UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

Instituto de Física

Uso de herramientas de Inteligencia Artificial para el estudio de líquidos formadores de vidrios

T E S I S

Que para obtener el grado de

Mestría en Ciencias (Física)

Presenta

José Angel Sánchez Reyna

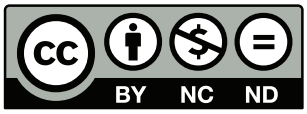
Directores de Tesis

Dr. Magdaleno Medina Noyola

Dr. Ricardo Peredo Ortiz

San Luis Potosí, SLP.

Julio 2025.



Dedico este trabajo, por supuesto, a mis padres, quienes siempre me han brindado su apoyo y han sido mi soporte.

Se lo dedico también a Ana Luisa, gracias por estar en mi vida y por tu apoyo incondicional.

Agradecimientos

Aquella persona que se haya embarcado en la realización de un trabajo tan importante y tan extenso, como lo es una tesis, seguramente conocerá de primera mano las dificultades, retos y problemas que, sin faltar, están presentes en todo momento.

No seríamos capaces de llegar a buen puerto sin el apoyo de ciertas personas que nos acompañan durante el tiempo que toma un trabajo como este, cada una de ellas de una manera particular y cada una de ellas igual de importantes.

Aquí menciono y agradezco a aquellas personas que me acompañaron y me apoyaron, a las que confiaron en mí. Sobre todo a quien, en todo momento, con su palabra era capaz de devolver la confianza que con tanta frecuencia se perdía en mí, y que en su ausencia, sin duda alguna, mi motivación se vería perdida.

La vida del estudiante de maestría, si quisiéramos reducirla a unas palabras, se reduce a asistir al instituto a tomar clase, trabajar, escoger un tema de tesis, colaborar, trabajar trabajar y trabajar. ¡Que estilo de vida tan monótono! Que difícil sería atravesar dos años así sin la compañía de nuestros amigos. No sólo para sufrir juntos una clase, o para apoyarse a completar pendientes complicados, sino que, además, la camaradería se extiende fuera del instituto, forjando así recuerdos imborrables, anécdotas para toda la vida, compartiendo sabiduría entre nosotros, de esa que suele aparecer en comunidad. Ellos representan una válvula de escape. Son las personas más cercanas y de mayor confianza con las que contamos. En fin, mis amigos Erika y Daniel fueron esos amigos que me dio el instituto y que significaron un escape de toda la presión que suele existir durante una maestría.

Únicamente es posible completar una empresa como ésta cuando a lado de nosotros se encuentran personas que saben guiarnos a través de los distintos retos, dificultades 7y problemas que se nos presentan sin remedio durante nuestro trabajo. Es gracias a

su experiencia y sabiduría transmitida al estudiante, es decir, a nosotros, que logramos resolver aquello a lo que nos enfrentamos. Gracias a su consejo encontramos la forma de ver y entender el problema y en consecuencia la forma de resolverlo. Así, le debo mi completa gratitud a mis asesores, el Dr. Magdaleno Medina Noyola y el Dr. Ricardo Peredo Ortiz. Sin la ayuda de estas brillantes personas, sin duda, no habría manera de completar con éxito este proyecto que el lector sostiene.

Igual de importante que las personas que nos acompañan directamente en este viaje, son quienes indirectamente están con nosotros en todo momento. Muchas veces el camino que elegimos seguir nos separa de nuestro lugar de nacimiento y por lo tanto nos vemos obligados a permanecer lejos de nuestros padres. Más doloroso es para ellos, pues somos hijos y por lo tanto en todo momento permanecemos en sus pensamientos. A pesar de que estemos inmersos en nuestros problemas y olvidemos llamarlos para decir "Hola", no habrá noche en la que seamos olvidados. Así, es mi placer darle las gracias a mis padres, quienes absolutamente nunca me dejan ni dejarán solo.

Ser un estudiante de maestría nos presenta un gran número de situaciones impredecibles. Una de ellas son las personas con las que llegamos a interactuar. Se convierten en conocidos, amigos. Se desarrolla, algunas veces, una confianza especial que nos une a ellos y que gracias al viaje de ser estudiante llega a tener lugar dicha conexión. Aquellas personas que merecen mención son: Orlando, Gaby, Regina, Perla, Alberto. Grandes personas, capaces, inteligentes mucho más que yo, sin duda, personas valiosas todas ellas, las cuales tengo en mucho aprecio y agradezco su amistad.

A todos ellos les doy mi más sincero agradecimiento. No tengo duda alguna en que si no fuera por estas personas la finalización de este trabajo de tesis se vería terriblemente comprometida. A todos, ¡Gracias!

