

Proyecto de Optimización I

Un método robusto basado en LOVO para la solución de problemas con mínimos cuadrados.

Equipo: Juan Luis Baldelomar Cabrera y Andrea Quintanilla Carranza

Marzo del 2021

1. Introducción

Los puntos atípicos son uno de los grandes problemas que se presentan en el análisis de datos. Existen diversas definiciones de puntos atípicos, intuitivamente, son aquellos que se encuentran a una distancia considerablemente mayor que la distancia que existe entre los puntos restantes.

La presencia de este tipo de datos en una muestra surge por diversos factores tales como el error humano al realizar un experimento, el error del aparato utilizado para medir los resultados, o el dato obtenido en cuestión tiene una probabilidad realmente baja de ocurrir.

Representan un problema en diversas aplicaciones pues son capaces de sesgar los valores descriptivos de una muestra como lo es la media y la varianza y, por lo tanto, vulnerar distintos algoritmos. Por ello han surgido diversas técnicas para detectarlos y lidiar con ellos. Algunos ejemplos prácticos en donde se necesitan detectar datos atípicos son: detección de fraudes, aplicaciones para préstamos, detección de intrusiones a sistemas operativos, monitoreo de desempeño de computadoras, análisis de imágenes satelitales, detección de estructuras moleculares nuevas en farmacéutica, entre otros. [HA04]

En este trabajo nos interesan los problemas que provocan cuando se busca ajustar modelos a la muestra. El método de mínimos cuadrados es eficiente y preciso en ausencia de datos atípicos y de ruido considerable. Sin embargo, como ocurre a menudo, cuando el conjunto de datos contiene puntos atípicos se necesitan métodos más robustos para alcanzar la misma precisión, ya que incluso en presencia de un sólo dato atípico los resultados del ajuste pueden ser muy deficientes [Rou84].

En este trabajo se buscó replicar el artículo *A robust method based on LOVO functions for solving least squares problems*, [CLS21] pues éste propone un esquema basado en Low Order-Value Optimization (LOVO) que ha demostrado a través de resultados numéricos y comparaciones con otros algoritmos ser capaz de detectar e ignorar puntos atípicos utilizando pocos parámetros y llegando a mejores ajustes.

El esquema LOVO se define de la siguiente manera. Consideremos las funciones residuales $R_i : \mathbb{R}^n \rightarrow \mathbb{R}$ para $i \in 1, \dots, r$. Para un valor fijo de $x \in \mathbb{R}^n$ podemos ordenar estas funciones de manera que

$$R_{i_1(x)}(x) \leq R_{i_2(x)}(x) \leq \dots \leq R_{i_r(x)}(x)$$

en donde el elemento $i_k(x)$ es el k -ésimo elemento más pequeño del conjunto $\{R_i(x) | i = 1, 2, \dots, r\}$. Ahora, si tomamos un entero p tal que $0 < p \leq r$, la función LOVO se define como

$$S_p(x) = \sum_{k=1}^p R_{i_k(x)}(x) \quad (1)$$

y el problema LOVO se traduce a minimizar esta función, de manera que deseamos encontrar

$$\min S_p(x). \quad (2)$$

Notemos que este problema es una generalización del problema de mínimos cuadrados si tomamos $R_i(x) = \frac{1}{2}F_i(x)^2$ y $F_i(x) = y_i - \phi(x, t_i)$ en donde $\phi(x, t_i)$ es el modelo buscado, x son los parámetros, t_i son las observaciones y y_i es el valor esperado. Observemos que tomando $p = r$, podemos ver que, en efecto, el problema a resolver es el problema de mínimos cuadrados. De esta manera obtenemos el problema LOVO particular:

$$\min S_p(x) = \min \sum_{k=1}^p \frac{1}{2} \left(F_{i_k(x)}(x) \right)^2 \quad (3)$$

Los puntos t_i son proporcionados exactamente, de manera que se tiene un modelo de regresión fijo. Notemos que si nos permitimos variar el número p , podríamos conseguir un modelo $\phi(x, t)$ que no se base en las peores $r - p$ observaciones de nuestra muestra al ajustarlo. Así, el número p es un parámetro que indica el número de puntos de confianza.

