Time series to image conversion approach

Time series to image conversion approach

- Method 1:Gramian Angular Field (GAF)
  - 数学知识预备
    - 点积
    - Gram矩阵（Gram Matrix）
  - 实现方式
    - 缩放序列
    - 噪声图片
    - 非稀疏性
  - 开始编码
    - 缩放序列
    - 转换坐标
    - 时间序列的内积
    - 稀疏表示
- Method 2 :Markov Transition Fields
  - 实现方式
    - 建立QxQ马尔可夫转移矩阵
    - 马尔可夫转移场MTF
- 未来展望
- *Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images*
  - Contribution
  - Methodology
    - 使用分段聚合近似（PAA）进行降维
    - 将时间序列数据编码为图像
    - 图像与串联
    - 卷积网络架构
    - 采用数据集
  - Future Work

虽然现在深度学习在计算机视觉和语音识别上发展得很好，但是碰到时间序列时，构建预测模型是很难的。原因包括循环神经网络较难训练[Recurrent Neural Networks are difficult to train]、一些研究比较难以应用，而且没有现存预训练网络，1D-CNN 不方便。但是如果将时间序列转换成图片，我们就能够充分利用目前机器学习计算机视觉上的优势。时间序列转换成图片的方法主要有三种：格拉姆角求和场（GASF），格拉姆角差场（GADF）和马尔可夫变换场（MTF）。

> **相关论文参考**
>
> 理论
>
> *Imaging Time-Series to Improve Classification and Imputation*

> *Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks*
>
> 应用
>
> *Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images*
>
> *Single Residential Load Forecasting Using Deep Learning and Image Encoding Techniques【上次分享过*

Method 1:Gramian Angular Field (GAF)

格拉姆角场GAF方法描述：将笛卡尔坐标系下的一维时间序列，转化为极坐标系表示，再使用三角函数生成GAF矩阵。

原文地址：https://medium.com/analytics-vidhya/encoding-time-series-as-images-b043becbdbf3

Imaging Time-Series to Improve Classification and Imputation https://arxiv.org/pdf/1506.00327.pdf

中文解释：https://zhuanlan.zhihu.com/p/83130649

实现代码：https://github.com/devitrylouis/imaging_time_series

数学知识预备

点积

内积是两个向量之间的运算，用来度量它们的「相似性」。它允许使用来自传统 *Euclidian Geometry* 的概念：长度、角度、第二维度和第三维度的正交性。

在二维空间上，两个向量 *u* 和 *v* 之间的内积定义为：$< u, v >= ||u|| \cdot ||v|| \cdot cos\theta$

如果 u 和 v 的范数为 1，我们就得到：$< u, v >= cos\theta$

因此，如果处理的是单位向量，他们的内积就只由角度θ决定了.

注意：在 Euclidian 集合（n 维）中，两个向量的内积的正式定义为：$< u, v >= \sum_{i=1}^{n} u_i \cdot v_i$

Gram矩阵（Gram Matrix）

定义：一组 n 个向量的 Gram 矩阵是由每一对向量的点积定义的矩阵。从数学上讲，这可以解释为：

$$G = \begin{pmatrix} \langle u_1, v_1 \rangle & \langle u_1, v_2 \rangle & \cdots & \langle u_1, v_n \rangle \\ \langle u_2, v_1 \rangle & \langle u_2, v_2 \rangle & \cdots & \langle u_2, v_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle u_n, v_1 \rangle & \langle u_n, v_2 \rangle & \cdots & \langle u_n, v_n \rangle \end{pmatrix}$$

假设所有的二维向量都是单位向量，我们会得到：

$$G = \begin{pmatrix} \cos(\phi_{1,1}) & \cos(\phi_{1,2}) & \cdots & \cos(\phi_{1,n}) \\ \cos(\phi_{2,1}) & \cos(\phi_{2,2}) & \cdots & \cos(\phi_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_{n,1}) & \cos(\phi_{n,2}) & \cdots & \cos(\phi_{n,n}) \end{pmatrix}$$

Gram 矩阵保留了时间依赖性。由于时间随着位置从左上角到右下角的移动而增加，所以时间维度被编码到矩阵的几何结构中。

实现方式

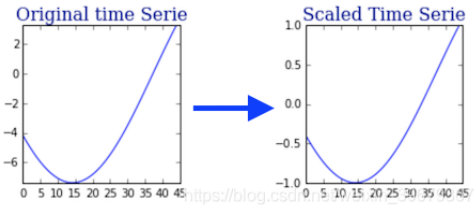
假设有一时间序列$X = x_1, x_2, x_3, x_4 \ldots x_n$

