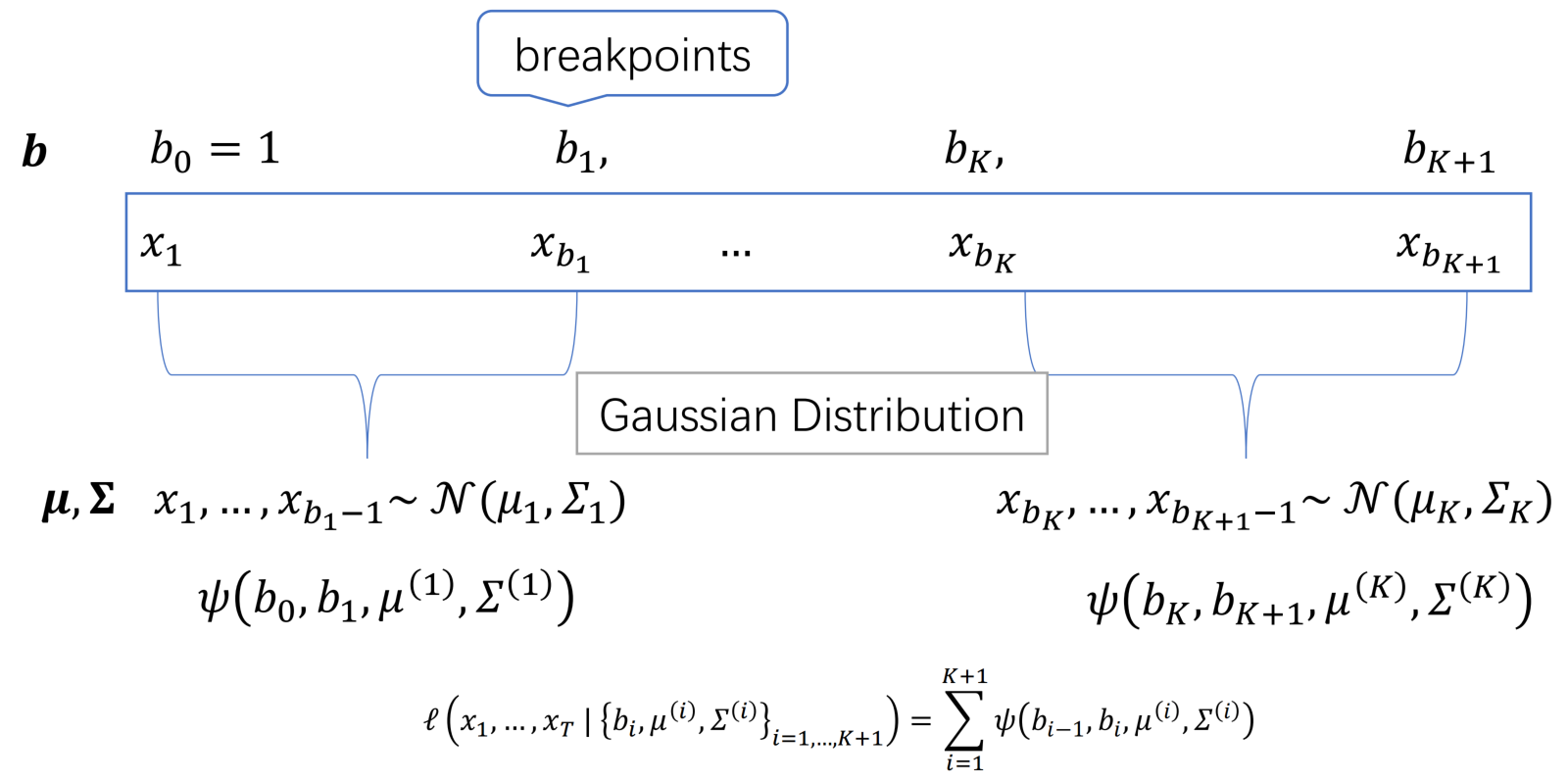
对于一段多元时间序列, 并假设, 其中高斯分布的均值和协方差仅会在切分点（breakpoint）发生改变，个切分点将时间序列切分为K+1个段（segment）。决定了切分点数量。而在每一个切片内，由相同的多元高斯分布生成，即：

