Bases de Gröbner y Machine Learning

Santiago González- Carvajal Centenera.
Tutora: M. Ángeles Zurro Moro,
Universidad Autónoma de Madrid.

May 29, 2020

Contenidos

Contenidos

Sistemas de ecuaciones polinomiales

$$\begin{cases} x^2 + y^2 - 4 & = 0 \\ \frac{xy}{2} + \frac{y^2}{9} - 1 & = 0 \end{cases}$$

- ¿Cómo lo resolvemos?
- ¿Cómo podemos interpretar el conjunto de soluciones?

Variedad afín

- El conjunto de soluciones $(a_1, ..., a_n) \in k^n$ de un sistema de ecuaciones:

$$\begin{aligned} f_1(x_1, ..., x_n) &= 0 \\ f_2(x_1, ..., x_n) &= 0 \\ &\vdots \\ f_s(x_1, ..., x_n) &= 0 \end{aligned}$$

  es denominado *variedad afín* definida por $f_1, ..., f_s$, y se denota por $V(f_1, ..., f_s)$.
- Un subconjunto $V \subset k^n$ es denominado *variedad afín* si $V = V(f_1, ..., f_s)$ para algún conjunto de polinomios $f_i \in k[x_1, ..., x_n]$.

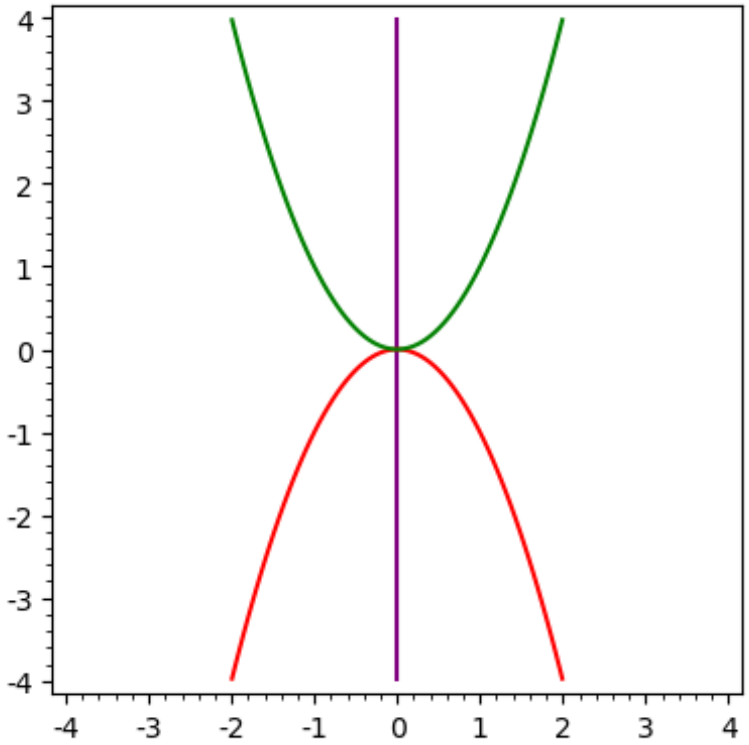
Ejemplo de variedad afín

Sistema de ecuaciones

$$\begin{cases} x &= 0 \\ y + x^2 &= 0 \\ y - x^2 &= 0 \end{cases}$$

- El punto $(0, 0)$ es el único que pertenece a la variedad.
- ¿Qué tiene de especial?

Situación geométrica



Orden monomial

Un *orden monomial* en $k[x_1, ..., x_n]$ es una relación de orden $>$ definida sobre el conjunto de los monomios x^α de $k[x_1, ..., x_n]$ que satisface:

1. Es una relación de orden total (lineal).
2. Es compatible con la multiplicación en $k[x_1, ..., x_n]$: para todo x^γ se da
$$x^\alpha > x^\beta \Rightarrow x^{\alpha+\gamma} > x^{\beta+\gamma}$$
3. Es un buen order: Toda colección de monomios no vacía tiene un elemento mínimo bajo la relación $>$.