缩放序列

使用一个限定在 [-1,1] 的最小-最大定标器（Min-Max scaler）来把时间序列缩放到 [-1,1] 里，这样做的原因是为了使内积不偏向于值最大的观测。

Given a time series $X = \{x_1, x_2, ..., x_n\}$ of n real-valued observations, we rescale X so that all values fall in the interval $[-1, 1]$:

$$\tilde{x_i} = \frac{(x_i - max(X) + (x_i - min(X))}{max(X) - min(X)} \quad (1)$$

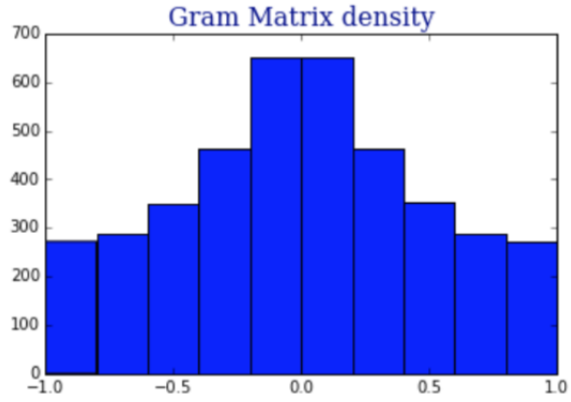


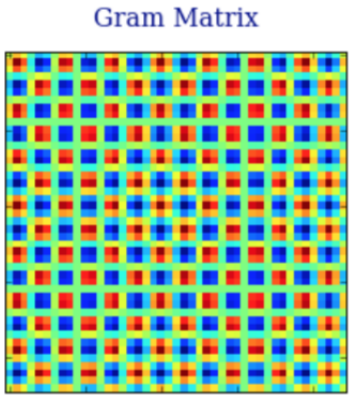
噪声图片

缩放完时间序列之后，我们计算每一对的点积并把它们放进 Gram 矩阵里：

$$G = \begin{pmatrix} x_1 \cdot x_1 & x_1 \cdot x_2 & \cdots & x_1 \cdot x_n \\ x_2 \cdot x_1 & x_2 \cdot x_2 & \cdots & x_2 \cdot x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n \cdot x_1 & x_n \cdot x_2 & \cdots & x_n \cdot x_n \end{pmatrix}$$