Resumen

En este trabajo se presenta una primera exploración en la introducción de herramientas de Inteligencia Artificial, particularmente de aprendizaje automático (Machine Learning), para el estudio de líquidos formadores de vidrios mediante la teoría Non-Equilibrium Self-Consistent Generalized Langevin Equation (NESCGLE). Se utilizan diferentes modelos de aprendizaje supervisado y no supervisado, como regresión lineal regularizada (RIDGE y LASSO), clasificación logística y *support vector machine* (SVC), así como algoritmos de agrupamiento (K-means), con el objetivo de reducir el costo humano y computacional de los cálculos asociados a la teoría y explorar nuevas formas de análisis.

Los modelos se entrenaron con datos obtenidos a partir de sistemas físicos modelados con diferentes potenciales de interacción (esfera dura, esfera suave y pozo cuadrado), permitiendo realizar predicciones sobre propiedades dinámicas relevantes como la viscosidad o el estado de arresto. Los resultados muestran que, incluso con modelos relativamente simples, es posible alcanzar niveles de precisión destacables en tareas de predicción de valores de alguna variable y clasificación de datos, abriendo la puerta a futuras aplicaciones más robustas.

Este trabajo no pretende cerrar ninguna discusión, sino más bien abrir nuevas posibilidades, tanto en el tratamiento de datos como en la forma de abordar problemas físicos complejos, apoyándose en la capacidad predictiva de las herramientas modernas.

Índice general

Dedicatoria III

Agradecimientos IV

Resumen VI

Índice I

Lista de figuras II

Introducción 1

1. Marco teórico 4

1.1. Machine Learning 4

1.1.1. Modelos de Regresión 5

1.1.2. Modelos de clasificación 8

1.1.3. Modelos de entrenamiento No-Supervisado 11

1.2. La teoría SCGLE 12

1.2.1. Factor de Estructura y propiedades dinámicas 13

1.2.2. Ecuaciones Fundamentales 18

1.2.3. Insumos 21

1.2.4. Criterio de Arresto 25

2. Resultados Clasificación 28

2.1. Clasificaciones y diagramas de arresto. 28

2.1.1. WCA 31

2.1.2. Square Well 33

3. Resultados Regresión 35

3.1. Regresión y viscosidad. 35

3.1.1. WCA 39

3.1.2. Square Well 41

3.2. Diagramas de arresto y viscosidad. 41

4. Resultados Clasificación No-supervisada 45

4.1. Aprendizaje No-supervisado. 45

4.1.1. K-means. 49

4.2. Un método diferente 50

5. Conclusiones y Perspectivas 56

A. Algoritmos. 59

A.1. Gradient Descent 59

A.2. PCA 60

A.3. Standard Scaler 61

A.4. Accuracy score 62

B. Scripts. 63

Índice de figuras

1.1. Función sigmoide, Eq. (1.12), nótese que los valores extremos son 1 o 0. . . 9

1.2. Hiperplano óptimo con un margen (entre los hiperplanos H_1 y H_2) máximo, tomado de [1]. 11

1.3. En rojo se muestra el potencial de esfera dura. Debido a que en $r = d$, el diámetro de las partículas, el potencial es infinito, se forma una "pared impenetrable" de potencial. En negro se muestra el potencial de WCA. A diferencia del de esfera dura, este potencial presenta un aumento suave según r se acerca a d por la derecha, donde se convierte en una "pared". . 23

1.4. Potencial de esfera dura más pozo cuadrado. El perfil de pozo cuadrado se refiere a un rango $(d, d + \lambda)$ donde está presente una atracción de valor constante, que para $r > d + \lambda$ desaparece. 24

1.5. Función de auto-correlación, sistema HS, para diferentes fracciones de volumen: $\phi = 0.57$ en morado; $\phi = 0.58$ en verde; $\phi = 0.59$ en azul; y $\phi = 0.60$ en naranja. El eje x es el tiempo de correlación. 26

1.6. Desplazamiento cuadrático medio para un sistema de esferas duras. Se presentan tres casos, para $\phi = 0.40$; $\phi = 0.58$; y $\phi = 0.60$. Es decir, un caso no arrestado, otro cerca de la tracisición y el último arrestado. 27

2.1. Tiempo de evolución τ en el eje x y la función de autocorrelación F_S en el eje y. Se puede observar que la transición ocurre entre $\phi = 0.58$ y $\phi = 0.59$. 29

2.2. Clasificación de factores de estructura del sistema *HS*, usando el conjunto de prueba, por el modelo *Logistic Regression* entrenado. 31

2.3. Clasificación de factores de estructura del sistema *HS*, usando el conjunto de prueba, por el modelo *SVC* entrenado. 32

2.4. Resultados de la clasificación de factores de estructura estáticos del sistema WCA. A la izquierda (a) utilizando *LR*, a la derecha (b) *SVC*. 32