|  |  |
| --- | --- |
|  | （3） |

其中，切分点满足，有

|  |  |
| --- | --- |
|  | （4） |

下标表示时间序列中的序号，上标表示第段划分，相应的划分起始位置的序号和。我们将这个模型叫做分段高斯模型Gaussian Segmentation Model(GSM)，示意图如下：



结合上面的正则化分段高斯问题的Log最大似然如下：

|  |  |
| --- | --- |
|  | （5） |

在GSM中，记 中的 ， 为 ，并简记为或者：

|  |  |
| --- | --- |
|  | （6） |

* 优化目标与模型参数

给定超参数与，优化目标为：

|  |  |
| --- | --- |
|  | （7） |

并且我们只考虑情形，即 必须是一组正定的协方差来保证和的数值为正数。

超参数对正则化似然的影响表现为：较大的K使得模型增加了更多的参数数目，将时间序列划分为更多更小的段，模型复杂度越高。同时，每一切分段内的样本数量变得更少，需要较大的来收缩逆协方差，控制高斯分布的似然提升，提高估计稳定性。如何利用交叉验证（Cross Validation）或样本外的贝叶斯信息率来确定K与 的值，将在后面的评价指标中介绍。如果能达到相同的评价指标，我们倾向于使用更小的K（更容易解释数据的模型）和更大的（更稳定的参数模型）。

在K与 的值确定的情况下，有个选择，我们转化为最大最大化问题：

|  |  |
| --- | --- |
|  | （8） |

在 给定下优化 则有：

|  |  |
| --- | --- |
|  | （9） |

其中为经验均值也为估计协方差矩阵，为经验协方差矩阵，为估计协方差矩阵：

|  |  |
| --- | --- |
|  | （10） |

注意上面的由于小于维数，所以是奇异的，但它一定是半正定的，所以必然是正定的，满足。事实上，估计的形式类似于协方差常用的Stein-type收缩估计(Ledoit&Wolf)。

给定超参数K与，正则化最大似然GSM模型最大化转变为对划分点的搜索问题：

|  |  |
| --- | --- |
|  | （11） |

* 计算代价分析与调优

给定时间序列，在GSM模型假设下，给定K与 以及，我们分析上述浮点计算代价。

公式组（10）K个分段内，经验均值需要次运算和个存储，需要次运算和。通常时间序列划分段数满足 ，例如，序列长度为1000，维度为20维，我们划分段数最多不超过50，以保证分段内的样本数量。所以，通常情形下，协方差矩阵不超过原始数据的存储代价。与均可以并行处理，使用K个处理器，均值计算代价满足 ，协方差计算代价满足。令，可以控制计算代价最多不超过阶，并且接近阶。

公式（9）在一个分段内计算似然值，需要阶次运算，矩阵相乘和求行列式均低于阶次运算，所以搜索过程的（9）的时间开销不会高于协方差计算步骤，在评估过程中，我们需要进行矩阵相乘和行列式，可以利用Cholesky 分解来减少浮点数计算和行列式求解的计算代价。对第i步的做如下分解：

|  |  |
| --- | --- |
|  | （12） |

这一步Cholesky分解需要次运算， 是正对角元的下三角矩阵，行列式计算需要次运算，即。逆运算为次，即。总体复杂度不高于，并行时间为量级。

* GGS模型的不定分段数量推广

我们上述讨论的GGS模型，是在给定超参数分段数量与正则化系数下，搜索优化目标函数，其中满足为互不相同的个划分点。

如果，我们约定划分点满足，即划分点可以重合，则优化目标函数性质不会改变。此时超参数表示最大切分点数量，我们记为。这个问题退化为找到满足互不相同划分点最优解即可。

### **搜索算法**

* 动态规划搜索

正则化最大似然（11）可以利用Bellman动态规划Dynamic Programming算法进行划分点的搜索。划分点对 的状态基数为 ，序列可以拆分为子序列状态搜索问题：如果状态, 则只由子序列的最优切分决定，因而搜索代价为，整体复杂度为。对于较小，较短的时间序列可以利用动态规划求解，我们关注通用性更强的近似线性复杂度算法。

