Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Profesor: Dr. Pablo Alvarado Moya
EL 5857 Aprendizaje Automático
I Semestre 2024

TEC | Tecnológico de Costa Rica

Entrega: **5 de marzo, 2024**

Tarea 2

Regresión polinomial y regresión ponderada localmente

I. Introducción

En esta tarea utilizaremos un escenario hipotético, en que queremos desarrollar un sensor económico para medir profundidad en el océano. Como datos sintéticos de referencia (*ground truth*) utilizaremos el perfil de la costa pacífica de Costa Rica tomado de este sitio.

El código base y los datos para esta tarea los encuentra en el repositorio de Github Classroom.

El sistema hipotético calcula datos de profundidad a partir de una red de sensores muy económicos, que sin embargo, producen muestras en posiciones medidas con ruido en lugares arbitrarios, y con valores ruidosos de profundidad.

Usted comparará los resultados de utilizar regresión polinomial y regresión ponderada localmente (LWR o LOWESS) para procurar reconstruir de las muestras un mapa de profundidad aceptable.

I.1. Objetivos

1. Evaluar estrategias de regresión no lineal y determinar las ventajas y desventajas de cada una.
2. Vectorizar procesos de cálculos complejos para aprovechar las capacidades de hardware moderno y las implementaciones que ofrecen los sistemas para álgebra lineal.
3. Aplicar los conceptos teóricos desarrollados en las lecciones en la solución de un problema de ingeniería.

II. Datos

Vamos a utilizar datos de profundidad tomados de este sitio, en particular una ventana en la costa pacífica de Costa Rica mostrada en la figura 1.

III. Código base

La función `gendata.m` construye datos simulados de la salida de los sensores, proveyendo pares de coordenadas (con ruido) y el valor de profundidad medido (con ruido también) en metros.

El archivo `showdata.m` genera con esa función, a manera de ejemplo, un set de datos con el 0,1 % de la rejilla real, y los muestra.

La misma función `gendata.m`, con la siguiente llamada particular:

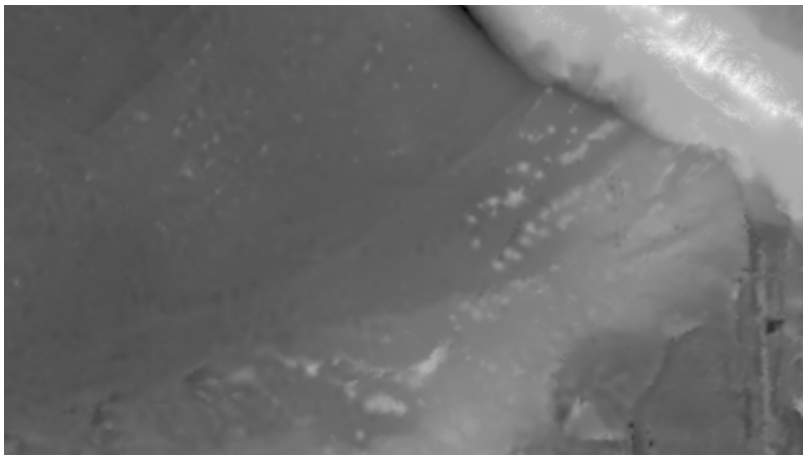

Figura 1: Mapa de profundidad de la costa pacífica de Costa Rica. Negro representa una profundad de aproximadamente -4000 msnm y blanco 4000 msnm.

```
[RX,ry]=gendata(1,0,0) ## Generate all data without noise
```

permite generar el 100 % de los datos sin errores y sin ruido, para ser utilizados como datos de referencia en el cálculo del error con cada método de regresión.

El archivo `regressall.m` es el archivo principal, desde donde se llaman las distintas implementaciones de técnicas de regresión. Usted deberá implementar las funciones correspondientes en los archivos `linreg.m`, `polyreg.m` y `lowess.m`, tal y como se describe en las siguientes secciones. El archivo `linreg_nointercept.m` provee una implementación de ejemplo de regresión lineal sin sesgo, lo que implica que el plano estimado en ese caso siempre pasa por el origen (algo así usualmente no tiene sentido en aplicaciones reales, pero sirve como punto de partida).

III.1. Pruebas unitarias

En el directorio `tests` hay varios archivos que ejecutan pruebas unitarias. Estos le permiten verificar el funcionamiento esperado de las funciones con casos sencillos de 1D y 2D en los tres regresores que debe implementar.