Podemos notar una desventaja en el contexto de este problema, y surge ante la necesidad de tener que escoger un valor para p . Es decir, se debe conocer la posible cantidad de puntos atípicos en la muestra, algo que como es de esperar no siempre es posible y menos cuando se está trabajando con datos de los cuales se tiene poco o ningún conocimiento. Este trabajo se centra en replicar un acercamiento para optimizar este tipo de problemas con un esquema de Levenberg-Marquardt modificado y un sistema de votación para resolver la problemática que surge al tener que escoger el valor de p .

Los algoritmos de optimización de Levenberg-Marquardt son un caso particular de los esquemas de optimización de región de confianza. Para el caso de funciones con una estructura como la de mínimos cuadrados, el modelo que se utiliza es un modelo lineal. Este enfoque elimina la necesidad de información de segundo orden. El esquema a utilizar será adaptado para lidiar con problemas LOVO justamente con el objetivo de evitar la necesidad de información de segundo orden. Además, el sistema de votación utilizado para estimar la cantidad de puntos atípicos consiste, esencialmente, en ejecutar el algoritmo para diversos valores de p y seleccionar la solución que se ajuste mejor al modelo a través de algunos criterios que serán descritos posteriormente.

2. Estado del arte

Existen diversos métodos de ajuste robusto, como el RANSAC (Random Sample Consensus) y el LMS (Least Median of Squares Regression), contra los que se comparó la propuesta que revisamos aquí. El primero tiende a identificar correctamente distintos tipos de puntos atípicos, sin embargo, sólo lo logra si el porcentaje de dichos puntos es bajo, es decir, cuando ronda el 5% [LWP05]. Además, si el modelo tiene varios parámetros el costo computacional se dispara pues crece exponencialmente en función de ellos. Existen variantes del RANSAC que buscan contrarrestar ese costo, pero si los puntos no atípicos tienen un ruido considerable, el modelo resultante se puede desviar completamente del deseado [FP08]. En cambio, el modelo de votación del método estudiado en este trabajo no se ve afectado al aumentar el número de parámetros del modelo.

Por otro lado, el LMS, que se basa en minimizar la mediana de los errores -en lugar de la suma de los errores que se hace en el método de mínimos cuadrados usual-, puede soportar una proporción de datos contaminados muy alta (de hecho, el máximo soportable) y lograr un buen ajuste. Sin embargo, es poco eficiente pues converge como $n^{-1/3}$ con n el número de observaciones, aunque existen variantes de él para lidiar con ello. [Rou84]

A continuación mencionamos otros métodos contra los que no se comparó el algoritmo que estudiamos, pero son de interés en el área y ayudan a ilustrar la gran variedad de técnicas que han nacido por la importancia y ubicuidad de los puntos atípicos.

Algunos métodos en lugar de utilizar la norma l_2 para medir el error, como se hace en el método de mínimos cuadrados, utilizan la norma l_1 pues esta última lidia correctamente con el problema de las imágenes de puntos atípicos. Sin embargo, bajo esos esquemas es necesario diseñar algoritmos específicos para cada modelo. Además de que suelen ser costosos computacionalmente y no lidian bien con el leverage, o puntos atípicos en el dominio de la muestra. [ASW04]

Otros métodos robustos enumeran todas las posibles soluciones y reconocen con mucha certeza al total de puntos atípicos. Pero, por lo mismo, su complejidad es combinatoria y el costo computacional crece fuertemente si crece el tamaño de la muestra y los parámetros del modelo. [ASW93]

El esquema que estudiamos es similar al propuesto en [AMY09] donde se propone un LOVO robusto utilizando regiones de confianza e información de segundo orden. Sin embargo, como se mencionó en la introducción, en el enfoque del presente trabajo se evita el uso de información de segundo orden gracias a la incorporación del método de Levenberg-Marquardt.