Ejemplos de órdenes monomiales

- *Orden lexicográfico*, *orden lexicográfico graduado*, *orden lexicográfico inverso graduado*, *orden asociado a una forma lineal*.
- En SageMath `TermOrder(M)`.
  - `M = matrix(3,[1,0,0,0,1,0,0,0,1])`.
  - `P.<x,y,z> = PolynomialRing(QQ, 3, order=TermOrder(M))`.
  - $x^3 * y * z^7 >= x * y^8 * z^2$?

Lema de Dickson

Sea $I = \langle x^{\alpha} | \alpha \in A \rangle \subseteq k[x_1, ..., x_n]$ un ideal monomial. Entonces, I se puede escribir de la forma $I = \langle x^{\alpha(1)}, ..., x^{\alpha(s)} \rangle$, donde $\alpha(1), ..., \alpha(s) \in A$. En particular, I tiene una base monomial finita.

- ¿Herramienta para la demostración?

Herramienta para la demostración

Proposición
Sea $I = \langle x^\alpha | \alpha \in A \rangle$ un ideal monomial. Entonces, un monomio x^β pertenece a I si y solo si x^β es divisible por x^α para algún $\alpha \in A$..

- Si x^β es múltiplo de x^α, por la definición de ideal. ✓
- Si $x^\beta \in I$. Escribimos x^β como c.l. de elementos de I, y desarrollamos la expresión, llegando a

$$x^\beta = \sum_{i=1}^{s} h_i x^{\alpha(i)} = \sum_{i,j} c_{i,j} x^{\beta(i,j)} x^{\alpha(i)}.$$

  Esta expresión es divisible por algún $x^{\alpha(i)}$. ✓

Demostración del Lema de Dickson (1/3)

Por inducción sobre el número de variables, n.

- Para $n = 1$. I está generado por los monomios x_1^α, con $\alpha \in A \subseteq \mathbb{Z}_{\geq 0}$. Tomamos β el elemento más pequeño de A y tenemos $I = \langle x_1^\beta \rangle$. ✓
- Para $n > 1$, asumiendo que se cumple para $n - 1$, escribimos $x_1, ..., x_{n-1}, y$, y expresamos cualquier monomio como $x^\alpha y^m$. Veamos la construcción de la base monomial finita.

Demostración del Lema de Dickson (2/3)

- Para $I \subseteq k[x_1, ..., x_{n-1}, y]$, tomando $J \subseteq k[x_1, ..., x_{n-1}]$ generado por los monomios x^α para los que $x^\alpha y^m \in I$, tenemos $J = \langle x^{\alpha(1)}, ..., x^{\alpha(s)} \rangle$.
- Consideramos el ideal $J_l \subseteq k[x_1, ..., x_{n-1}]$ generado por los monomios x^β tales que $x^\beta y^l \in I$. Para estos ideales tenemos $J_l = \langle x^{\alpha_l(1)}, ..., x^{\alpha_l(s_l)} \rangle$.

Demostración del Lema de Dickson (3/3)

- Luego, tenemos que I está generado por los monomios

$$
\begin{aligned}
de\ J &: x^{\alpha(1)}y^m, ..., x^{\alpha(s)}y^m, \\
de\ J_0 &: x^{\alpha_0(1)}, ..., x^{\alpha_0(s_0)}, \\
de\ J_1 &: x^{\alpha_1(1)}y, ..., x^{\alpha_1(s_1)}y, \\
&\vdots \\
de\ J_{m-1} &: x^{\alpha_{m-1}(1)}y^{m-1}, ..., x^{\alpha_{m-1}(s_{m-1})}y^{m-1}.
\end{aligned}
$$

- Ahora, aplicamos la herramienta vista, y obtenemos la base monomial finita. ✓

Teorema de la base de Hilbert

Todo ideal $I \subseteq k[x_1, ..., x_n]$ tiene un conjunto generador finito. Es decir, $I = \langle g_1, ..., g_t \rangle$ con $g_1, ..., g_t \in I$.