2.5. Resultados de la clasificaciónde factores de estructura del sistema SW. Se eliminaron los casos debajo de la curva spinodal. En negro se muestra la línea de arresto teórica y en verde la curva spinodal. 34

3.1. Resultados de predicción de viscosidad para el sistema *HS*, a la izquierda (a) utilizando *LASSO* alcanzando un *accuracy score* de 0.99774 y a la derecha (b) RIDGE con un *accuracy score* de 0.9995. 37

3.2. Resultados de la predicción de factores de estructura del sistema *WCA* utilizando modelos entrenados con *HS*. A la izquierda (a) utilizando *LASSO* y a la derecha *RIDGE*. En azul se muestran los datos teóricos y en rojo las predicciones de los modelos. 39

3.3. Resultados de la predicción de factores de estructura del sistema *SW* utilizando modelos entrenados con *HS*. A la izquierda (a) utilizando *LASSO* y a la derecha *RIDGE*. En azul se muestran los datos teóricos y en rojo las predicciones de los modelos 39

3.4. Resultados de predicción de viscosidad para el sistema *WCA* cuando se entrenaron los modelos con datos del mismo sistema *WCA*. A la izquierda (a) utilizando *LASSO* y a la derecha (b) *RIDGE*. En azul se presentan los datos teóricos y en rojo las predicciones. 40

3.5. Resultados de predicción de viscosidad para el sistema *SW* cuando se entrenaron los modelos con datos del mismo sistema *SW*. A la izquierda (a) utilizando *LASSO* y a la derecha (b) *RIDGE*. En azul se presentan los datos teóricos y en rojo las predicciones. 42

3.6. Diagrama de arresto (ϕ, T, η) donde la dimensión de viscosidad corresponde a la barra de colores. Resultados teóricos. 43

3.7. Resultados del diagrama de arresto con viscosidades, utilizando el clasificador de *SVC* y el modelo de regresión RIDGE, los cuales fueron entrenados previamente, como se explica en secciones anteriores. 43

4.1. . En a) los módulos dinámicos $G'(\omega)$ y $G''(\omega)$, en este caso particular se cruzan entre ellos una vez. En b) la curva $\tan(d)$, corresponde a los módulos dinámicos en a) y cruza una vez por 1. 47

4.2. Para comparar los resultados del modelo *K-means*, se realizó una clasificación manual de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$. En un script en Python se programaron las condiciones siguientes: si el número de veces que los módulos dinámicos se cruzan entre sí es 2 entonces pertenece a la región III, si se cruzan 1 vez, entonces pertenece a la región II; si ni hay cruces, entonces pertenece a I. 49

4.3. Gráfica sobre la cantidad de "información" contenida en los datos, el número mayor de dimensiones corresponde a las características con las que contamos originalmente tabla 4.2. Según ese número disminuye lo hace a su vez el número de características. Podemos ver que cuando se reduce de 87 características a 9 la cantidad de "información" se mantiene en 1.0. 50

4.4. Resultados clasificación no supervisada de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$, usando un modelo de *K-means*. 51

4.5. Resultados de clasificación de estados arrestados (rojo) y no arrestados (azul) por los modelos LR a la izquierda y SVC a la derecha. Utilizando como características de entrenamiento las presentadas en la tabla 4.3. . . . 52

4.6. Gráficas para diferentes espacios de características, a saber: "kurtosis vs mean" (a); "kurtosis vs ϕ" (b); y "std vs mean" (b). En azul se muestran casos clasificados como no-arrestados, en rojo los arrestados. 54

4.7. Gráfica donde se muestra el resultado de reducir la dimensionalidad de las características de 9 a 2, las nuevas características se nombran "x_1" y "x_2". 55

A.1. Perfil típico de la función *Mean Square Error* ($MSE(x)$). Gradient Descent busca alcanzar el mínimo de ésta función modificando el vector de parámetros $\vec{\theta}$ hasta alcanzar el vector óptimo. 60

Introducción

El objetivo de la presente tesis consiste en introducir herramientas de Inteligencia Artificial (en particular *Machine Learning*) en el manejo y análisis de los datos determinados con la teoría *Non-Equilibrium Self-Consistent Generalized Langevin Equation* (NESC-GLE). En años recientes, Machine Learning promete ser una herramienta importante para el desarrollo científico y tecnológico, ya que se ha alcanzado un éxito sin precedentes para resolver problemas físicos de gran complejidad. Por ejemplo, Buttini *et. al* [2], utilizaron un algoritmo de aprendizaje no supervisado para la detección de diferentes tipos de partículas en sistemas coloidales. En otro trabajo, Que-Salinas *et. al* [3], lograron predecir exitosamente el estado termodinámico de un líquido sólo a partir de la función de distribución radial (RDF) utilizando una *artificial neural network (ANN)*. También se ha tenido éxito en combinación con datos provenientes de experimentos, teorías y/o simulaciones, para reducir el tiempo de espera, así como el consumo computacional. En esta dirección, Jarzemski *et. al* [4], demostraron que con el uso de un modelo de *K-means clustering* se puede mejorar la rapidez de los cálculos hasta en 74 veces para obtener mapas de propiedades térmicas.