3. Desarrollo

Como mencionamos anteriormente, uno de los objetivos del problema es minimizar la función

$$\min S_p(x) = \min \sum_{k=1}^p \frac{1}{2} \left(F_{ik(x)}(x) \right)^2$$

para un valor fijo de p , el segundo objetivo es encontrar un p óptimo. Para resolver el primer problema es útil considerar $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_q\}$, el conjunto formado por todos los subconjuntos de tamaño p de $\{1, 2, \dots, r\}$. Donde r representa la cantidad total de puntos en la muestra. De esta manera, para cada \mathcal{C}_i y para cada posible parámetro $x \in \mathbb{R}^n$, podemos definir:

$$f_i(x) = \sum_{k \in \mathcal{C}_i} R_k(x) \tag{4}$$

y

$$f_{\min}(x) = \min\{f_i(x), i = 1, \dots, q\} \tag{5}$$

Es decir, $f_{\min}(x)$ es el residuo mínimo que se alcanza para los parámetros x al variar entre conjuntos de p observaciones. De esta última ecuación podemos ver que $S_p = f_{\min}$. Podemos asegurar que si todas las f_i son continuas, entonces S_p es continua, sin embargo no hay un resultado análogo en el caso en el que sean todas las f_i sean diferenciables ya que f_{\min} no necesariamente es una función suave.

Para facilitar las siguientes observaciones, definimos como $F_{C_i}(x) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ como el mapeo que lleva a x al vector $F_{C_i}(x) = [F_{i_1}(x), \dots, F_{i_p}(x)]$ con $i_k \in C_i$, en algún orden fijo. Supondremos que las F_j son continuamente diferenciables y denotaremos como $J_{C_i}(x)$ al Jacobiano de $F_{C_i}(x)$.

Definamos ahora el conjunto de índices mínimos de la siguiente manera

$$I_{\min}(x) = \{i \in \{1, \dots, q\} \mid f_{\min}(x) = f_i(x)\}$$

Esta definición es útil para describir el algoritmo, sin embargo, a nivel de implementación el C_i asociado al $i \in I_{\min}(x)$, se consigue haciendo un ordenamiento de las funciones residuales $R_i(x)$ y seleccionando los primeros p resultados.

Buscamos entonces minimizar $f_{\min}(x)$, en la práctica se minimiza $f_i(x)$ para algún $i \in I_{\min}(x)$. Fijado un índice $i \in I_{\min}(x_k)$, el método iterativo de optimización propuesto, calcula la dirección de la actualización $x_{k+1} = x_k + d_k$ de la misma manera que en algoritmo de Levenberg-Marquardt, es decir, se aproxima el hessiano de la siguiente manera:

$$\nabla^2 f_i(x) = J_{C_i}(x)^T J_{C_i}(x) + S(x) \approx J(x)^T J(x) + \gamma_k I$$

donde $\gamma_k \in \mathbb{R}^+$ es conocido como parámetro de amortiguamiento, éste sirve para aproximar a la secante $S(x)$. Así, la dirección de d_k se obtiene resolviendo el siguiente sistema de ecuaciones:

$$\left(J_{C_i}(x_k)^T J_{C_i}(x_k) + \gamma_k I \right) d_k = -\nabla f_i(x_k) \quad (6)$$

ya que la matriz es simétrica, éste se resolvió por medio de la factorización de Cholesky. Para asegurar suficiente descenso, dado que se está trabajando con un esquema de región confianza, se debe calcular la medida:

$$\rho_{k,i} = \frac{f_{\min}(x_k) - f_{\min}(x_k + d_k)}{m_{k,i}(0) - m_{k,i}(d_k)}$$

donde:

$$\min_{d \in \mathbb{R}^n} m_{k,i}(d) = \frac{1}{2} \|F_{C_i} + J_{C_i}(x_k) d\|_2^2 + \frac{\gamma_k}{2} \|d\|_2^2$$

y compararla con cierta tolerancia. Pero en el artículo estudiado concluyeron que para los ejemplos revisados, era suficiente con asegurar decrecimiento, es decir, la dirección se acepta si:

$$f_i(x_k + d_k) < f_i(x_k)$$

El algoritmo de optimización resultante se muestra a continuación:

Algoritmo 1: LM-LOVO: Algoritmo de optimización

Entrada: $x_0 \in \mathbb{R}^n$; λ_{\min} ; $\epsilon > 0$; $\lambda_0 \in \mathbb{R}_+$; $\bar{\lambda} > 1$; $\mu \in (0, 1)$ y $p \in \mathbb{N}$