- El Lema de Dickson, Diapositiva 9, nos da una forma de construir una base monomial finita de un ideal monomial.
- El Teorema de Hilbert únicamente asegura la existencia de una base finita para cualquier ideal.

Contenidos

S- polinomio

Sean $f, g \in k[x_1, ..., x_n]$ no nulos. Fijo un orden monomial y sean $LT(f) = cx^\alpha$ y $LT(g) = dx^\beta$, con $c, d \in k$. Sea x^γ el $mcm(x^\alpha, x^\beta)$. El $S - polinomio$ de f y g, que denotaremos por $S(f, g)$, es el polinomio:

$$S(f,g) = \frac{x^\gamma}{LT(f)} f - \frac{x^\gamma}{LT(g)} g$$

Criterio de Buchberger

Un conjunto finito $G = \{g_1, ..., g_t\}$ es una base de Gröbner de $I = \langle g_1, ..., g_t \rangle$ si y solo si $\overline{S(g_i, g_j)}^G = 0$ para todo $i \neq j$.

- Nos da una manera de comprobar si una base es de Gröbner.
- Pero, ¿cómo las calculamos? Algoritmo de Buchberger.

Algorithm 1: `Buchberger Algorithm`

Input : $F = (f_1, ..., f_s)$
Output: base de Gröbner $G = \{g_1, ..., g_s\}$ de $I = \langle F \rangle$, con $F \subseteq G$

1 Initialize $G := F$;
2 Initialize $G' := \emptyset$;
3 **while** $G \neq G'$ **do**
4 $G' := G$;
5 **foreach** *pair* $p \neq q$ **in** G' **do**
6 $S := \overline{S(p,q)}^{G'}$;
7 **if** $S \neq 0$ **then**
8 $G := G \cup \{S\}$;
9 **end**
10 **end**
11 **end**
12 **return** G

Herramienta para la demostración

Condición de la Cadena Ascendente

Sean $I_1 \subseteq I_2 \subseteq I_3 \subseteq \ldots$ una cadena ascendente de ideales en $k[x_1, \ldots, x_n]$. Entonces, existe un $N \geq 1$ tal que

$$I_N = I_{N+1} = I_{N+2} = \ldots .$$

Idea de la demostración:

- Consideramos $I = \bigcup_{i=1}^{\infty} I_i$.
- Demostramos que I es un ideal.
- Aplicamos el Teorema de la Base de Hilbert al ideal I y la cadena se tiene que estabilizar.

Demostración del algoritmo de Buchberger (1/3)

Vamos a demostrar que

- El conjunto G obtenido mediante el algoritmo 1 es una base de Gröbner.
- El algoritmo 1 termina.

Demostración del algoritmo de Buchberger (2/3)

Para lo primero:

- $G \subseteq I$ en todas las etapas del algoritmo?
- Inicialmente sí. Y, al ampliar G también, debido a que $G \cup \{\overline{S(p,q)}^{G'}\} \subseteq I$, ya que $p, q \in G' \subseteq G$ y $G' \subseteq I$.
- $F \subseteq G$, luego G es una base.
- Cuando $G = G'$, $\overline{S(p,q)}^{G'} = 0$ para todo $p, q \in G$. Luego, G es una base de Gröbner de $\langle G \rangle = I$ por el Criterio de Buchberger, véase diapositiva 17.

Demostración del algoritmo de Buchberger (3/3)

Para lo segundo:

- Tenemos que $\langle LT(G')\rangle \subseteq \langle LT(G)\rangle$ porque $G' \subseteq G$. De hecho si $G' \neq G$ es estricto. Veamos porqué.
- Supongamos que $\overline{S(p,q)}^{G'} \neq 0$ ha sido añadido a G. Entonces por la herramienta 10 tenemos $LT(r) \notin \langle LT(G')\rangle$, aunque $LT(r) \in \langle LT(G)\rangle$.
- Los ideales $\langle LT(G')\rangle$ forman una cadena ascendente. Por la herramienta 19 tenemos que la cadena se estabilizará y tendremos $\langle LT(G')\rangle = \langle LT(G)\rangle$. Y como acabamos de ver $G' = G$.