我们查看一下 G 的值来看一下这个图片：





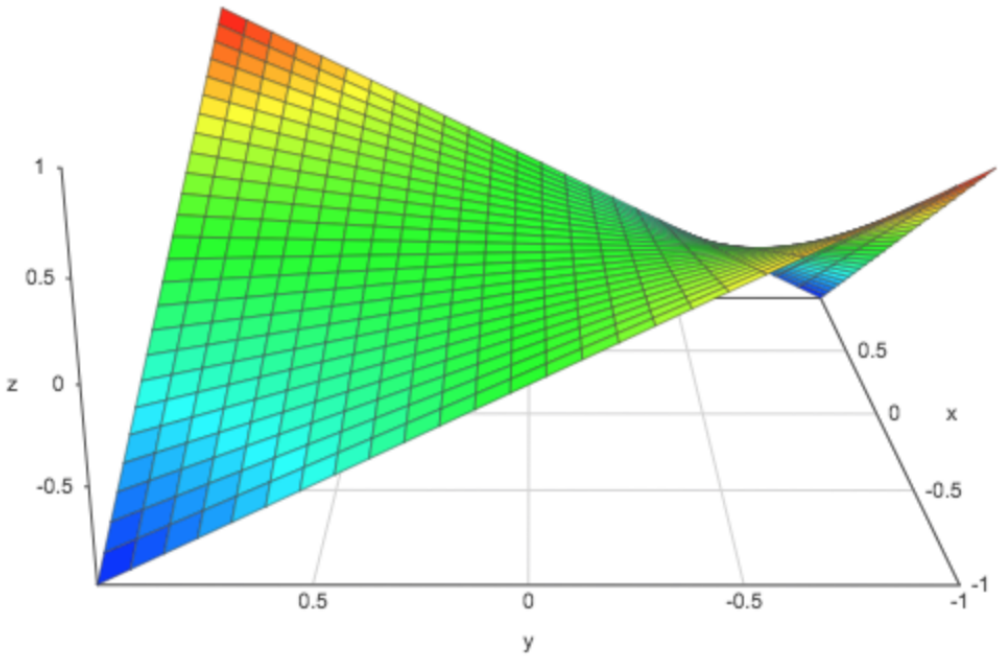
可以看到：

1. 输出似乎遵循以 0 为中心的高斯分布。
2. 得到的图片是有噪声的。

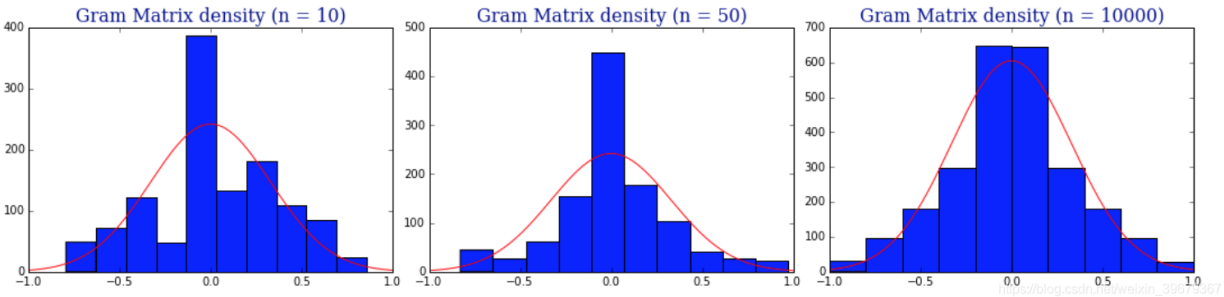
前者解释了后者，因为数据的高斯分布越多，就越难将其与高斯噪声区分开来。这对我们的神经网络来说是个问题。此外，**CNN 在处理稀疏数据方面表现更好**（CNN work better with sparse data）已经得到证实。

非稀疏性

高斯分布并不奇怪。当看三维内积值z的图像，对所有 (x, y)∈R^2的可能的组合,我们得到一个点积的三维表面：



假设时间序列的值服从均匀分布 [-1,1]，则矩阵的值服从高斯分布。下面是长度为 n 的不同时间序列的 Gram 矩阵值输出的直方图：



由上可以看出，如果直接进行简单的数学变换的话，得到的数据是含有较多噪声且不稀疏的，不便于CNN进行处理，因此我们对GAF进行了改进。

开始编码

由于单变量时间序列是一维的，点积不能区分有价值的信息和高斯噪声，除了改变空间，没有其他利用「角」关系的方法。因此，在使用类似于 Gram 矩阵的构造之前，我们必须将时间序列编码为至少二维的空间。为此，我们将在一维时间序列和二维空间之间构造一个**双射映射**，这样就不会丢失任何信息。

缩放序列

我们运用以下公式对时间序列进行缩放：

我们的极坐标编码将是双射的，使用 *arccos*函数双射（参见下一步)。

Given a time series $X = \{x_1, x_2, ..., x_n\}$ of n real-valued observations, we rescale X so that all values fall in the interval $[-1, 1]$:

$$\tilde{x_i} = \frac{(x_i - max(X) + (x_i - min(X))}{max(X) - min(X)} \qquad (1)$$

转换坐标

接下来我们需要构造一个一维时间序列到二维平面的双射映射，**将缩放后的时间序列转换到「极坐标」**。构造映射时需要考虑**两个量，时间序列的值及其对应的时间戳**。这两个变量分别用**角度**和**半径**表示。

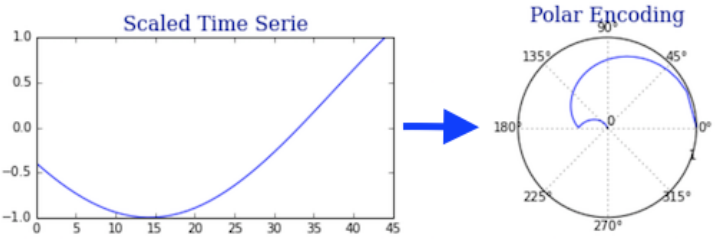
假设我们的时间序列由 **N 个时间戳** *t* 和对应的 *x* 组成，那么：

- 角度是用*arccos*(*x*) 计算的，值在[0,*π*] 之间。
- 首先计算半径变量，我们把**区间 [0,1] 分成 N 等份**。因此，我们得到 *N*+1 个分隔点 { 0,⋯,1 }然后我们丢弃 0，并连续地将这些点与时间序列关联起来。