En años recientes, la Inteligencia Artificial ha provocado cambios importantes en la forma en que interactuamos y aprovechamos la tecnología. En consecuencia, se otorgó el premio nobel de Física 2024 a Geoffrey Hinton y John Hopfield por "sus descubrimientos

e invenciones fundamentales que permiten el aprendizaje automático con "redes neuronales" artificiales" [5]. Los primeros trabajos realizados en la dirección del aprendizaje automatizado se tomaron en la época de los 80's, con los trabajos de Hopfield y Hinton [5][6]. Desde entonces se han creado todo tipo de arquitecturas de redes neuronales, así como *frameworks* de trabajo, de forma que a día de hoy es muy accesible conseguir alguno de ellos, como lo son *TensorFlow*, *SciPy* o *Scikit-Learn*, entre otros. Haciendo uso de alguno de estos "motores" de IA, sólo resta escribir un *script* en algún lenguaje de programación que implemente estos *frameworks* (en el caso de este trabajo, usaremos Python como lenguaje de programación principal). De esta forma es accesible para un usuario principiante usar técnicas de Machine Learning. El usuario deberá tener cuidado con el tratamiento de los datos y los resultados que se obtengan de un modelo entrenado.

De esta forma, en el presente trabajo de tesis, estamos interesados en el fenómeno de solidificación amorfa así como en la teoría llamada "*Non-Equilibrium Self-Consistent Generalized Langevin Equation*" NESCGLE, que es nuestra principal herramienta para atacar este fenómeno. Nos proponemos llevar a cabo una primera exploración del uso de estas herramientas de Machine Learning en combinación con la teoría NESCGLE, con el fin de encontrar nuevas formas y métodos para tareas específicas, así como reducir tiempos de trabajo y recursos computacionales. Para lograrlo, generamos datos para distintos sistemas físicos haciendo uso del poder predictivo de esta teoría junto al paquete *NESCGLE* (que es un conjunto de programas escritos en el lenguaje de programación *Julia*), los cuales transformaremos según sea necesario, de forma que adquieran el formato correcto para ser entradas para los modelos de Machine Learning.

El presente trabajo de tesis está organizado como se describe a continuación. En ésta sección de Introducción se han presentado el objetivo y las motivaciones para llevar a cabo el presente trabajo. En el capítulo 1, se muestra el marco teórico donde se revisan los

modelos de Machine Learning que fueron utilizados durante el presente trabajo. Además, se revisan las ecuaciones de la teoría SCGLE, así como los sistemas físicos que fueron tomados en cuenta para la generación de datos. En el capítulo 2, se presentan los resultados obtenidos de los modelos de clasificación. En el capítulo 3, se presentan los resultados de los modelos de regresión utilizando sistemas de esfera dura, esfera suave y esfera dura más pozo cuadrado. En el capítulo 4, se presentan los resultados de una clasificación donde utilizamos aprendizaje no-supervisado que proponemos para separar en grupos un conjunto de datos de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$. También se presenta una forma alternativa para las características utilizadas durante el manejo de los modelos de clasificación supervisada del capítulo 1. Finalmente, en el capítulo 5, se presentan las conclusiones y perspectivas del presente trabajo de tesis.

Capítulo 1

Marco teórico

En este capítulo se presentan los fundamentos teóricos requeridos para llevar a cabo el presente trabajo de tesis. En primer lugar, se discuten los distintos tipos y modelos de Machine Learning que fueron utilizados, enfocándonos sólo en su funcionamiento, sin pretender hacer una descripción profunda de los mismos. Para mayor información revise la referencia [7]. En segundo lugar, se presentan las ideas físicas detrás de la teoría Self-Consistent Generalized Langevin Equation (SCGLE).

1.1. Machine Learning

En esta sección discutimos los diferentes modelos de Machine Learning utilizados. Comenzando por describir, en primer lugar, los modelos de regresión. En particular nos interesamos por RIDGE *Regression* y LASSO *Regression*. En seguida, discutimos los modelos de clasificación supervisada. En particular, Linear Clasification y SVC. Finalmente, describimos en términos generales el concepto aprendizaje no supervisado, concentrándonos en un modelo conocido como K-means Clustering.