Salida : x_k

Inicializa $k = 0$

Fija $i_k \in I_{\min}(x_k)$

mientras $\|\nabla f_{i_k}(x_k)\| < \epsilon$ **hacer**

$\lambda = \lambda_k$

haz

$\lambda = \lambda \bar{\lambda}$

$\gamma_k = \lambda \|\nabla f_{i_k}(x_k)\|^2$

Calcula d_k resolviendo 6

mientras $f_i(x_k + d_k) > f_i(x_k)$;

$\lambda_{k+1} = \max\{\lambda_{\min}, \lambda/\bar{\lambda}\}$

$x_{k+1} = x_k + d_k$

$k = k + 1$

fin

devolver x_k

3.1. Sistema de Votación

Hasta ahora hemos visto como minimizar nuestra función de costo para un valor fijo de p . Podemos ver que esto implica que se conoce la cantidad de puntos confiables en un conjunto de datos, lo cual no siempre es cierto. Por lo tanto, el trabajo propone un sistema de votación que nos permite determinar la cantidad de puntos confiables para detectar los posibles outliers. Para esto el algoritmo solicita una un rango $\{p_{\min}, \dots, p_{\max}\}$, con $0 \leq p_{\min} < p_{\max}$, de la cantidad de puntos que se consideran confiables en el conjunto de datos. Luego ejecutamos el algoritmo de optimización propuesto para $p = p_{\min} \dots, p_{\max}$ y almacenamos las soluciones en un vector.

Luego de haber almacenado estas soluciones se construirá una matriz de similitud M entre cada una de las soluciones x_i^* y x_j^* en donde en este caso utilizamos como medida de similitud la norma euclidiana de la diferencia de dos vectores de manera que

$$M_{i,j} = \|x_i^* - x_j^*\|^2.$$

Antes de construir esta matriz se deben preprocesar estas soluciones. Se debe descartar aquellas soluciones con valor q para el parámetro p para las cuales el método lovo no logro converger. Esto se hace haciendo que $M_{q,i} = M_{i,q} = \infty$. También se debe descartar aquellas que definitivamente no son un mínimo global del problema que buscan resolver. Para esto consideremos que si tenemos dos valores para el parámetro p con valores de q_1 y q_2 con $q_1 < q_2$ en donde x_{q_1} y x_{q_2} son los valores que minimizan la función para el valor correspondiente de p , entonces es de esperar que $S_{q_1}(x_{q_1}) < S_{q_2}(x_{q_2})$ debido a que q_2 considera una mayor cantidad de puntos y por lo tanto se suma más funciones residuales. Entonces si $S_{q_1}(x_{q_1}) \geq S_{q_2}(x_{q_2})$ podemos eliminar este punto como posible solución. Para esto asignamos en la matriz de similitud $M_{q_1,i} = M_{i,q_1} = \infty$, para todo $i = p_{\min}, \dots, p_{\max}$.

En la fase donde se preprocesan las soluciones también se trata de descartar el valor obtenido para p_{\max} . Para esto se escoge el valor q tal que $S_q(x_q)$ tiene el menor valor de los elementos que pasaron el primer criterio de preprocesamiento descrito en el párrafo anterior. Luego debemos eliminar $x_{p_{\max}}^*$

si $S_q(x_q^*) < S_{p_{\text{máx}}}(x_{p_{\text{máx}}}^*)$ y el número de puntos observados (t_i, y_i) que cumplen

$$|y_i - \phi(x_q^*, t_i)| < |y_i - \phi(x_{p_{\text{máx}}}^*, t_i)|$$

para $i = 1, 2, \dots, r$ es mayor o igual a $r/2$.

Por último, el sistema de votación compara que tan parecidas son las soluciones obtenidas a través de su matriz de similitud. Con base en ella se vota por la solución que sea más parecida a las otras y para esto se requiere de un parámetro ϵ . Este puede ser proporcionado por el usuario, pero se ha observado que la siguiente elección da buenos resultados en la práctica

$$\epsilon = \text{mín}(M) + \text{avg}(M)/(1 + \sqrt{p_{\text{máx}}}).$$

Es importante notar que en caso de que el algoritmo no converja, se devolverá el modelo ajustado a todos los puntos hasta $p_{\text{máx}}$. El algoritmo es el siguiente

Algoritmo 2: Sistema de Votacion

Entrada: $x_0 \in \mathbb{R}^n$; $\epsilon > 0$, $0 \leq p_{\text{mín}} < p_{\text{máx}}$

Salida : estimación de x_k .