用公式表示如下：

$\Theta_i = arccos(x_i)$

$r_i = \frac{i}{N}$



> 这些编码有几个优点：
>
> 1. 整个编码是**双射的**（作为双射函数的组合）/在给定时间序列的情况下，映射在极坐标系中具有唯一逆函数的结果只有一个。
> 2. 与笛卡尔坐标相反，**极坐标保留绝对的时间相关性**【用积分解释，当f(x(t))为在一段时间内为同一值，f(x(t))在不同时间的相同时间间隔内的积分是一样的，但是如果换在极坐标系里面，它们的积分是不一样的】。它通过 *r* 坐标保持时间依赖性。随着时间的增加，相应的值在跨度圆上的不同角点之间发生扭曲。
>
> nates preserve absolute temporal relations. In Cartesian coordinates, the area is defined by $S_{i,j} = \int_{x(i)}^{x(j)} f(x(t))dx(t)$, we have $S_{i,i+k} = S_{j,j+k}$ if $f(x(t))$ has the same values on $[i, i+k]$ and $[j, j+k]$. However, in polar coordinates, if the area is defined as $S'_{i,j} = \int_{\phi(i)}^{\phi(j)} r[\phi(t)]^2 d(\phi(t))$, then $S'_{i,i+k} \neq S'_{j,j+k}$. That is, the corresponding area from time stamp i to time stamp j is not only dependent on the time interval $|i-j|$, but also determined by the absolute value of i and j. We will discuss this in more detail in another work.

时间序列的内积

完成时间序列的编码问题后，接下来要考虑的是我们如何使用内积运算来处理**稀疏性**。

由于任何类似于内积的操作都不可避免地将两个不同观测值的信息转换成一个值，所以我们不能同时保留两个角度给出的信息。我们必须做出一些让步。

为了最好地从两个角度解释个体和连接信息，作者定义了内积的另一种操作：***x*⊕*y*=cos(*θ*1+*θ*2)**

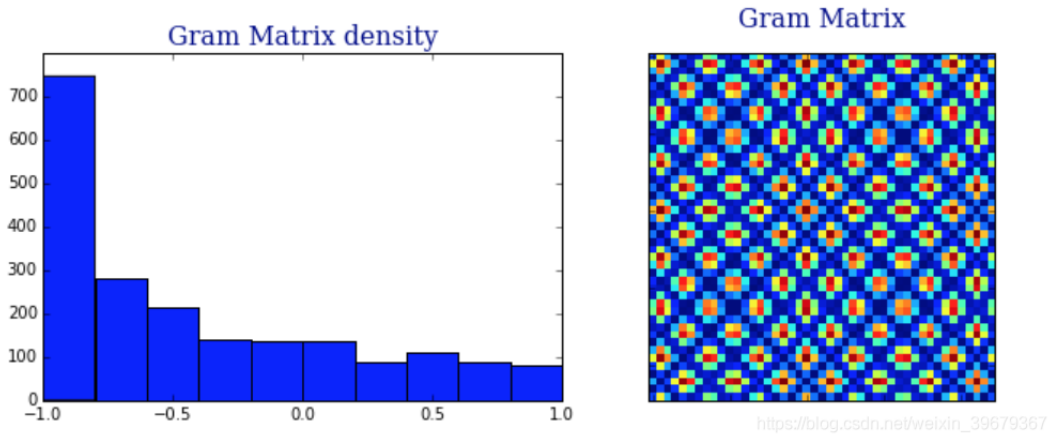
这就产生了如下的类 Gram 矩阵:

作者的选择这样做的动机是：相对于笛卡尔坐标，极坐标保留绝对的时间关系。

> 优势：
>
> 1. 对角线$G_{i,i}$由(缩放后的)时间序列的原始值(时间间隔k=0)构成（我们将根据深度神经网络学习到的高层特征来近似重构时间序列）；
> 2. 时间相关性是通过时间间隔 *k* 的方向叠加，用相对相关性来解释的。

稀疏表示

现在我们来画出格拉姆角场（Gramian Angular Field）的值的密度分布：



从上图中我们可以看出，格拉姆角场要稀疏得多，【更利于CNN处理。

为了解释这一点,让我们用 u⊕v 在笛卡尔坐标重新表示：

$$GASF: cos(\theta_1 + \theta_2) = cos(acrcos(x) + acrcos(y)) = x \cdot y - \sqrt{1-x^2} \cdot \sqrt{1-y^2}$$

$$GADF: sin(\theta_1 - \theta_2) = \sqrt{1-x^2} \cdot y - \sqrt{1-y^2} \cdot x$$

新构造的运算对应于传统内积的惩罚版本：

$x \oplus y = x \cdot y - \sqrt{1-x^2} \cdot \sqrt{1-y^2}$

可以看到：

- 惩罚将平均输出移向 -1；
- *x* 和 *y* 越接近0，惩罚越大。主要的原因是，这些点更接近高斯噪声；
- 对于*x*=*y*：会转换为 -1；
- 输出很容易与高斯噪声区分开。

缺点：

1、主对角线生成的 GAM 很大是由于 *n*↦*n*^2，而原始时间序列的长度为 n。作者建议通过使用**分段聚合近似PAA**（Piecewise Aggregation Approximation）减少大小，平滑时间序列，提高分辨率。