Bases de Gröbner en SageMath

- `P.<x,y> = PolynomialRing(QQ, 2, order='deglex').`
- `I = ideal(5*x + 3*y - 1, x^2 + y^2- 1).`
- `G = I.groebner_basis().`
- `G = [y^2 - 3/17*y - 12/17, x + 3/5*y - 1/5]`
- ¿Respecto a otros órdenes monomiales?

Contenidos

Teorema de finitud

Sea $I \subseteq k[x_1, ..., x_n]$ un ideal y fijo un orden monomial sobre $k[x_1, ..., x_n]$. Consideramos las siguientes condiciones:

1. Para cada i, $1 \leq i \leq n$, existe un $m_i \geq 0$ tal que $x_i^{m_i} \in \langle LT(I) \rangle$.
2. Si G es una base de Gröbner de I, entonces para cada i, $1 \leq i \leq n$, existe un $m_i \geq 0$ tal que $x_i^{m_i} = LT(g)$ para algún $g \in G$.
3. El conjunto $\{x^\alpha \mid x^\alpha \notin \langle LT(I) \rangle\}$ es finito.
4. El k-espacio vectorial $k[x_1, ..., x_n]/I$ tiene dimensión finita sobre k.
5. La variedad $V(I) \subseteq k^n$ es un conjunto finito.

Entonces 1-4 son equivalentes y todas ellas implican 5. De hecho, si k es algebraícamente cerrado, 1-5 son equivalentes.

Ideales de dimensión 0

- En particular, para cualquier cuerpo k algebraícamente cerrado, por ejemplo $k \subseteq \mathbb{C}$, un ideal que satisface cualquiera de las condiciones anteriores se denomina **ideal de dimensión cero**.
- Nos centraremos en la condición 5:
  "La variedad $V(I) \subseteq k^n$ es un conjunto finito".
- ¿Cómo determinamos los puntos de $V(I)$?