1.1.1. Modelos de Regresión

La regresión lineal pertenece a una subclasificación de *Machine Learning* denominada "aprendizaje supervisado". El aprendizaje supervisado consiste en entrenar algún modelo modelo con ejemplos de entrada y su respectiva salida conocida, de modo que pueda aprender la relación entre ambas. La regresión lineal es un método estadístico utilizado para analizar la relación entre dos o más variables que presentan una correlación, permitiendo hacer predicciones a partir de dicha relación. En términos generales, la regresión busca determinar una variable dependiente en función de un conjunto de variables independientes [8, 9, 10]. Nombramos al conjunto de variables independientes como *las características*, correspondientes a alguna variable objetivo (es decir, una variable objetivo, lo puede ser el estado de arresto o la viscosidad, dicho de otra forma, es aquello que deseamos predecir o clasificar; y sus características el factor de estructura). Entonces, la regresión lineal o *Linear Regression* en inglés, realiza una predicción al calcular una suma pesada de las características más una constante, $\hat{y} = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$, donde θ_i son los pesos o parámetros, y x_i son las características de alguna instancia de algún conjunto de datos, en forma vectorial se ve,

$$\hat{y} = h_{\vec{\theta}}(\vec{x}) = \vec{\theta} \cdot \vec{x}, \tag{1.1}$$

donde $h_{\vec{\theta}}(\vec{x})$ es la función hipótesis usando los pesos $\vec{\theta}$, la función hipótesis arroja, a su vez, el valor predicho por el modelo $\hat{y}$. El conjunto óptimo de parámetros o pesos se determina durante el entrenamiento, donde se busca el $\vec{\theta}^*$ que minimiza una función de error. *Linear Regression* utiliza como función de error el *Mean Square Error* (MSE), como en la Eq. (1.2) [11].

$$MSE(\vec{\theta}) = \frac{1}{m} \sum_{i=i}^{m} \left(\vec{\theta} \cdot \vec{x}_i - y_i \right)^2, \tag{1.2}$$

donde el índice i corre sobre el conjunto de datos y y_i es el valor teórico correspondiente. Para minimizar el *MSE* se suele utilizar un algoritmo conocido como *Gradient Descent*, que se detalla en el apéndice A.1. En general, su objetivo es ajustar los parámetros con un ciclo iterativo, con el objetivo de encontrar el mínimo de la función de error. Una vez se ha alcanzado dicho mínimo de *MSE*, el modelo ha sido entrenado con un $\vec{\theta}^*$ óptimo y desde ese momento se puede usar para realizar predicciones sobre nuevos datos. En particular, en el presente trabajo de tesis usamos la regresión RIDGE y LASSO que a continuación de definen.

RIDGE Regression

RIDGE Regression (también conocida como *Tikhonov Regularization*), no es más que una versión regularizada de *Linear Regression* [8]. Esto significa que a la función de error se le añade un "término de regularización"

$$\alpha \sum_{i=1}^{n} \theta_i^2, \tag{1.3}$$

que en la función de error se verá de la siguiente forma

$$J(\vec{\theta}) = MSE(\vec{\theta}) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2, \tag{1.4}$$

esto provoca que, además de mejorar el ajuste de los parámetros a los datos, los mantiene tan mínimos como sea posible, especialmente aquellos que sean de poca relevancia al entrenamiento. α es un hiperparámetro que controla la magnitud de la regularización del modelo. Si α es muy grande, entonces los pesos se acercan a cero, si $\alpha \to 0$ entonces es como si el término de regularización no estuviese presente.

LASSO Regression

Least Absolute Shrinkage and Selection Operator Regression (*LASSO Regression*) es, como *RIDGE Regression*, otra versión regularizada de *Linear Regression*. Para éste algoritmo, el término de regularización en la función de error es como en la siguiente ecuación

$$J(\vec{\theta}) = MSE(\vec{\theta}) + \alpha \sum_{i=1}^{n} |\theta_i|. \tag{1.5}$$

La particularidad de este modelo es que tiende a eliminar los parámetros de las características que son menos relevantes en el entrenamiento, $\theta_j = 0$ [8, 12, 13]. Un problema surge debido a que éste modelo tiene el potencial de hacer cero a alguno o varios de los parámetros. Cuando éstos entren a *Gradient Descent*, $J(\vec{\theta})$, se volverá no diferenciable. En estos casos, es conveniente utilizar el *"subgradient vector"* $g(\vec{\theta}, J)$ que se define de la siguiente manera

$$g(\vec{\theta}, J) = \nabla_\theta MSE(\theta) + \alpha \begin{pmatrix} sign(\theta_1) \\ sign(\theta_2) \\ \cdots \\ sign(\theta_n) \end{pmatrix}, \tag{1.6}$$

donde

$$sign(\theta_i) = \begin{cases} -1 & \text{, si } \theta_i < 0 \\ 0 & \text{, si } \theta_i = 0 \\ +1 & \text{, si } \theta_i > 0 \end{cases}, \tag{1.7}$$

así, es posible obtener el mínimo para la función de error, y en consecuencia, encontrar el vector óptimo de parámetros, $\vec{\theta}^*$ [8].