Definir $C \in \mathbb{R}^s = 0$ con $s = p_{\text{máx}} - p_{\text{mín}} + 1$

Calcular x_p^* a través del algoritmo 1 con p fijo para todo $p \in \{p_{\text{mín}}, \dots, p_{\text{máx}}\}$

Preprocesar Soluciones

Construir Matriz de Similaridad M

para $p = p_{\text{mín}}, \dots, p_{\text{máx}}$ **hacer**

$k = 0$

para $q = p_{\text{mín}}, \dots, p_{\text{máx}}$ **hacer**

si $M_{pq} < \epsilon$ **entonces**

$k = k + 1$

fin

fin

$C[p] = k$

fin

devolver x_p^* tal que $p = \arg \max_{q=p_{\text{mín}}, \dots, p_{\text{máx}}} \{C_q\}$

4. Resultados

A continuación describiremos los resultados obtenidos. Las gráficas se pueden observar en la sección de anexos. Es importante resaltar que algunos de los resultados a replicar causaron algunos problemas ya que en el paper original omiten algunas consideraciones y trucos de implementación que ayudar a dar mayor estabilidad numérica al proyecto. Estas consideraciones se pueden observar en el código del repositorio original del proyecto. Por ejemplo, la matriz a descomponer con la factorización de Cholesky contiene valores infinitos en ciertas iteraciones, para lidiar con ello consideran como dirección de descenso al gradiente y redefinen al x_k generando sus entradas con una distribución normal estándar.

Para la generación de datos utilizamos cuatro variantes propuestas por los autores de este trabajo que generan un conjunto de datos en base a un modelo que es enviado como parámetro y la cantidad de datos que se desea. Los modelos utilizados fueron los siguientes:

- Modelo lineal: $\phi(x, t) = x_1 t + x_2$
- Modelo cúbico: $\phi(x, t) = x_1 t^3 + x_2 t^2 + x_3 t + x_4$

- Modelo exponencial: $\phi(x, t) = x_1 + x_2 \exp(-x_3 t)$
- Modelo logístico: $\phi(x, t) = x_1 + \frac{x_2}{1 + \exp(-x_3 t + x_4)}$

Se generan pequeñas perturbaciones provenientes de una distribución normal para simular el ruido en los datos, y un porcentaje de los datos son convertidos en puntos atípicos agregando ruido que proviene de otra distribución con mayor varianza para que estos datos se alejen más del verdadero modelo. Específicamente, los datos etiquetados como no outliers o trusted points son de la forma: $y_i = \phi(x^*, t_i) + \xi_i$, mientras que los outliers son de la forma $y_i = \phi(x^*, t_i) + 7s\xi'_i|\xi_i|$, donde $\xi_i \sim \mathcal{N}(0, 200)$, $\xi'_i \sim \mathcal{U}(1, 2)$ y t_i son puntos equiespaciados en el intervalo $[1, 31]$. El valor absoluto $|\xi_i|$ aplicado para generar los outliers se utilizó puesto que en los detalles sobre los resultados se menciona en el paper original que se desea que los outliers estén sólo de un lado de la curva, sin embargo, en la teoría no hay una supuesto sobre ello.

Los puntos utilizados como x^* se muestran en la siguiente tabla y como x_0 se utilizó al vector cero en todos los casos y se utilizaron muestras de tamaño 100.

Modelo	x^*
Lineal	$(-200, 1000)$
Cúbico	$(0.5, -20, 300, 1000)$
Exponencial	$(5000, 4000, 0.2)$
Logístico	$(6000, -5000, -0.2, -3.7)$

En las gráficas que mostraremos como resultados observaremos la siguiente estructura. Los puntos de color verde son los puntos que fueron clasificados de manera correcta por el modelo final obtenido (ya sean outliers o trusted points). Los puntos de color azul son los que fueron clasificados de manera incorrecta. Los puntos con forma de triángulo son los verdaderos outliers del modelo, etiquetados según la forma en la que fueron generados, y los restantes, con un círculo.