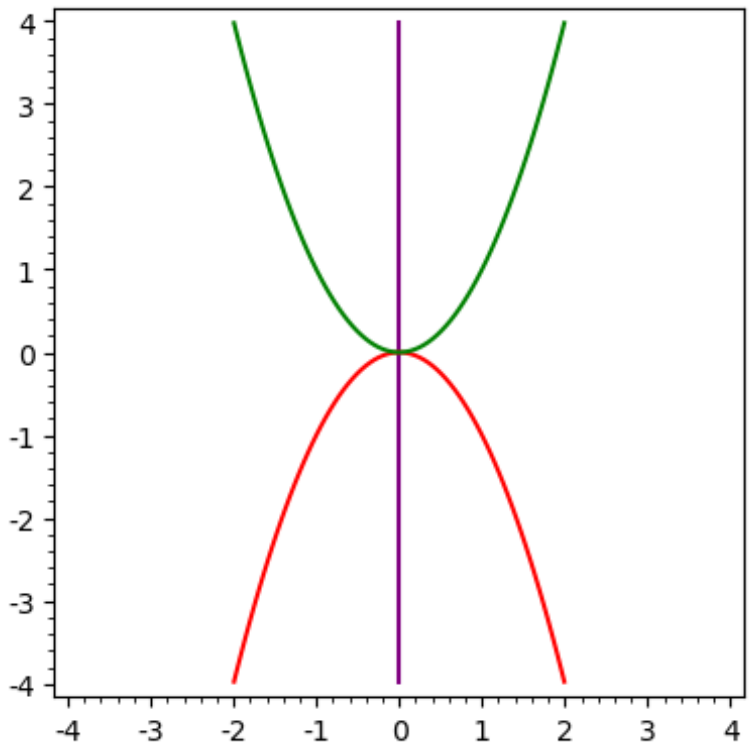
Ejemplo de ideal de dimensión 0

Sistema de ecuaciones

$$\begin{cases} x &= 0 \\ y + x^2 &= 0 \\ y - x^2 &= 0 \end{cases}$$

- Si tomamos $I = \langle x, y + x^2, y - x^2 \rangle$ como ideal en $\mathbb{C}[x, y]$, ¡I es un ideal de dimensión 0!
- En particular $V(I) = \{(0, 0)\} \subseteq \mathbb{C}^2$ finito.

Situación geométrica



Preguntas

- ¿Cómo calculamos *de forma efectiva* los puntos cuando el sistema que determina $V(I)$ se complica?
- ¿Podemos saber, a priori, si el cálculo de una base de Gröbner facilitará el problema?
- ¿Podemos abordar la pregunta anterior utilizando Machine Learning?

Contenidos

¿En qué consiste el Machine learning ?

- *"Machine learning consiste en programar computadores para optimizar un criterio de ejecución mediante datos de ejemplo o experiencia previa."* (Ethem Alpaydin, 2014).
- En la actualidad, todos generamos datos y los consumimos.
- El Machine Learning está presente en nuestra vida diaria.
- Anuncios personalizados, reconomiento facial, diagnóstico médico, reconocimiento del lenguaje, etc.

Tipos de problemas

- **Reglas de asociación**: encontrar relaciones entre distintas entidades. Por ejemplo, análisis de cestas en un supermercado.
- **Clasificación**: inferir una regla para predecir una clase determinada de entre un conjunto. Por ejemplo, en un banco, clasificar el riesgo de dar un préstamo a un cliente entre alto y bajo.
- **Regresión**: predecir un valor numérico. Por ejemplo, predecir el precio de vehículos.

Algunos conceptos básicos

- **Dataset**: conjunto de datos que vamos a utilizar para generar el modelo.
- **Train**: consiste en entrenar el modelo recibiendo las entradas y la salida correspondiente a las mismas.
- **Test**: consiste en comprobar el funcionamiento del modelo prediciendo las salidas a partir de las entradas. El modelo solo recibe las entradas.
- **Features**: características que funcionan como entrada del algoritmo de ML.
- **Label**: es la clase asociada a las features. Funciona como salida del algoritmo de ML.
- **Example**: muestra que contiene las entradas y su salida correspondiente.

Tipos de aprendizaje

- **Aprendizaje supervisado**: sabemos tanto las entradas como las salidas. Los problemas de clasificación y regresión son de este tipo.
- **Aprendizaje no supervisado**: solo conocemos la entrada. Por ejemplo, se utiliza en procesamiento del lenguaje natural, concretamente, se entrenan modelos utilizando Wikipedia.
- **Aprendizaje por refuerzo**: la salida es una secuencia de acciones donde cada acción es considerada buena si su ejecución conlleva la consecución del objetivo. Por ejemplo, en sistemas de navegación o juegos.

El experimento

Objetivo del experimento: Aplicar "Machine learning" para decidir si el cálculo de una Base de Gröbner es rentable.

Criterio de Rentabilidad

Fijo un orden monomial sobre $k[x, y]$. Sea $G = \{g_1, ..., g_t\}$ una Base de Gröbner de $I \subseteq k[x, y]$ distinto de $\{0\}$. Sean

$$n = \#\{g \in G \mid \text{g solo depende de una variable}\}$$
$$m = \#\{g \in G \mid \text{g depende de ambas variables}\}.$$

Decimos que la Base de Gröbner es rentable si y solo si $n \geq m$.

