Examen - Procesos Gaussianos y Puesta en Marcha

MA6202 Laboratorio de Ciencia de Datos

Otoño 2020

1 Introducción

La siguiente evaluación corresponde al examen del curso de laboratorio de ciencia de datos. A modo de contexto, el conjunto de datos a trabajar consiste en el reporte estadístico oficial de calidad del aire en Beijing. Este conjunto esta compuesto por registros de frecuencia horaria en 11 estaciones de monitoreo ubicadas en distintos puntos de la capital, la distribución de las estaciones puede apreciar en la figura (1). Los datos a utilizar están disponibles en este link [1].

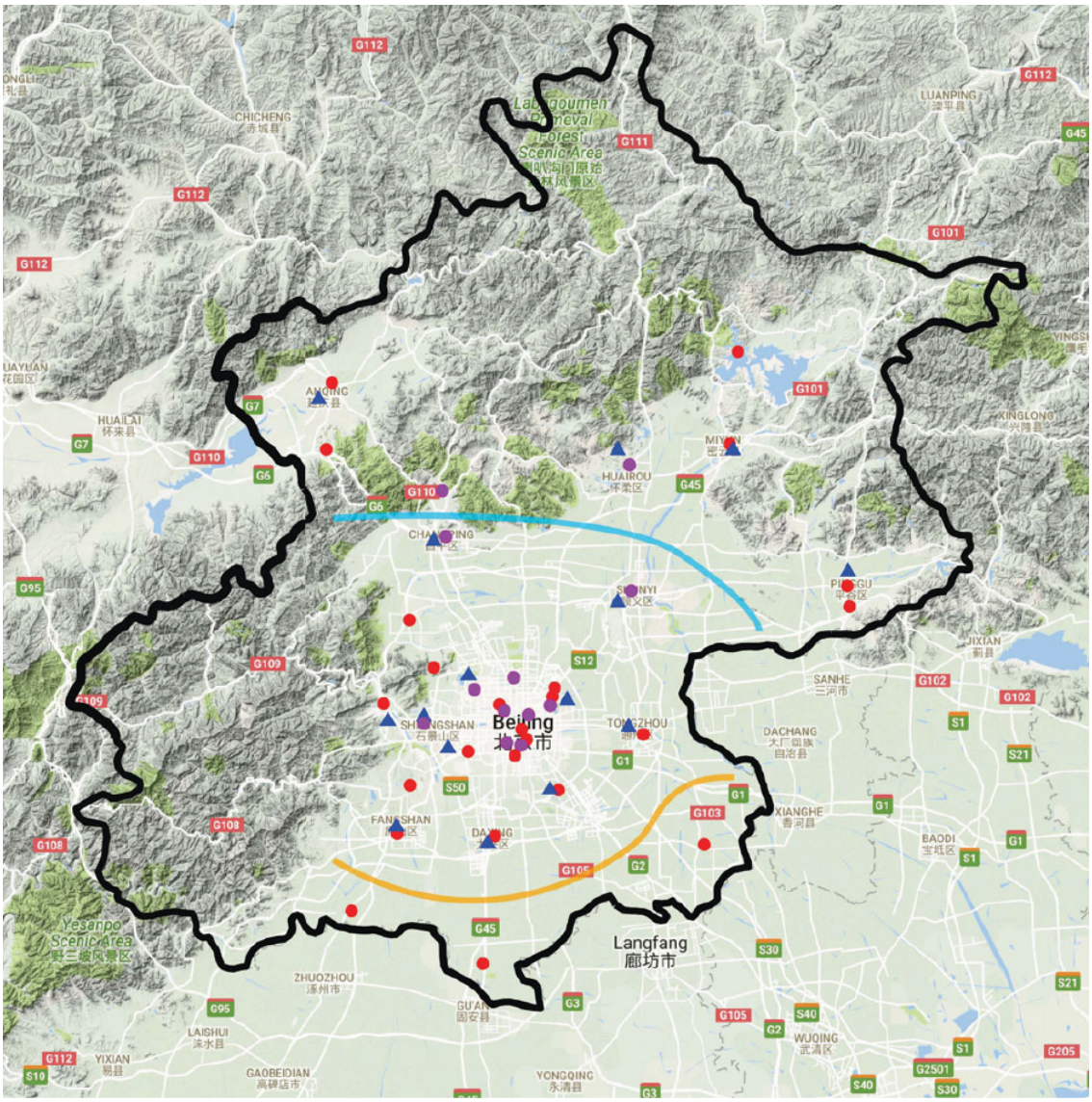

Figura 1: Estaciones de monitoreo en Beijing. los puntos violetas muestran su ubicación, los triángulos son estaciones meteorológicas y los puntos rojos son otras estaciones no incluidas en el conjunto de datos. Las lineas azul y naranjas dividen a la metrópolis en zona norte, centro y sur.

Se busca hacer un análisis de estos datos utilizando procesos gaussianos para finalmente generar una API con los modelos obtenidos. Esta evaluación es de carácter **individual**, se evaluará la presentación de sus resultados en dos modalidades. La primera por medio de un informe, cuyas condiciones de entrega son:

- La extensión máxima del informe es de 6 planas a las que puede añadir 2 para demostraciones.
- Debe adjuntar un repositorio `git` donde se incluya todo su código.
- A lo menos 1 `commit` por cada pregunta de la tarea
- Por lo menos 1 `merge` a través de su trabajo.
- Incluya un documento `jupyter notebook` llamado `examen.ipynb` en el cual se exponga todo el procedimiento realizado.
- Por último es necesario también entregar un archivo *.zip* con su modelo final.

La segunda opción de entrega es una presentación de sus resultados de manera remota a coordinar con el profesor. En esta modalidad no se requiere preparar un informe pero generar una presentación y mostrar sus resultados de investigación.

P1. Carga y Exploración