Para utilizarlos debe agregar ese directorio a la ruta de GNU/Octave, de modo que puedan ser encontrados. Si usted mantiene los dos directorios `code` y `tests`, entonces haga lo siguiente en la línea de comandos:

```
> cd code
> octave
> addpath("../")
> unit_tests
```

III.2. Consejos de implementación

Para extraer la norma de cada vector fila en una matriz se podría hacer uso de la función `vecnorm`, aunque esta función aplica la raíz cuadrada, que en realidad no es necesaria, porque en este caso se requiere la norma al cuadrado. Por tanto se recomienda utilizar `sumsq` o `dot`.

Algunos de los problemas se pueden plantear por medio del uso de tensores de tercer orden (arreglos de tres dimensiones), para lo cual necesitará el uso de la función `permute`, que le permite *transponer* matrices y vectores en filas, columnas y planos a su conveniencia.

Recuerde que en GNU/Octave puede usar `tic` y `toc` para medir el tiempo utilizado por su código.

IV. Tareas

En los puntos en donde corresponda, utilice "vectorización" para la implementación de sus funciones. Es decir, intente expresar hasta donde sea posible los cálculos en términos de operaciones entre matrices y vectores.

Recuerde que la implementación de productos con matrices diagonales usualmente se puede implementar aprovechando la difusión ("*broadcasting*") que hace GNU/Octave con el producto de matrices con vectores punto a punto.

Los tres tipos de regresión deben funcionar con datos unidimensionales y bidimensionales.

Para esta tarea habrá en la rúbrica un factor multiplicativo dependiente de la actividad individual de cada persona en el repositorio Git.

IV.1. Regresión lineal y polinomial

1. Corrija en el archivo `linreg.m` la regresión lineal para que la estimación pueda encontrar planos que no tengan que pasar por el origen. 3 pts

   La implementación que encontrará allí es solo "de relleno", y corresponde a la misma que está en `linreg_nointercept.m`.

2. Agrege en el archivo `regressall.m` código para mostrar en otra figura el plano resultante de la regresión que no pasa por el origen. 2 pts

   Note que allí en `regressall.m` ya hay código para mostrar la versión de regresión lineal que sí pasa por cero.

3. Corrija el archivo `polyreg.m` para que la estimación encuentre superficies polinomiales que no tengan que pasar por el origen. 10 pts

   La implementación que encontrará allí es solo "de relleno", y corresponde a la misma que está en `linreg_nointercept.m`.

   Observe que, debido a la potenciación y al rango de valores utilizados de entrada, es necesaria la normalización de los datos. Puede utilizar la clase `normalizer.m` presente en el código base.

   La función de GNU/Octave `bsxfun` puede serle de utilidad.

4. Agrege en el archivo `regressall.m` código para mostrar una figura con una superficie polinomial, con algun orden mayor que 5, resultantes de la regresión polynomial que no pasa por el origen. 2 pts

IV.2. Derivaciones teóricas

Considere un problema de regresión lineal en el que queremos "ponderar" de forma distinta cada dato de entrenamiento, tal y como vimos en clase con el método de regresión ponderada localmente

(LWR o LOWESS). Específicamente queremos minimizar la función de error

$$J(\underline{\boldsymbol{\theta}}) = \frac{1}{2}\sum_{i=1}^{m} w^{(i)} \left(\underline{\boldsymbol{\theta}}^T \underline{\boldsymbol{x}}^{(i)} - y^{(i)}\right)^2$$

que se puede reescribir como

$$J(\underline{\boldsymbol{\theta}}) = \frac{1}{2}(\boldsymbol{X}\underline{\boldsymbol{\theta}} - \underline{\boldsymbol{y}})^T \boldsymbol{W}(\boldsymbol{X}\underline{\boldsymbol{\theta}} - \underline{\boldsymbol{y}})$$

para una matriz diagonal $\boldsymbol{W}$ apropiada, donde $\boldsymbol{X}$ es la matriz de diseño y $\underline{\boldsymbol{y}}$ el vector de salidas, tal y como lo hemos venido trabajando en el curso.

1. Si todos los pesos $w^{(i)}$ son iguales a 1, entonces $\boldsymbol{W} = \boldsymbol{I}$ y, en clase vimos que la ecuación normal es para ese caso

   $$\boldsymbol{X}^T\boldsymbol{X}\underline{\boldsymbol{\theta}} = \boldsymbol{X}^T\underline{\boldsymbol{y}}$$

   y el valor de $\underline{\boldsymbol{\theta}}$ que minimiza $J(\underline{\boldsymbol{\theta}})$ está dado por $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\underline{\boldsymbol{y}}$.