1.1.2. Modelos de clasificación

La clasificación es un subgrupo del aprendizaje supervisado [14, 8]. El objetivo de los modelos que pertenecen a este subgrupo es el de clasificar un conjunto de objetos O en un grupo de clases C, las cuales pueden ser desde 2 hasta n [14]. Para cada elemento u del conjunto O existe un par correspondiente $(\vec{x}_u^i, \vec{y}_u^i)$, donde $\vec{x}_u^i$, es el vector que contiene las características de u [14, 15]. El vector $\vec{y}_u^i$ es quien contiene la información referente a cuál de las clases del grupo C pertenece, es decir, es quién provee de "las respuestas correctas" al modelo. Para lograr una clasificación se necesita encontrar algún tipo de regla con la información proporcionada, es decir, se pretende encontrar una función f que asigne a todo vector $\vec{x}_u$ una clase del conjunto C [14].

Linear Classification

Una de los algoritmos que pertenece a este tipo de modelos y que usamos en este trabajo es la llamada "regresión logística" (*Logistic Regression*). En *Logistic Regression*, se estima una probabilidad $P(x_u^i)$ de que un elemento u del conjunto O pertenezca a alguna clase de C. Si contamos con un problema de sólo dos clases, se dice se trata de un problema de clasificaciíon binaria. En dicha situación (que es también el problema en este trabajo), buscamos una probabilidad mayor a 50 % de pertenecer a una clase (la clase positiva), en caso contrario, se dice pertenece a la clase negativa [8]. Para encontrar esa probabilidad, *Logistic Regression* construye una suma pesada de las características de entrada más un término de tendencia

$$P(\vec{x}_u) = \sigma(h_\theta(\vec{x}_u)), \tag{1.8}$$

que también se puede escribir como

$$P(\vec{x}_u) = \sigma(x^T\theta), \tag{1.9}$$

donde

$$h_\theta(\vec{x}_u) = \theta_0 + \theta_1 x_{u1} + \theta_2 x_{u2} + ... + \theta_n x_{un}, \tag{1.10}$$

$$h_\theta(\vec{x}_u) = \theta_0 + \sum_1^n \theta_i x_i. \tag{1.11}$$

De la Eq. (1.10), $h_\theta(\vec{x}_u)$ es la suma pesada. Los valores $\theta_1, \theta_2, ..., \theta_n$, son los pesos o parámetros, que se busca optimizar en el entrenamiento y que multiplican a las n características contenidas en los vectores $\vec{x}_u$. La función $\sigma(t)$ en Eq. (1.9) es una función sigmiode que arroja un número entre 0 y 1, cuya expresión es Eq. (1.12). Podemos ver su comportamiento en la Fig. 1.1. Por tanto, el resultado de (1.10) será la entrada en esta función (1.12) y el resultado que arroje determinará la probabilidad de pertenecer a una clase o a otra [8].

$$\sigma(t) = \frac{1}{1 + e^{-t}}. \tag{1.12}$$

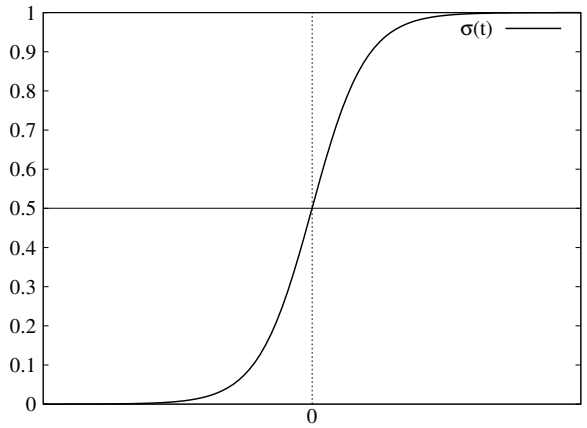


Figura 1.1: Función sigmoide, Eq. (1.12), nótese que los valores extremos son 1 o 0.

La función de coste de *Logistic Regression*, para todo el conjunto de entrenamiento, es como en la Eq. (1.13). La idea es que, a través del logaritmo, el modelo estime probabilidades altas para la clase positiva y probabilidades bajas para la clase negativa. Esto se logra utilizando un algoritmo de optimización para encontrar el vector óptimo de pesos $\vec{\theta}$ al minimizar la función de coste. Comúnmente se utiliza el algoritmo *Gradient Descent* que se discute en el apéndice A.1.

$$K(\vec{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log(P^{(i)}) + (1-y^{(i)})log(1-P^{(i)})\right]. \tag{1.13}$$

Support Vector Classification

Support Vector Machine (SVM) es un algoritmo de aprendizaje supervisado, cuyo propósito principal es la clasificación y la regresión [8]. En este trabajo explotamos las ventajas de SVM para tareas de clasificación, es decir, utilizamos *Support Vector Classification* (SVC). Algunas de esas ventajas son tiempos más cortos de predicción, además de ser una excelente técnica para relaciones lineales y no lineales [16].