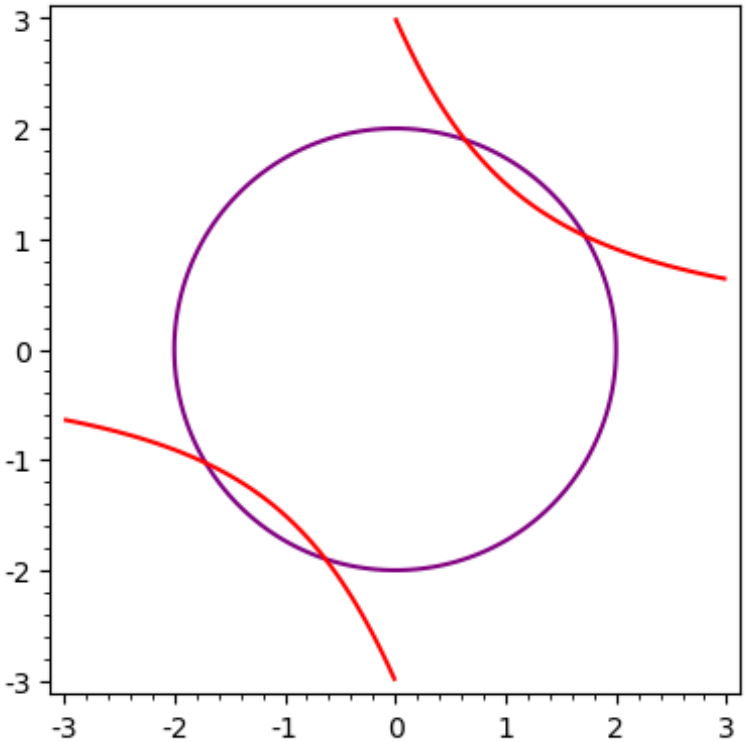
Ejemplo de base rentable (1/2)

Sistema de ecuaciones

$$\begin{cases} x^2 + y^2 - 4 &= 0 \\ \frac{xy}{2} + \frac{y^2}{9} - 1 &= 0 \end{cases}$$

- Hemos tomado $I = \langle x^2 + y^2 - 4, \frac{xy}{2} + \frac{y^2}{9} - 1 \rangle \subseteq \mathbb{C}[x, y]$ con el orden lexicográfico.
- ¿Puntos de $V(I)$?

Situación geométrica



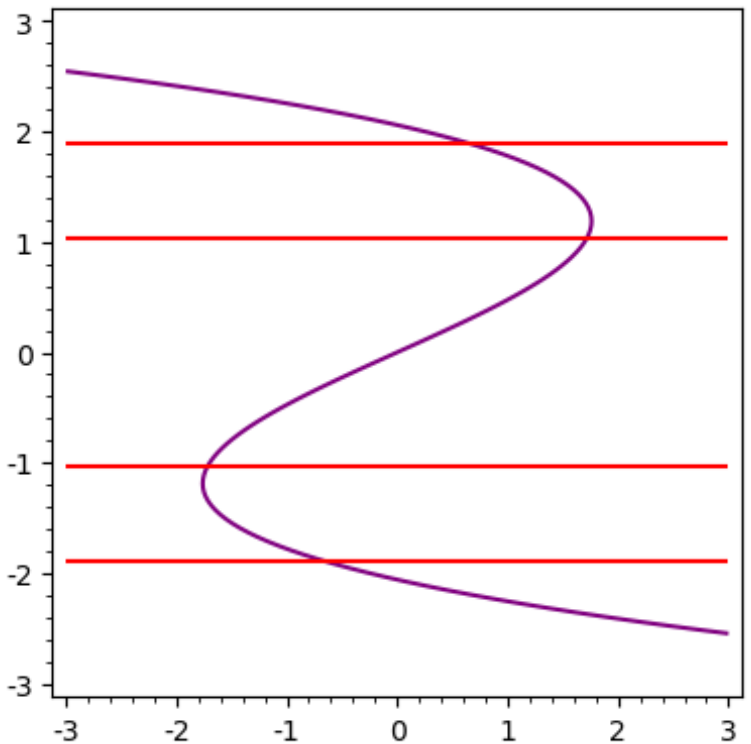
Ejemplo de base rentable (2/2)

Sistema de ecuaciones

$$\begin{cases} x + \frac{85}{162}y^3 - \frac{20}{9}y = 0 \\ y^4 - \frac{396}{85}y^2 + \frac{324}{85} = 0 \end{cases}$$

- Hemos calculado una base de Gröbner. De hecho, esta base cumple nuestro criterio de rentabilidad, Diapositiva 34.
- Más fácil ahora ¡La segunda ecuación solo depende de y!