2、这个操作不是真正意义上的内积。

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.axes_grid1 import ImageGrid, make_axes_locatable
from pyts.image import  GramianAngularField
'''
读取时间序列的数据
怎么读取需要你自己写
X为ndarray类型数据
'''
# Transform the time series into Gramian Angular Fields
gasf = GramianAngularField(image_size=28, method='summation')
X_gasf = gasf.fit_transform(X)
gadf = GramianAngularField(image_size=28, method='difference')
X_gadf = gadf.fit_transform(X)
```

```
# Show the results for the first time series
axs = plt.subplots()
plt.subplot(211)
plt.imshow(X_gasf[0], cmap='rainbow', origin='lower')
plt.title("GASF", fontsize=16)
plt.subplot(212)
plt.imshow(X_gadf[0], cmap='rainbow', origin='lower')
plt.title("GADF", fontsize=16)

#plt.axes((left, bottom, width, height)
cax = plt.axes([0.7, 0.1, 0.02, 0.8])
plt.colorbar(cax = cax)
plt.suptitle('Gramian Angular Fields', y=0.98, fontsize=16)
plt.tight_layout()
plt.show()
```

Method 2 :Markov Transition Fields

MTF (Markov transition field)：把时间序列的值量化然后计算时序上的转移概率,

原文地址：Imaging Time-Series to Improve Classification and Imputation https://arxiv.org/pdf/1506.00327.pdf

中文解释：https://zhuanlan.zhihu.com/p/83130649

> 除了 GSAF 和 GSDF 之外，*《Imaging Time Series to Improve Classification and Imputation》*，*《Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks》*，*《Encoding Temporal Markov Dynamics in Graph for Time Series Visualization》*也提到了把时间序列转换成矩阵 Image 的算法 MTF。在 pyts 开源工具库里面，也提到了 MTF 算法的源码。

实现方式

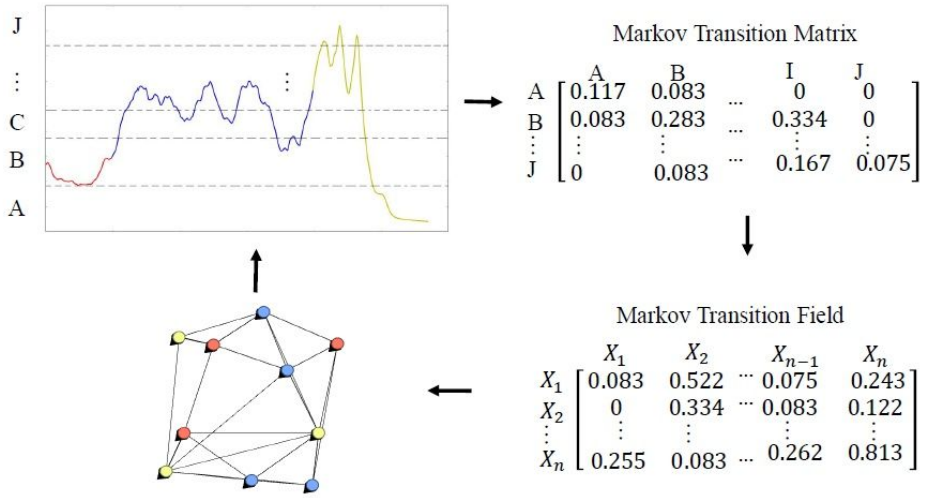


Figure 1: Illustration of the proposed encoding map of a Markov Transition Field. X is a sequence of the typical time series in the 'ECG' dataset. X is first discretized into Q quantile bins. Then we calculate its Markov Transition Matrix W and finally build its MTF M by Equation (2). We reduce the image size from 96×96 to 48×48 by averaging the values in each non-overlapping 2×2 patch.