   Encuentre el gradiente $\nabla_{\underline{\boldsymbol{\theta}}} J(\underline{\boldsymbol{\theta}})$ e iguálelo a cero para encontrar una versión generalizada de las ecuaciones normales en este contexto con ponderación. Encuentre una forma cerrada del valor de $\underline{\boldsymbol{\theta}}$ que minimiza a $J(\underline{\boldsymbol{\theta}})$, en función de $\boldsymbol{X}$, $\boldsymbol{W}$ y $\underline{\boldsymbol{y}}$. 10 pts

2. Suponga que tenemos un conjunto de entrenamiento $\{(\underline{\boldsymbol{x}}^{(i)}, y^{(i)}); i = 1 \ldots m\}$ de m ejemplos independientes, pero en el que los $y^{(i)}$ se observaron con varianzas distintas. Específicamente, suponga que

   $$p\left(y^{(i)}|\underline{\boldsymbol{x}}^{(i)}; \underline{\boldsymbol{\theta}}\right) = \frac{1}{\sqrt{2\pi}\sigma^{(i)}} \exp\left(-\frac{(y^{(i)} - \underline{\boldsymbol{\theta}}^T\underline{\boldsymbol{x}}^{(i)})^2}{2(\sigma^{(i)})^2}\right)$$

   o en otras palabras, $y^{(i)}$ tiene media $\underline{\boldsymbol{\theta}}^T\underline{\boldsymbol{x}}^{(i)}$ y varianza $(\sigma^{(i)})^2$, donde las $\sigma^{(i)}$ son fijas y conocidas para cada dato i.

   Demuestre que encontrar el estimado de máxima verosimilitud de $\underline{\boldsymbol{\theta}}$ se reduce a resolver un problema de regresión lineal ponderada.

   Indique claramente cómo se relaciona cada peso $w^{(i)}$ en términos de $\sigma^{(i)}$. 10 pts

   Advertencia: Note que $\sigma^{(i)}$ y $w^{(i)}$ varían con cada dato i, por lo que no se pueden "sacar" de sumatorias o productos por no ser un solo valor común a todos los datos.

IV.3. Regresión ponderada localmente de la superficie

1. Implemente la regresión lineal ponderada localmente en el archivo `lowess.m` en la función homónima. Allí encontrará una implementación de relleno a ser modificada (también por el momento calcula la regresión lineal sin sesgo).

   Use las ecuaciones normales que usted derivó en el punto IV.2.1. Para cada dato $\underline{\boldsymbol{x}}$ a predecir (es decir, cada dato de la rejilla), el i-ésimo dato del conjunto de entrenamiento $\underline{\boldsymbol{x}}^{(i)}$ tiene un peso de ponderación local dado por

   $$w^{(i)} = \exp\left(-\frac{\|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^{(i)}\|^2}{2\tau^2}\right)$$

con el parámetro de ancho de banda τ. 20 pts

Advertencia: El producto de una matriz diagonal por otra matriz o vector se puede hacer de forma mucho más eficiente en espacio de memoria y en cálculos utilizando *broadcasting* y el producto de Hadamard.

2. En el archivo `regressall.m`, en una figura aparte, grafique los datos resultantes de la regresión ponderada localmente sobre la rejilla brindada. 2 pts

IV.4. Medición del error

1. Implemente código para calcular el error entre los datos calculados con cualquiera de los métodos de regresión anteriores sobre la rejilla regular deseada, comparados con los datos de referencia. Especifique qué definición de error usa. 10 pts

   La función `sub2ind` puede serle de utilidad para extraer los valores de profundidad sobre la rejilla deseada de los datos de referencia. Esa función le permite convertir de manera eficiente pares de índices de fila/columna de una matriz a un índice lineal, con el que puede acceder el vector de etiquetas de profundidad.

   Para usarla, se parte del hecho de que los datos de referencia `RX` (ver línea 13 de `regressall.m`), contienen posiciones con coordenadas enteras consecutivas, por lo que el mayor valor encontrado en ellas corresponde al tamaño de esa matriz y el menor valor es exactamente 1. Por otro lado, puesto que `xx,yy` corresponde a otra rejilla regular, cuya coordenadas (línea 41 de `regressall.m`) son un subconjunto del total, entonces con

   ```
   gidx = sub2ind([maxy,maxx],yy(:),xx(:)); ## Find grid indices
   ```

   se obtienen primero los índices en `RX` correspondientes a las coordenadas "aplastadas" de la subrejilla `yy,xx`; y con

   ```
   rval = rz(gidx); ## Reference values at the indices
   ```

   se obtienen entonces los valores de referencia corespondientes a esas coordenadas `yy,xx`. Como usted debió obtener los valores de regresión para precisamente esos puntos `yy,xx` entonces ya tiene los valores de referencia contra qué comparar los resultados de los métodos de regresión.