* **1-OPT性质**划分的定义

不能通过改变（挪动和新增）任一划分点来提高目标函数的序列划分。

* 子路径划分Split Subroutine

第i段时间序列的切分片段，分段似然 。子路径划分算法寻找满足的划分点t，使得切分片段分为两个部分与的似然加和最大。

|  |  |  |
| --- | --- | --- |
| 算法1 寻找最优子切分段 Split Subroutine | | |
| 输入：时间序列片段和经验均值和协方差 | | |
| 1 | ; | |
| 2 | for do: | |
| 3 |  | Update ; |
| 4 |  | Calculate |
| 5 |  | If Then and ; |
| 6 | End for | |
| 返回：划分点 和 似然差 | | |

运行算法1的时间为阶，算法记为。

* 划分点调整 Adjust Breakpoints

|  |  |  |  |
| --- | --- | --- | --- |
| 算法2 划分点调整 Adjust Breakpoints | | | |
| 输入：时间序列片段 | | | |
| 1 | Repeat | | |
| 2 |  | for do: | |
| 3 |  |  | Calculate Empirical Parameters of as Eq10; |
| 4 |  |  | ; |
| 5 |  |  | If set ; |
| 6 | Until Stationary | | |
| 返回： | | | |

运行算法2的时间为阶，算法记为。算法2基于目前的划分段对划分序列进行调整，提升似然目标或者维持不变，循环直到满足**1-OPT性质**。

* 贪心高斯划分 Greedy Gaussian Segmentation GGS

|  |  |
| --- | --- |
| 算法3 贪心高斯划分 Greedy Gaussian Segmentation | |
| 输入：时间序列片段 | |
| 1 | 初始化 |
| 2 | for do: |
| 3 | for do: |
| 4 | Calculate Empirical Parameters of as Eq10; |
| 5 | ; |
| 6  7 | If All are negative and :  Then Keep and **Break**; |
| 8 | Add a new breakpoint to at the with the largest corresponding value of ;  Relabel thecollection with members; |
| 9 |  |
| 返回： | |

算法3 GGS算法采用贪心策略，每次循环分为两个部分: (Add步)和 (Adjust步)。其中，Add步可以并行计算。Add步在原有的划分基础上，顺序遍历所有的分段，寻找到一个提高目标函数的新的划分点，将该段分裂为两个并重新标记的序号。Adjust步在划分基础上，顺序遍历所有的（相邻）分段，对相邻两个分段寻找最优划分点并替换原有的划分点。Adjust步会循环多次直到稳定。因而满足Add步退出条件的划分，必然满足**1-OPT性质**，既不能新增一个划分点，也不能挪动一个划分点来提升目标函数。

我们指定超参数，即划分点数量的最大值，GGS算法给出满足的解，直到该解满足**1-OPT性质**而停止。很明显，满足的解必然符合**1-OPT性质**，但满足**1-OPT性质**的解未必是最优解，比如我们并未检验同时挪动或者新增两个或者多个划分点目标函数是否可以继续提升，GGS是解决GSM模型的一种贪心近似搜索策略，也是确定性算法。

* 复杂度分析

不考虑并行化，GGS运行步迭代需要的时间复杂度为,其中是算法2的平均迭代次数，我们无法确定的上界，经验性结论是,因而对较大的和适中的，GGS算法相较于动态规划是近似线性的。