假设有一时间序列$X = x_1, x_2, x_3, x_4 \dots x_n$

建立QxQ马尔可夫转移矩阵

我们把它们的**值域**分成Q个桶，那么每个x_i都可以被映射到一个相应的q_j上（j∈[1，Q]）。因此，我们通过沿时间轴以一阶马尔可夫链的方式**桶之间的迁移次数进行计数**来构造Q×Q加权邻接矩阵W（Thus we construct a Q×Q weighted adjacency matrix W by counting transitions among quantile bins in the manner of a first- order Markov chain along the time axis.）。$w_{i,j}$**表示在桶**j**中的元素被在桶**i**中的元素跟随的概率**（is given by the frequency with which a point in quantile qj is followed by a point in quantile qi.）。在通过$\sum_j w_{i,j} = 1$归一化（normalization），W变成马尔可夫转移矩阵。它对X的分布和时间步长ti的时间依赖性不敏感。但是，我们在W上的实验结果表明，摆脱时间依赖性会导致矩阵W中的信息丢失过多。

马尔可夫转移场MTF

为克服此缺点，我们在此基础上设想了马尔可夫转换场（MTF）M，这个M就是一个概率转移矩阵，定义如下：

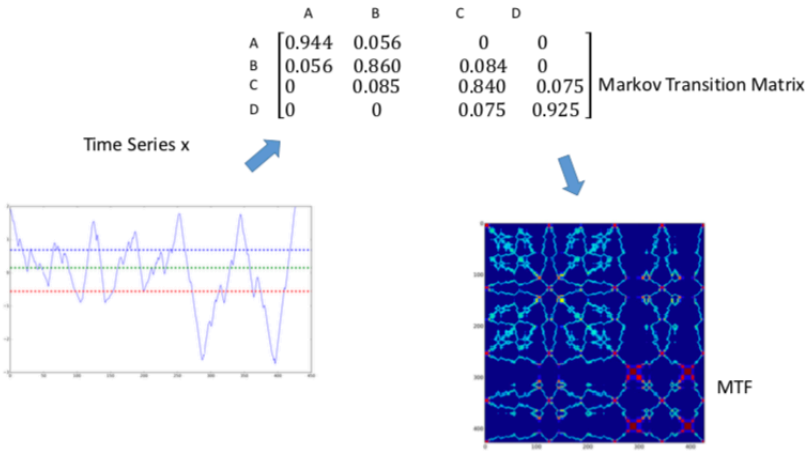
$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix}$$

其中$m_{i,j}$表示桶i中的元素迁移至桶j中的概率$P(q_i -> q_j)$，同样有$\sum_{1<=j<=Q} m_{i,j} = 1$，我们同样可以构造出一个Q×Q矩阵将时间序列可视化。

通过在每个像素M_{ij}处分配从时间戳i的桶到时间戳j的桶，MTF M实际上是对**时间序列的多跨度转换概率进行编码**。$m_{ij||i-j|=k}$表示**时间间隔为(|i-j|=k)k的点之间的转移概率。**例如，$m_{ij||i-j|=1}$显示出了沿着时间轴具有跳步的过渡过程(transition process along thetime axis with a skip step)。主对角线m_{ii}是在k = 0的特殊情况下，在时间戳i处捕获了从每个桶到其自身的概率（自转换概率）。

为了使图像尺寸易于管理和提高计算效率，我们通过使用**模糊内核**(blurring kernel)$\{\frac{1}{m}^2\}_{m*m}$对每个不重叠的m×m色块中的像素进行平均来减小MTF大小。 也就是说，我们将长度为m的每个子序列中的转移概率汇总在一起。 下图显示了将时间序列编码为MTF的过程。

如下图，Illustration of the proposed encoding map of Markov Transition Fields. X is a sequence of time-series in the 'ECG' dataset . X is first discretized into Q quantile bins（被横线划成了4个部分）. Then we calculate its Markov Transition Matrix W and finally build its MTF with eq. (8).



> 时间序列的降维方法有两种：
>
> 分段聚合（PAA）：使用局部平均等方法，把时间序列进行降维；
>
> 核变换（Kernel）：使用 Bivariate Gaussian 核或者均值核来把时间序列进行降维。