2. Para el caso de regresión polinomial, evalúe el error para varios órdenes de polinomio. Grafique el error en función del orden del polinomio. 6 pts

3. Muestre la superficie reconstruida con el regresor de menor error en el punto anterior. 2 pts

4. Para el caso de regresión localmente ponderada, evalúe el error para varios valores de τ y elija finalmente el τ que produce el menor error, con respecto a los datos de referencia. Para ello grafique el error en función de τ. 6 pts

5. Muestre la superficie reconstruida para el τ de menor error del punto anterior. 2 pts

V. Puntos extra

Se otorgarán 6 %, 4 % y 2 % extra a aquellas personas que entreguen el código con las primeras tres posiciones de velocidad para la regresión polinomial y la ponderada localmente (cada una por aparte). Para ello se utilizará un programa que evalúa las funciones implementadas en `polyreg.m` y `lowess.m`, por lo que debe asegurar que se respete la interfaz solicitada de dichas funciones, o su implementación quedará automáticamente descalificada de esta oportunidad.

VI. Entregables

1. Repositorio Git en el Github Classroom con los aportes individuales constantes debidamente reportados. Se recomienda ya el uso de *branches*, usando una para cada pequeña tarea realizada, y haciendo los correspondientes *merge* a la rama principal, una vez que la tarea esté lista.

2. Archivo PDF con las demostraciones matemáticas de la primera parte, y las gráficas generadas para la segunda parte.

3. Archivos de GNU/Octave que usted realice para resolver los puntos anteriores.

4. Archivo README con instrucciones de cómo ejecutar su código.

VII. Notas

- Esta tarea es preferible resolverla en parejas. Envíe por favor lo antes posible al profesor cualquier cambio en los grupos realizado para esta tarea, respecto a los grupos de la tarea 1.

- Esta tarea requiere tiempo para comprender/derivar la matemática y hacer el programa, que por lo general sale en relativamente pocas líneas. Aproveche las oportunidades en clase y consulta para preguntar lo antes posible si surgen dudas.

- *Advertencia:* En las derivaciones teóricas tenga cuidado en no sacar de sumas o productos aquellos términos que dependen del índice de la suma o producto.

Toda tarea debe ser resultado del trabajo intelectual propio de la persona o personas que la entregan. Además de la literatura de referencia, solo puede utilizarse el material expresamente así indicado en la tarea, lo que incluye código brindado por el profesor o indicado en los enunciados a ser utilizado como base de la tarea. Expresamente quedan excluidos como material de referencia los trabajos entregados por otros estudiantes en el mismo semestre o semestres anteriores.

Nótese que esto no elimina la posibilidad de discutir estrategias de solución o ideas entre personas y grupos, lo cual es incluso recomendado, pero la generación concreta de cada solución, derivación o programa debe hacerse para cada entrega de forma independiente.

Para toda referencia de código o bibliografía externa deben respetarse los derechos de autor, indicando expresamente de dónde se tomó código, derivaciones, etc. Obsérvese que el código entregado por el profesor usualmente ya incluye encabezados con la autoría correspondiente. Si un estudiante agrega código a un archivo, debe agregar su nombre a los encabezados si la modificación es de más del 50 % del archivo, o indicar expresamente en el código, con comentarios claros, la autoría de las nuevas líneas de código, pues a la autora o al autor del archivo original no se le debe atribuir código que no es suyo.

Si se detecta código o deducciones teóricas iguales o muy cercanas a trabajos de otros estudiantes del mismo semestre o de semestres anteriores, se aplicará lo establecido por la reglamentación vigente, en particular el Artículo 75 del Reglamento de Régimen de Enseñanza y Aprendizaje.

Modificaciones de comentarios, cadenas alfanuméricas, nombres de variables, orden de estructuras independientes, y otras modificaciones menores de código se siguen considerando como clones de código, y las herramientas automatizadas de detección reportarán la similitud correspondiente.

Los estudiantes que provean a otros estudiantes del mismo o futuros semestres soluciones de sus tareas, también son sujetos a las sanciones especificadas en la reglamentación institucional. Por lo tanto, se advierte no poner a disposición soluciones de las tareas a otros estudiantes.