En todos los casos tenemos un conjunto de datos de 100 puntos en donde 10 de ellos tienen la categoría de punto atípico. También en todos los resultados se muestran modelos similares a los esperados, es decir, los parámetros óptimos del modelo no muestran tener mucha influencia de outliers fuertemente distantes.

También en todas las gráficas presentadas los outliers visualmente distantes de la muestra de puntos confiables fueron correctamente clasificados mientras que los cercanos fueron clasificados como puntos de confianza. Así, en realidad tiene sentido que sean considerados como puntos confiables, porque se encuentra realmente cerca de los datos. Por lo tanto este punto "mal clasificado" en realidad no lo es, es solo algo que en el caso de las pruebas se sabe de antemano porque sabemos la etiqueta de los datos, pero si este fuera un conjunto de datos del cual no sabemos la categoría de los puntos, sería deseable que fueran considerados como puntos confiables.

Por otro lado, podemos observar que aunque el algoritmo suele distinguir cuando un punto es outlier, varios puntos confiables son clasificados de manera errónea como outliers. Es decir, tenemos varios falsos positivos (si consideramos a los outliers como la clase positiva). Son abundantes en el modelo cúbico y en el logístico.

Esto puede deberse a imprecisiones en el sistema de votación debido a que este es el que tiene mayor peso en el resultado final para decidir con cuantos puntos quedarse. Al final este sistema selecciona la solución x_q^* que recibió mayor cantidad de votos y toma los q puntos más cercanos al modelo. Sin embargo, este factor tal vez podría ser solucionado modificando el ϵ del sistema de votación. En la figura 2 podemos observar el efecto de realizar un ajuste para un valor de ϵ fijo. En este caso 1 punto que en realidad es un outlier es clasificado como punto confiable, ya que los otros dos triángulos azules

que se encuentran mezclados con los datos es normal y congruente que el modelo los clasifique como puntos confiables. Sin embargo los demás puntos son clasificados correctamente. Es importante resaltar que en ambos casos la curva ajustada es muy parecida a la otra, por lo cual la elección del ϵ de manera automática que se toma en el paper original resulta ser una buena elección, debido a que el interés es obtener un modelo que pueda predecir de manera correcta los puntos en un futuro sin que los puntos atípicos afecten el modelo final. También se aumentó el valor de ϵ para disminuir falsos positivos en el modelo cúbico (figura 4), encontrando también mejoras. Pero al seguir aumentando dicho parámetro no se logra eliminarlos por completo.

También se realizaron algunas pruebas con los tests que tienen en su repositorio de github, y en algunas de ellas el algoritmo fallaba puesto que la factorización de la matriz contenía valores infinitos continuamente debido a que el parámetro γ crecía fuertemente. Sin embargo, en los ejemplos del artículo se obtuvieron resultados similares. No logramos replicar todos los resultados en él por errores que no lográbamos depurar en nuestra implementación, por ejemplo, las estadísticas que realizan sobre scores de clasificación en varias corridas.

5. Conclusiones

1. El algoritmo propuesto en el paper original es capaz de construir un modelo que se ajuste a los datos eliminando o reduciendo la influencia de los puntos atípicos al modelo. A pesar de que en los resultados se pudo observar en algunos escenarios una gran cantidad de falsos positivos (es decir, puntos que no eran puntos atípicos que eran clasificados de esa manera por el algoritmo), los modelos ajustados eran parecidos al verdadero modelo del cual provenían los datos.
2. Se logró observar de manera empírica que el ϵ propuesto para el sistema de votación provee buenos resultados debido a que a pesar de que en los conjuntos de prueba se obtuvieron varios falsos positivos respecto a los puntos que no son outliers, modificar de manera manual este parámetro ϵ resulta únicamente en la clasificación correcta de estos puntos, sin embargo la curva ajustada no presenta cambios significativos.