```
import matplotlib.pyplot as plt
from pyts.image import MarkovTransitionField
'''
读取时间序列的数据
怎么读取需要你自己写
X为ndarray类型数据
'''
# MTF transformation
mtf = MarkovTransitionField(image_size=24)
X_mtf = mtf.fit_transform(X)

# Show the image for the first time series
plt.figure(figsize=(5, 5))
plt.imshow(X_mtf[0], cmap='rainbow', origin='lower')
plt.title('Markov Transition Field', fontsize=18)
plt.colorbar(fraction=0.0457, pad=0.04)
plt.tight_layout()
plt.show()
```

未来展望

1、未来的重要工作将涉及开发递归神经网络以处理流数据。我们还对**不同的深度学习架构如何在GAF和MTF图像上perform**感兴趣。未来的另一项重要工作是学习GAF图像上具有更高级功能的深度生成模型。

2、未来的重要工作将涉及将我们的**方法应用于大量的数据**，并在一个**更复杂的参数空间中进行搜索**，以解决现实世界中的问题。另一个有趣的未来工作是**通过GAF和MTF图像建模时间序列**。 我们的目的是将学习的时间序列模型应用于回归/计算和异常检测任务。 为了将我们的方法扩展到流数据，我们假设设计了具有递归网络结构的在线学习方法。

Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images

Yang, C.-L.; Chen, Z.-X.; Yang, C.-Y. Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images. Sensors 2020, 20, 168.

文章地址：https://www.mdpi.com/1424-8220/20/1/168#cite

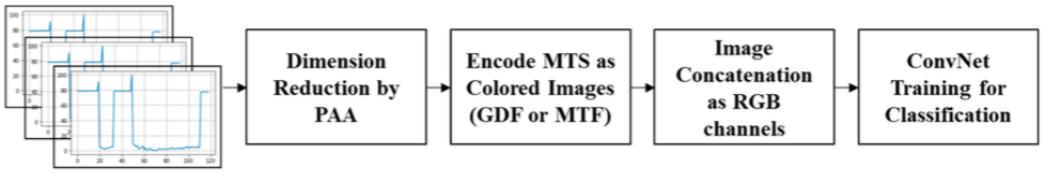
本文提出了一个使用多元时间序列传感器数据作为输入来执行传感器分类的框架。该框架将多元时间序列数据编码为二维彩色图像，并通过卷积神经网络（ConvNet）将图像连接为一个较大的图像进行分类。这项研究应用了三种变换方法将时间序列编码到图像中：格拉姆角求和场（GASF），格拉姆角差场（GADF）和马尔可夫变换场（MTF）。使用两个开放的多元数据集来评估使用不同转换方法，串联图像的顺序以及ConvNet体系结构的复杂性对分类准确性的影响。结果表明，转换方法的选择和串联顺序对预测结果没有明显影响。令人惊讶的是，ConvNet的简单结构足以用于分类，因为它在VGGNet的复杂结构上表现同样出色。将结果与其他分类方法进行了比较，发现该框架在分类准确性方面优于其他方法。

Contribution

- 这项工作旨在将用于多元时间序列（Multivariate Time Series ）分类的二维图像转换方法从单变量时间序列输入扩展到多元时间序列输入；
- 论文提出的创新图像串联可以将多元时间序列数据转变成多个颜色通道进行组合，作为ConvNet的输入；
- 论文提议的框架可通过使用相对简单的网络来提高MTS分类的准确性；
- 结果表明选择图像变换方法，并且图像串联的顺序对于分类准确性并不重要。

Methodology

本研究旨在为利用深度学习技术对多元时间序列数据进行分类提供一个框架。本研究首先应用MTF，GASF和GADF将多元时间序列数据转换为图像。 然后，将转换后的图像进行串联以供ConvNet处理，以识别图像中的特征以进行分类。 基本上，此框架包括四个步骤：（1）减少时间编码的维数；（2）图像编码（3）图像级联；以及（4）ConvNet分类模型训练。



使用分段聚合近似（PAA）进行降维

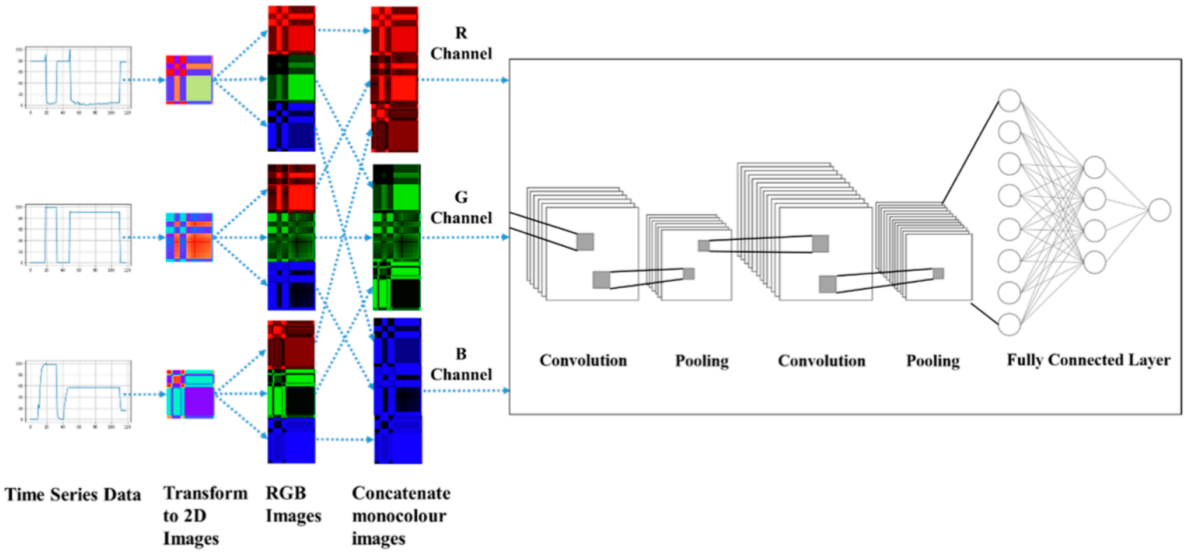
图像由像素组成，因此可以将其视为 n × n矩阵，其中n定义图像大小。当时间序列数据的长度为n时，任何一种转换方法的图像大小均为n × n。由于每批时间序列数据的长度可以变化，因此将原始数据直接转换为图像将导致图像大小不同。因此，为了获得ConvNet的相同大小的图像，在本研究中，应用分段聚合近似（PAA）方法对原始时间序列数据进行降维，然后再将时间序列数据转换为图像。请注意，在将时间序列传输到图像之前，应用PAA也是数据预处理的常规方法。