El objetivo de SVC es encontrar el hiperplano, en el espacio de características, tal que el margen entre el hiperplano y los datos más cercanos a él sea máximo [16, 1], como se ve en la figura 1.2. Ese hiperplano se denomina como *optimal separation hyperplane*, y se define a partir de la siguiente ecuación

$$w^{\mathsf{T}}x_i + b = 0, \tag{1.14}$$

donde w es el vector de parámetros que define el hiperplano óptimo, b es un término de tendencia y x_i son los datos de entrenamiento. De la figura 1.2 podemos ver que sobre el margen (o la distancia entre H_1 y H_2) algunos datos pueden estar sobre H_1 y H_2, estos

puntos se conocen como los *support vectors*. De esta forma se discrimina de qué lado del hiperplano pertenecen los datos a clasificar.

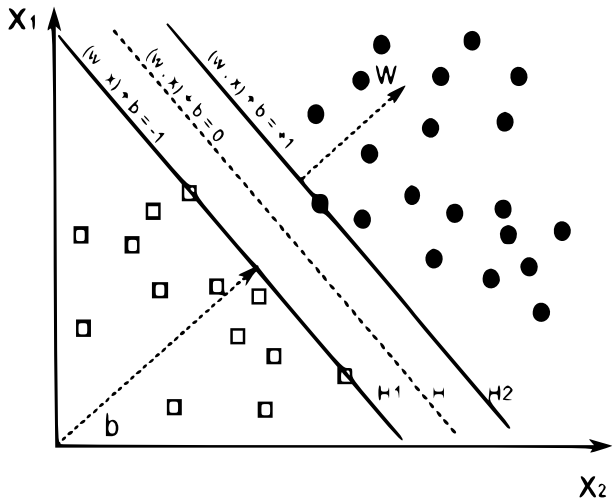


Figura 1.2: Hiperplano óptimo con un margen (entre los hiperplanos H_1 y H_2) máximo, tomado de [1].

1.1.3. Modelos de entrenamiento No-Supervisado

El entrenamiento no supervisado es un subcampo de *Machine Learning*. Se trata de un modelo que sólo recibe variables de entrada o características, $x_1, x_2, ..., x_i$, pero que no recibe etiquetas o valores objetivo [15]. Su propósito es obtener representaciones de los datos de entrada de forma que se puedan usar para distintos propósitos, tales como *decision making*, predicción de datos de entrada, conectar datos de entrada a otros modelos eficientemente, etc, [8, 15]. En pocas palabras, *unsupervised learning* (UL) se puede entender como un modelo que encuentra patrones en datos que se podrían considerar como ruido o sin estructura. Algunos algoritmos de UL son: Clustering, Anomaly detection y Density estimation [8]. En este trabajo hacemos uso de un algoritmo de *clustering*.

K-means

Se trata de un algoritmo que particiona, en k grupos o clusters, datos de entrada a través de un método iterativo que busca llegar a un mínimo local de una función criterio. En términos generales, el algoritmo consta de dos fases. En la primera se seleccionan k centros al azar z_j con $j = 1, 2, .., k$. En seguida, se calculan las distancias de cada uno de los datos a los k centros, $D(z_j, x_i)$, donde D es una función que obtiene la distanica entre dos puntos, y se agrupan con el centro más cercano. Una vez todos los datos pertenecen a un grupo "primario", se re-calculan los centros de los grupos formados. Este procedimiento se repite iterativamente hasta que la función criterio alcanza un mínimo [17, 18]. Esta función criterio se define de la siguiente manera

$$E(\vec{z}, \vec{x}) = \sum_{i=1}^{n} \sum_{j=1}^{k} |z_j - x_i|^2. \tag{1.15}$$

1.2. La teoría SCGLE

En esta sección presentamos los fundamentos de la teoría *Self-Consistent Generalized Langevin Equation* (SCGLE). Comenzamos discutiendo el problema de los sólidos amorfos fuera del equilibrio termodinámico. A continuación, se discuten el significado de la función de distribución par (FDP) y su importancia como una alternativa a las teorías clásicas (Física Estadística); además, se discutirán los conceptos de Factor de Estructura, (S(k)), como una propiedad estática, así como los conceptos de Función de Dispersión Intermedia $F(k, \tau)$, el desplazamiento cuadrático medio $\langle \Delta r^2 \rangle$) y la viscosidad η, como propiedades dinámicas. En seguida, se presenta un recorrido a través de las ideas y ecuaciones sobre las que se fundamenta la teoría SCGLE, empezando por la ecuación de Langevin, hasta llegar al conjunto cerrado de ecuaciones de la SCGLE. A continuación, discutimos los distintos

sistemas físicos utilizados para la modelación de líquidos formadores de vidrios, respecto a su potencial de interacción. En específico se describirán tres tipos: esferas duras, esferas suaves y pozo cuadrado. Finalmente, realizamos una descripción del concepto arresto dinámico, sus características y cómo identificarlo.