6. Anexos

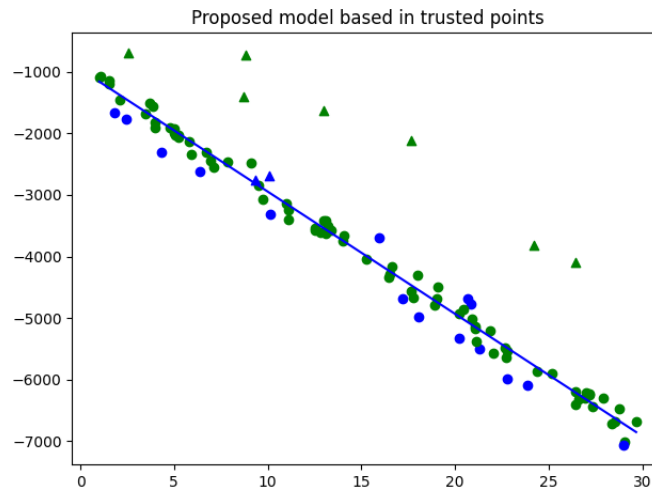


Figura 1: Ajuste Lineal para 100 puntos. Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

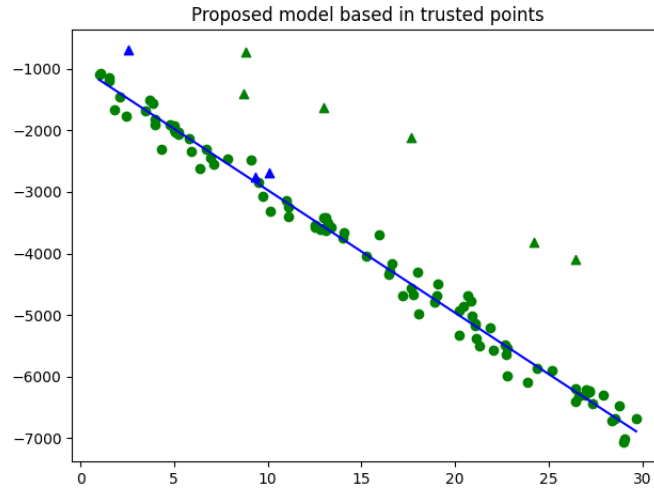


Figura 2: Ajuste Lineal para 100 puntos para $\epsilon = 60$. Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

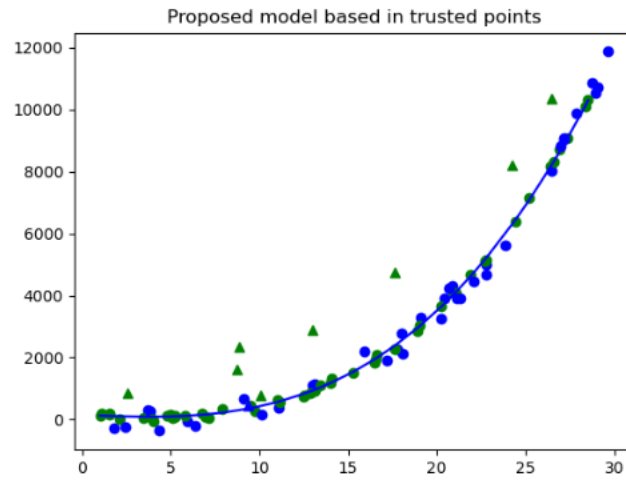


Figura 3: Ajuste Cúbico para 100 puntos. Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

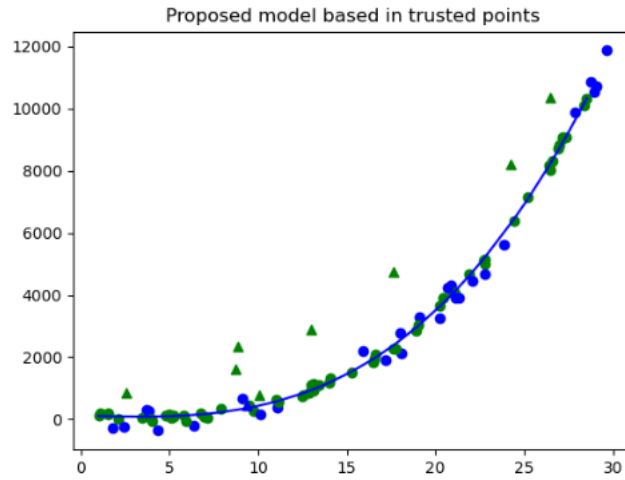


Figura 4: Ajuste Cúbico para 100 puntos aumentando el ϵ Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