将时间序列数据编码为图像

在这项研究中，形成了一个包含多元时间序列数据MTS的3维矩阵。首先，使用GDF或MTF方法将时间序列数据编码为具有二维的彩色图像。由于图像可以是任何颜色，因此需要再增加一个尺寸来表示颜色。例如，可以用3个颜色通道分别用红色，绿色和蓝色（RGB）表示图像。然后，存在第一个维度中的3个元素。请注意，更多颜色可用于表示更多颜色通道。在这项工作中，只有RGB通道可以评估框架的概念。

图像与串联

MTS数据转换会生成多个图像（每个单变量时间序列一个图像）。这些图像必须在馈送ConvNet之前合并。本研究采用了Yang等人提出的级联方法。对于RGB图像聚合，首先将每个彩色图像分为三个单色图像：在这种情况下为红色，绿色和蓝色（RGB）。然后将这些单色图像连接在一起，形成更大的图像。图4说明了级联RGB图像的框架。请注意，如果将更多时间序列数据用作分类的输入，则将相应地生成更多2D图像。但是，在这种情况下，将仅构建三个RGB通道。基本上，此设计将保持相同数量的网络结构输入通道，这将有利于保持ConvNet网络结构的简单性。这种设计特别方便应用于异常检测等领域，在该领域中，可以通过各种传感器在边缘计算上处理时间序列数据，并且可以将图像文件上载为ConvNet的输入，而ConvNet可能位于不同的位置例如在云计算环境上。



卷积网络架构

在这项研究中，对于每个时间序列数据，二维变换图像的大小固定为 128 × 128像素。由于建议的级联方法的性质，如果存在*m*个时间序列，则ConvNet的输入图像的大小固定为（128 ×米） × 128每个单色通道。对于RGB图像，将分配三个通道。

为了评估ConvNet体系结构的复杂性是否会影响分类准确性，在本研究中，研究了两种ConvNet，分别称为简单ConvNet和VGG16。Simonyan和Zisserman提出的VGG16模型是2014年ImageNet大规模视觉识别竞赛（ILSVRC）的模型[20]。

对于简单的ConvNet，我们采用了Palm设计的非常流行的模型[34]。两个卷积层，内核大小为5 × 5，两个最大池化层 2 × 2像素窗口和步幅为2，建议使用一个全连接层。最大池化后，输入图像的高度和宽度变为一半。根据[19]中建议的设置，将学习率设置为0.0023，并将整流非线性作为激活函数应用于所有隐藏层。为了防止过度拟合的问题，根据[35]中的建议实施了早期停止方法。此方法还可以减少内存并减少计算时间。

由于VGGNet使用更多的层和较小的卷积滤波器来构建更深的网络结构深度，因此在这项工作中，我们将VGGNet视为一个较大的学习网络，有望对图像进行更准确的分类。本研究采用了典型的VGG16，它具有13个卷积层，内核大小为3 × 3，最多5个池化层 2 × 2像素窗口和3个完全连接的图层。学习率基于[20]设置为0.00023 。大多数可学习的参数都用于第一个完全连接的层。VGG16中可学习的参数数量为201,330,688，比简单的ConvNet（251,542）大800倍。显然，与典型的ConvNet相比，可以预期VGG16将花费更多的执行时间和内存。

采用数据集

The Wafer dataset was collected from six vacuum chamber sensors that monitored the manufacture of semiconductor microelectronics.

The ECG dataset in which exactly one heart beat exists per series was collected from two electrodes that recorded heartbeats as normal or abnormal.

Future Work

有进一步研究该模型的未来方向。首先，在这项工作中，仅使用一个ConvNet来训练数据。可以构建另一个框架，该框架为**每个时间序列数据使用并行的ConvNet**，并将它们连接到最后一层进行预测。值得评估的是，并行网络是否会提高准确性。其次，开发一种可以同时在**时间范围内保留动态和静态信息**，或者**滤除时间序列中不相关的噪声的变换方法**，可能有助于提高图像的特征区别性。第三，**检查单色多于RGB是否可以进一步改善分类**可能会很有趣。最后但并非最不重要的一点是，由于当前框架仅应用于二进制分类数据集中。