En la presente sección se realizan los pasos de carga y limpieza de datos que permitirán realizar las secciones posteriores con un `DataFrame` consolidado que presente los tipos de datos adecuados para la información contenida en sus columnas.

Exponga en el reporte todas las decisiones que llevó en esta sección, además de discutir acerca de los aspectos específicos señalados en cada pregunta.

1. Convierta la información de las columnas `'year'`, `'month'`, `'day'` y `'hour'` en formato `datetime64`, mediante `pandas.to_datetime`. Añada una columna al `DataFrame` que contenga, para cada fila, el número de horas a partir de las 0 hrs del primero de marzo de 2013 (fecha inicial del conjunto de datos).

De ahora en adelante se entendera por *columnas de contaminantes* a `'PM2.5'`, `'PM10'`, `'SO2'`, `'NO2'`, `'CO'` y `'O3'`.

2. Analice las correlaciones de presencia de valores faltantes para cada contaminantes en las diferentes estaciones. Genere visualizaciones y discuta lo observado.
3. En la columnas de contaminantes, rellene los valores faltantes mediante interpolación. Para ello utilice el método `pandas.DataFrame.interpolate`. Pruebe diferentes métodos de interpolación, seleccione uno y fundamente su elección. ¿Se puede justificar un esquema de llenado de valores faltantes con este método?
4. Analice las correlaciones para cada contaminantes en las diferentes estaciones. Genere visualizaciones, discuta lo observado y seleccione el contaminante que tenga el menor promedio de correlación entre estaciones. Para dicho contaminante, seleccione las tres estaciones que tengan el promedio de correlación más alto.
5. Mediante el función `statsmodels.tsa.seasonal.seasonal_decompose` descomponga en una tendencia, una señal periódica y su residuo; la señal del contaminante seleccionado en cada una de las 3 estaciones seleccionadas . Genere visualizaciones, discuta lo observado además de explicar el comportamiento de dicha función.

P2. Modelación

A continuación se busca modelar la interacción temporal del contaminante objetivo en las tres estaciones seleccionadas. Para ello se utilizan **procesos gaussianos**.