Situación geométrica



Detalles del experimento

- Modelo utilizado: Support Vector Machine (SVM): *Separa las clases mediante un hiperplano.*
- Implementado en SageMath (sobre Python 2).
- Paquetes: `sklearn`, `numpy`, etc.
- Dataset generado con polinomios aleatorios.
- ¿Cómo es nuestro dataset? ¿Features? ¿Label?

Dataset (1/2)

- Dataset que contiene dos o tres polinomios por fila. Los generadores del ideal. Son 2 ó 3 dependiendo del fichero.
- 2 variables, orden lexicográfico, coeficientes racionales, grado entre 1 y 10.
- Tiene un total de 10000 filas.
- Los polinomios son generados de manera aleatoria.
- A partir de este dataset, hemos generado el dataset con las features y la label. El que contiene los examples, para entrenar el modelo.

Dataset (2/2)

Las *features* utilizadas son:

- Número de polinomios homogéneos.
- Diferencia total entre el grado total.
- Diferencia total entre el número de términos.
- Número de términos que dependen de x.
- Número de términos que dependen de y.
- Diferencia entre el número de términos que depende de x y el número de términos que depende de y.
- Número de componentes homogéneas.

Se tiene label = `true` si la base es rentable,
y label = false si no lo es.

Descripción del experimento

- 5200 de las 10000 bases son rentables. 80% para train; 20% para test.
- Las métricas utilizadas son:
  - $precision = \dfrac{verdaderospositivos}{verdaderospositivos + falsospositivos}$.
  - $recall = \dfrac{verdaderospositivos}{verdaderospositivos + falsosnegativos}$.
  - $F1\text{-score} = 2\dfrac{precision \cdot recall}{precision + recall}$.

Matriz de confusión

	False	True
False	608	339
True	305	748

Informe de clasificación

	Precision	Recall	F1-score	Support
False	0.67	0.64	0.65	947
True	0.69	0.71	0.70	1053
micro avg	0.68	0.68	0.68	2000
macro avg	0.68	0.68	0.68	2000
weighted avg	0.68	0.68	0.68	2000

Conclusiones

- Resultados para 2 generadores. Para 3, porcentaje de bases rentables demasiado alto (94.68%).
- El porcentaje de acierto global es 67.8%. Es mejorable, pero no está mal para una primera aproximación y el tiempo del que se ha dispuesto.
- El modelo acierta en un 71% de los casos al predecir que la base es útil. ¡Lo cual es su objetivo!

Trabajo futuro

- Ampliar el experimento a 3 variables y aplicarlo a CAD.
- Probar otros modelos.
- Pensar en nuevas *features* que añadir.
- Emplear otros lenguajes de programación para mejorar el rendimiento.

Referencias

- E. Alpaydin, *Introduction to Machine Learning*, Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts, third edition ed., 2014.
- T. Becker and V. Weispfenning, *Gröbner Bases. A Computational Approach to Commutative Algebra*, Springer-Verlag, 1993.
- D. A. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, vol. 185, Springer-Verlag, GTM, Berlin, Heidelberg, 2005.

- D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer-Verlag, UTM, Berlin, Heidelberg, 2007.
- Z. Huang, M. England, D. J. Wilson, J. Bridge, J. H. Davenport,and L. C. Paulson, *Using Machine Learning to improve cylindrical algebraic decomposition*, Mathematics in Computer Science, 13 (2019), pp. 461–488.

¡Muchas gracias
por su atención!