* 加速搜索和变种

算法3的简化版本，随机初始化划分点，运行满足**1-OPT性质**,时间复杂度为。的一个回溯变种是，每次移除掉一个似然率降得最少的划分点，再增加一个似然率提升最多的样本点。同样的回溯方法，可以运行从个划分点开始向下调整直到满足**1-OPT性质**。

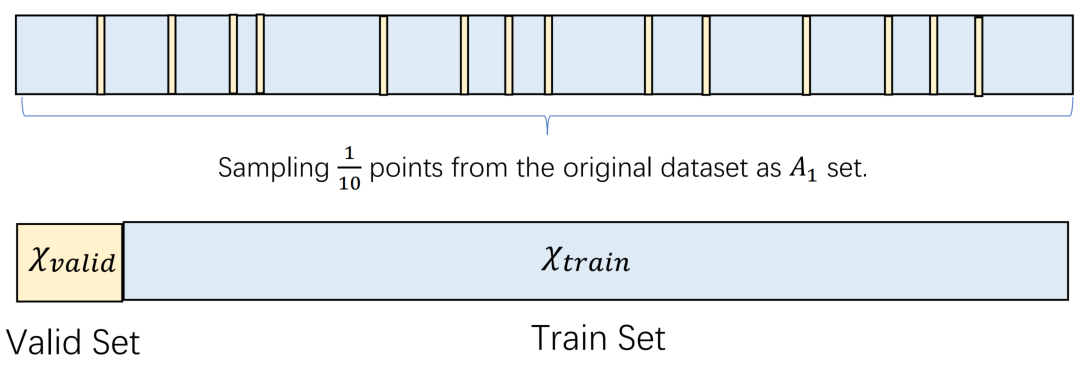
### **训练方法**

* 观点回顾

统计学习观点，将最小二乘回归的错误分为偏差bias和方差variance两个部分，训练集上的优化解决的是如何降低bias，而bias降低必然意味着需要更大的模型容量（参数规模或者模型空间更大）和更精确的优化，这会导致预测variance变大（预测和推广能力变差），在机器学习领域，很多简单的方法如早期停止法（Early Stop）等，就是为了避免在训练集上做过度拟合（Overfit）。正则化方法利用结构化经验风险最小化Structural Empirical Risk Minimization来降低训练集的拟合程度（VC维增长），以提高模型的推广能力（泛化界），这些都由超参数来控制（见模型内容中的描述），那么怎么确定最优的超参数呢？首先，需要建立描述bias和variance的实验性近似评价指标，即下面的和。

* 评价指标和样本外测试

GSM模型有两个超参数（hyperparameters） 和，常用的方法如BIC(或者AIC)即贝叶斯、Akaike信息率（详见Hastie统计学习基础Ch7）。我们通过样本外测试来选择超参数。对输入的长度的时间序列，随机抽取出10%的数据样本，并留下0.9数量的样本。留下的样本作为训练集（Train Dataset），抽取的样本作为验证集（Validation Dataset）。