A modo de recuerdo, un proceso gaussiano consiste en una familia de variables aleatorias, tales que cualquier conjunto finito de estas presenta una distribución gaussiana conjunta. Para modelar problemas de regresión en

aprendizaje de máquinas, se utiliza este tipo de procesos como una **distribución prior** sobre el conjunto de funciones f aproximantes, esto se denota por:

$$f \sim GP(m(\cdot), k(\cdot,\cdot))$$

Donde $m(\cdot)$ y $k(\cdot,\cdot)$ corresponden a las funciones de media y covarianza respectivamente. Se denota el conjunto de entrenamiento por $D = \{(x_i, y_i) \mid i = 1, \ldots, N\}$ donde x_i corresponde a un punto en el tiempo para cierta estación de monitorieo e y_i pasa a ser el valor del contaminante medido por tal estación en aquel instante.

En general tiene sentido considerar que $f \sim GP(0, k(\cdot,\cdot))$, así para $f = [f(x_1), \ldots, f(x_n)]$ se tendrá que $f \sim GP(0, K(X,X))$, donde $K(X,X)_{ij} = k(x_i, x_j)$ es la matriz de Gramm asociada al kernel k. Al considerar que la distribución prior conjunta entre los puntos de entrenamiento f y los puntos a predecir f_* se puede obtener:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

Luego, al usar el teorema de bayes:

$$P(f, f_* \mid y) = \frac{P(y \mid f, f_*)\, P(f, f_*)}{P(y)} = \frac{P(y \mid f) P(f, f_*)}{P(y)}$$

Donde $P(y \mid f, f_*) = P(y \mid f)$, pues la verosimilitud se modela como independiente de f_* dado f. En el problema de regresión, esto se expresa según

$$y \mid f \sim N\left(f, \sigma_n^2 I_n\right)$$

Donde I_n es la matriz identidad de $N \times N$. Así, la distribución predictiva pasa a ser

$$P(f_* \mid y) = \int P(f, f_* \mid y)\, df = \frac{1}{P(y)} \int P(y \mid f) P(f, f_*)\, df$$

Como se trabaja con distribuciones normales, se tiene:

$$\begin{aligned} E[f_* \mid y] &= K(X_*, X)\, \Lambda^{-1} y \\ \mathrm{Cov}[f_* \mid y] &= K(X_*, X_*) - K(X_*, X)\, \Lambda^{-1} K(X, X_*) \end{aligned}$$

Donde f_* son los puntos a predecir o *puntos test*. Observe que los valores de media y covarianza predictivos determinan de manera única un proceso gaussiano como medida posterior sobre las funciones cuya distribución prior definimos inicialmente. Donde

$$\Lambda = K(X,X) + \sigma_n^2 I_N$$

Vale destacar que el calculo de Λ^{-1} representa un gran gasto computacional pues escala según $\mathcal{O}(n^3)$. Observe además que dada la función k, se tiene una expresión cerrada para el modelo predictivo, esto es una ventaja, pues permite obtener los hiperparámetros del modelo (asociados al kernel y al ruido de la verosimilitud). La obtención de dichas magnitudes se puede realizar por la maximización de la verosimilitud marginal. Esta función puede ser calculada al introduciendo valores latentes, en efecto, sea θ el conjunto de hiperparámetros a optimizar y $K_X(\theta)$ la matriz de Gramm cuyos parémtros son θ. La verosimilitud marginal tima la forma :

$$p(y \mid X, \theta) = \int P(y \mid f, X) p(f \mid X, \theta) df$$

Donde la distribución de observaciones $P(y \mid f, X)$ es condicionalmente independiente de la función latente f. Dado que se trabaja bajo una prior dada por un proceso gaussiano, se tiene que $f \mid X, \theta \sim N(0, K_X(\theta))$, que en términos de la log-verosimilitud queda expresado por:

$$\log P(y \mid X, \theta) = -\frac{1}{2} y' \left(K_X(\theta) + \sigma_n^2 I_N\right)^{-1} y - \frac{1}{2} \log \left|K_X(\theta) + \sigma_n^2 I_N\right| - \frac{N}{2} \log 2\pi$$

Tal expresión puede ser maximizada numéricamente con (por ejemplo) métodos de descenso de gradiente, así obtenemos $\theta^* = \arg\max_\theta \log p(y \mid X, \theta)$.

