

Tarea 2: Regresión polinomial con descenso de gradiente

Para los datos del ejemplo en clase con precios de casas en Escazú, deseamos ahora explorar modelos polinomiales, utilizando para ello el diseño apropiado de vectores de características, tal y como se revisó en la lección 7.

El archivo de datos `escazu40.dat` así como las plantillas base de las funciones a implementar los encuentra en la carpeta **Documentos/Tareas/tarea2** en el **tecDigital**.

Para la tarea debe utilizar las plantillas entregadas, pues el profesor para evaluar utilizará otro programa de prueba que ejecutará las funciones con determinados puntos de inicio, métodos y otros hiperparámetros, para corroborar el funcionamiento correcto.

Puede revisar y utilizar como base el código brindado como ejemplo en la lección 2, en particular `batch_grad_descent.m`, `stoch_grad_descent.m` y `normalizer.m`, así como los ejemplos de la lección 7. Pueden agregarse funciones, clases o módulos propios si así se requiriera.

1. Construya su función de hipótesis polinomial $h_{\underline{\theta}}(x)$:

$$h_{\underline{\theta}} = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^n$$

en el archivo `hypothesis.m` siguiendo la interfaz allí indicada.

Además, implemente la función de error $J(\underline{\theta})$ en el archivo `loss.m` para la hipótesis polinomial así como el gradiente de dicha función en el archivo `gradloss.m`. Estas últimas dos funciones reciben una matriz de diseño \mathbf{X} con varios datos en sus filas y el valor deseado de salida \mathbf{y} para cada dato, tal y como se describe en la documentación de `descentpoly.m`.

Observe que el orden del polinomio se puede calcular a partir de la dimensión de $\underline{\theta}$.

2. En `descentpoly.m` usted debe implementar los métodos de optimización de descenso de gradiente (“batch”), descenso estocástico de gradiente (“stochastic”), y este último con “momentum”, RMSProp (“rmsprop”), o ADAM (“adam”), lo que se selecciona con el parámetro opcional “method”, que por defecto tiene valor “batch”. Todos los métodos excepto el “batch” usan minilotes con reemplazo.
3. Para el caso particular de aproximaciones cuadráticas, usted tiene $\underline{\theta} = [\theta_0 \ \theta_1 \ \theta_2]$. Muestre la trayectoria de valores $\underline{\theta}$ para algún caso que usted considere interesante, de forma similar a lo ilustrado en la figura 1.

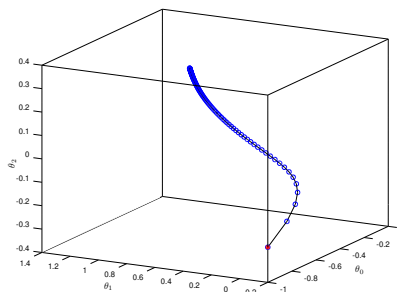


Figura 1: Trayectoria de minimización en el espacio paramétrico, iniciando en $[-1; -0.2; -0.3]^T$.

4. Muestre la evolución de la hipótesis en ese mismo caso hasta llegar al mínimo en el espacio de entrada, tal y como se ilustra en la figura 2.

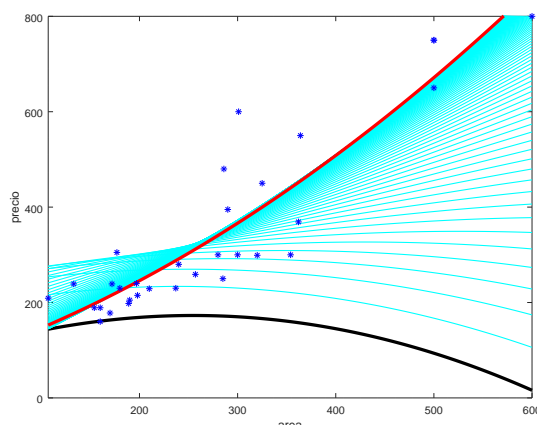


Figura 2: Espacio de entrada. Los puntos de entrenamiento están marcados con *. En negro se ilustra la curva correspondiente al punto inicial, y en cian cada curva intermedia hacia la solución óptima final, mostrada en rojo.

5. Muestre para varias tasas de aprendizaje α las curvas de evolución del error $J(\underline{\theta})$, de forma similar a lo ilustrado en la figura 3.

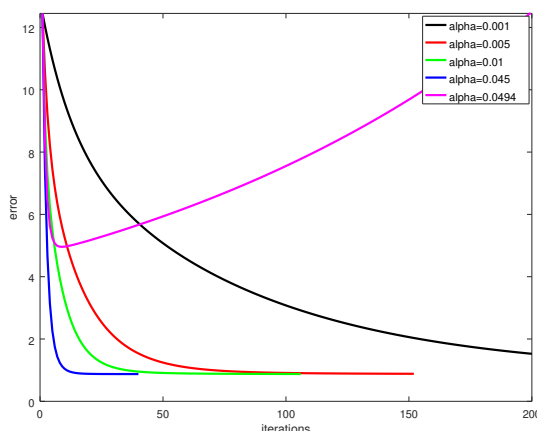


Figura 3: Evolución del error $J(\underline{\theta})$ en función del número de iteración, para varios valores de la tasa de aprendizaje.

6. Repita los puntos 3. al 5. con los otros métodos implementados. Observe que para cada caso solo debería ser necesario cambiar una cadena alfanumérica con el método correspondiente.
7. Genere una figura para cada método (“batch”, “stochastic”, etc.) en la que se superpongan las aproximaciones finales para regresiones con al menos 4 órdenes distintos de polinomios (lineal, cuadrática, cúbica, etc.), de forma similar a lo mostrado en la figura 4.

Entregable: archivos de GNU/Octave, y README con instrucciones de cómo ejecutar el código. Deben mostrarse las trayectorias en el espacio paramétrico, la evolución de las curvas aprendidas (precio(area)) y las curvas de evolución del error en los dos casos estocástico y por lotes.

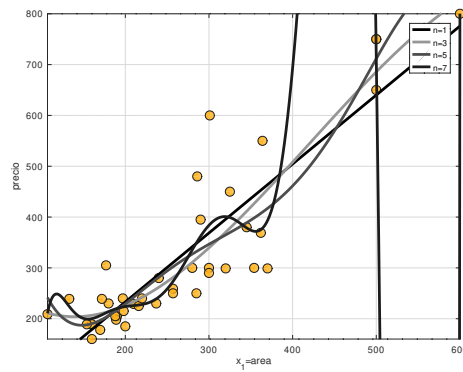


Figura 4: Ejemplo de figuras obtenidas para el método “batch”.

Esta tarea se realiza en parejas o individual. Informar al profesor si el grupo de trabajo en esta tarea cambió respecto al de la tarea 1.

Solo lo que sea subido al tecDigital será calificado. Entregas por correo serán ignoradas, por lo que asegúrese de haber enviado correcciones a su grupo (aún si es individual) a tiempo.

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.