评估超参数，需要重复进行上述数据集的划分过程，在训练集上完成GGS算法，并得到训练集Train Set的划分和相应的参数估计，进而计算样本落在第i段内的似然率：

|  |  |
| --- | --- |
|  | （13） |

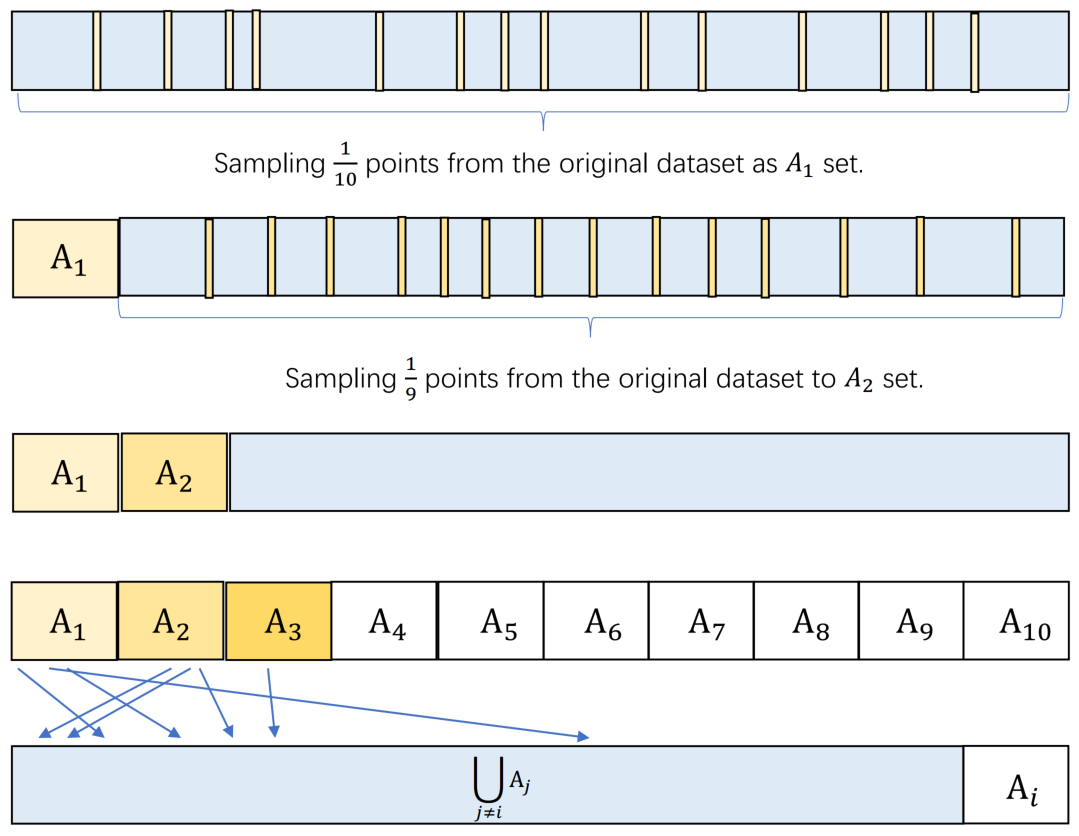
评价指标为“平均样本似然”（Average log-likelihood per sample），即：

|  |  |
| --- | --- |
|  | （14） |

我们依据可以确定训练集的平均样本似然,验证集数据点来自具体哪个分段也可以由确定，只不过验证集不参与训练没有划分点，记平均样本似然为。

* N-fold交叉验证

为了得到给定超参数和的和评价，需要进行多次随机抽样实验并取实验结果的平均值。N折交叉验证（N-fold Cross-Validation）是一种更为简单的方法，它将数据集划分为N份，每次做留1（leave-one-out）训练，下面给出示意图（N=10）：



通过平均抽样，将数据集划分为，样本互不交叉。运行GGS算法10次，每次利用进行训练得到模型，计算全体样本的平均训练似然率和留1验证似然率：

|  |  |
| --- | --- |
|  | （14）  （15） |

这就是的评价指标。

* Candidate模型

我们定义Candidate模型的标准：

容忍训练和验证指标的偏离范围，如果过大则表明出现过拟合，即在训练集上的似然率过度提升，理论上讲，一定存在一个更大的 或者更小的也能达到，从而降低使得模型容量小，更简单和稳定。的设置，将过滤掉那些过拟合却意外获得较高的那些超参数决定的模型，符合这一标准的作为Candidate模型。

令，即长度的时间序列有个分段，通常选择一个区间内的指数级别递增序列,记为。GGS算法给出相应下，从到时的评估值。如果GGS算法停止在，记所有的的为。我们找出符合Candidate标准的模型，且在参数下最优，具体搜索方法可以利用网格搜索或者插值搜索。