A continuación entrenará 3 procesos gaussianos independientes, uno para cada estación. La variable de respuesta corresponde al contaminante seleccionado previamente. Para ello:

1. Genere una agrupación del conjunto de datos, donde se tenga disponibilidad de la concentración diaria promedio del contaminante objetivo para cada estación.

2. Utilice la librería `gpytorch`, acá deberá generar una clase `ExactGP` que maximice la log verosimilitud marginal utilizando Adam como algoritmo de optimización.

3. Entrene su modelo sobre un conjunto de entrenamiento consistente en los dos primeros años del dataset generado con concentraciones diarias. Cada modelo entrenado debe ser capaz de predecir al menos 60 días luego del conjunto de entrenamiento.

4. En el modelo anterior, utilice una combinación de al menos dos kernels, indicando una distribución prior para cada uno de sus hiperarámetros. Explicite sus elecciones en base a un análisis sobre el conjunto de datos.

En la formulación anterior, los modelos entrenados son independientes entre si. Sin embargo, deben existir correlaciones entre sus series temporales pues corresponden a estaciones de monitoreo del mismo contaminante en una misma ciudad.

Teniendo en cuenta lo anterior, se puede generar un modelo que tome en cuenta las concentraciones diarias de cada estación y las correlaciones entre estación. Esto se puede hacer mediante un **proceso gaussiano multi-output**. La formulación de este tipo de modelos consiste en agrupar las observaciones para M canales (estaciones) en el vector $\mathbf{y} = (y_{11}, \ldots, y_{N1}, \ldots, y_{12}, \ldots, y_{N2}, \ldots, y_{1M}, \ldots, y_{NM})^T$, donde y_{nl} corresponde a la observación $n-$ esima del canal l, esta observación esta asociada al input x_n. Si se establece un proceso gaussiano como distribución prior sobre las funciones latentes $\{f_l\}$ asociadas a cada canal, es posible inducir correlaciones entre canal utilizando un kernel de la forma:

$$\langle f_l(\mathbf{x}), f_k(\mathbf{x}')\rangle = K^f_{lk} k^x(\mathbf{x}, \mathbf{x}') \quad y_{il} \sim \mathcal{N}\left(f_l(\mathbf{x}_i), \sigma_l^2\right)$$

Donde K^f es una matriz de Gramm que especifica similitudes entre canal, k^x es un kernel sobre los inputs y σ_l^2 es la varianza del ruido para el canal l. Asuma que el proceso gaussiano recientemente definido tiene media 0:

5. Deduzca una expresión para la media predictiva de tal proceso gaussiano multi-ouput.

6. Discuta sobre la posibilidad de aprender los hiperarámetros de este tipo de modelos.

7. Finalmente, proponga un kernel multi-output e implémentelo utilizando `gpytorch` sobre las tres estaciones seleccionadas. Discuta sobre su elección de hiperparámetros y distribuciones prior.

P3. Puesta en Marcha

En esta última sección se evaluará la puesta en marcha de su entorno de investigación. Para ello:

1. Genere un entorno de desarrollo basado en módulos. Monte tal entorno en un repositorio de control versiones.

2. Utilice una herramienta de manejo de dependencias y tests.

3. Genere una API con flask basándose en su construcción.

Referencias

[1] Zhang, S., Guo, B., Dong, A., He, J., Xu, Z. and Chen, S.X. (2017) Cautionary Tales on Air-Quality Improvement in Beijing. Proceedings of the Royal Society A, Volume 473, No. 2205, Pages 20170457.

[2] Edwin V. Bonilla and Kian Ming and A. Chai and Christopher K. I. Williams (2007),Multi-task Gaussian Process Prediction.