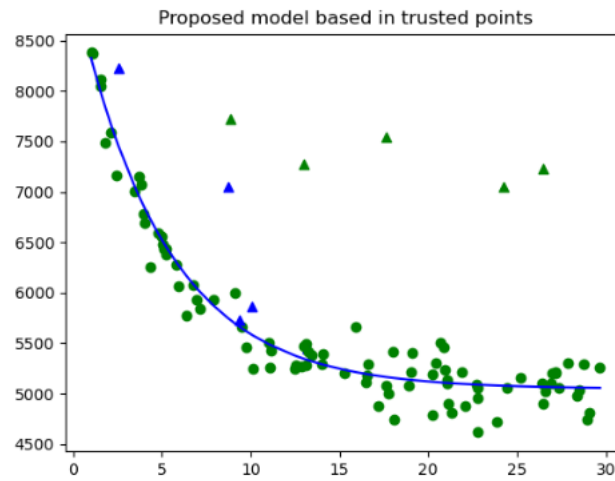


Figura 5: Ajuste Exponencial para 100 puntos. Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

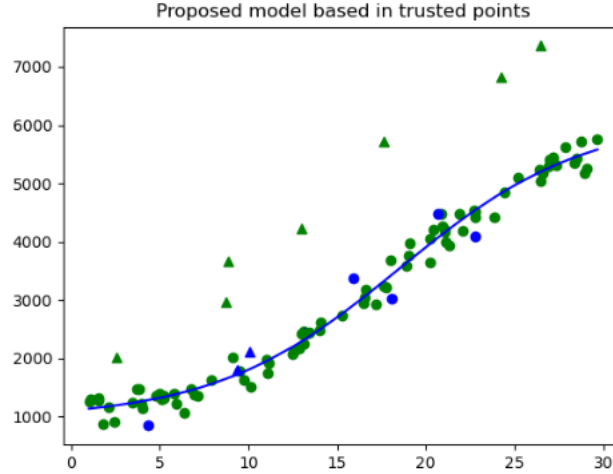


Figura 6: Ajuste Exponencial para 100 puntos. Correctamente clasificados (verde) vs. mal clasificados (azul). Clases outliers (triángulos) y puntos confiables (círculos).

Bibliografía

- [ASW93] I. A. Al-Subaihi y G. A. Watson. “Ellipse fitting by accumulating five-point fits.” En: *Pattern Recognition Letters* 14 (1993).
- [ASW04] I. A. Al-Subaihi y G. A. Watson. “The use of the l_1 and l_∞ norms in fitting parametric curves and surfaces to data.” En: *Applied Numerical Analysis and Computational Mathematics* 1 (2004).
- [AMY09] R. Andreani, J. M. Martínez y F. S. Yano. “Low Order-Value Optimization and applications.” En: *Journal of Global Optimization* 43 (2009).
- [CLS21] E. V. Castelani, R. Lopes y W. V. I. Shirabayashi. “A robust method based on LOVO functions for solving least squares problems*.” En: *Journal of Global Optimization* (2021).
- [FP08] L. Fan y T. Pylvänäinen. “Robust scale estimation from ensemble inlier sets for random sample consensus methods.” En: *Proceedings of the 10th European Conference on Computer Vision* 5304 (2008).
- [HA04] V. Hodge y J. Austin. “A survey of outlier detection methodologies.” En: *Artificial Intelligence Review* 22: 85 - 126 (2004).
- [LWP05] D. Li, D. Winfield y D. J. Parkhursts. “Starburst: a hybrid algorithm for video-based eye tracking combining featurebased and modelbased approaches.” En: *Proceedings of the Computer Vision and Pattern Recognition Workshops* (2005).
- [Rou84] P. J. Rousseeuw. “Least median of squares regression.” En: *Journal of the American Statistical Association* 79.338 (1984).