Es bien sabido que la materia comúnmente se presenta en tres fases: fase sólida, líquida o gaseosa. Para describir un sistema termodinámico en estado gaseoso y sólido se suele utilizar la teoría termodinámica, donde se busca encontrar una ecuación de estado, $f(N,T,V)=0$, y a partir de ella obtener las propiedades termodinámicas de dicho sistema [19, 20]. Otra teoría que trata el problema a partir de las componentes fundamentales de dicho sistema y sus interacciones entre los mismos, es la Física Estadística; donde a partir de una función de partición, $Q(N,T,V)$, se pretende encontrar sus propiedades termodinámicas [21]. Estas teorías funcionan bien bajo ciertas condiciones. La principal es que el sistema debe estar en equilibrio termodinámico $d\mathcal{F}(N,T,V)=0$, es decir, cuando se alcanza un máximo de la energía libre de Helmholtz $\mathcal{F}(N,T,V)$. Cuando esta condición no se cumple, las teorías clásicas dejan de tener validez [22, 23].

La mayor parte de los materiales con los que podemos interactuar se encuentran fuera del equilibrio termodinámico. Por mencionar algunos: vidrios, geles, cerámicos, maderas, alimentos, entre otros. En consecuencia, las teorías clásicas no son capaces de proporcionar una descripción válida de estos sistemas. Se vuelve imperante contar con una forma alterna para describir a dichos sistemas. Este método alternativo se describe en las subsecciones siguientes.

1.2.1. Factor de Estructura y propiedades dinámicas

A consecuencia de la falta de capacidad descriptiva de las teorías clásicas (Termodinámica y Física Estadística), cuando nos enfrentamos al problema de sólidos amorfos

fuera del equilibrio termodinámico, o incluso cuando nos enfrentamos a líquidos con una estructura compleja, es indispensable contar con una manera alternativa de describir estos materiales que son tan comunes y que son la mayoría de aquellos con los que interactuamos. Esa alternativa trata con la manera en la que los componentes de los sistemas termodinámicos se distribuyen en el espacio en el que están contenidos y la manera en que interactúan entre ellos a partir de un potencial de interacción a pares. La idea principal de esta descripción es la función conocida como la función de distribución par, o también conocida como, función de distribución radial (FDR) $g(r)$ [24, 21]. La función de distribución radial es de notable importancia ya que a partir de ella es posible obtener otras funciones termodinámicas [21]. Se define de la siguiente manera

$$g(r) = \frac{1}{N^2}\langle \sum_{i=1}^{N} \sum_{j \neq i} \delta(\vec{r} - \vec{r_{ij}}) \rangle. \tag{1.16}$$

La FDR nos provee información de cómo se correlacionan las densidades respecto de una partícula del sistema, y en consecuencia, de la estructura que presenta el sistema. También, nos indica una probabilidad condicional de encontrar una partícula en $r + dr$ tal que existe una partícula en r [21]. Como mencionamos antes, a partir de $g(r)$ podemos obtener distintas propiedades termodinámicas, por ejemplo

$$\frac{U^{ex}}{N} = 2\pi\rho \int_0^\infty u(r) g(r) r^2 dr, \tag{1.17}$$

$$\frac{\beta P}{\rho} = 1 - \frac{2\pi\beta\rho}{3} \int_0^\infty u'(r) g(r) r^3 dr, \tag{1.18}$$

la Eq. (1.17) se conoce como la ecuación de energía, donde, $u(r)$ es el potencial de interacción y ρ es la densidad de bulto. La Eq. (1.18) se conoce como la ecuación de presión, donde $\beta = 1/k_B T$, ρ es la densidad de bulto y $u'(r)$ es la derivada del potencial de

interacción respecto a r [24].

Otra función importante es la denominada función de correlación total, que se define como $h(r) = g(r) - 1$. Se obtiene con la ecuación de Orstein-Zernike, Eq. (1.19), en donde se hace la suposición de que el sistema termodinámico es uniforme e isotrópico. Esta ecuación depende también de la función de correlación directa, $c(r)$. Por lo tanto, tenemos dos incógnitas y se vuelve necesario una ecuación de cerradura [21]. Existen más de una ecuación de cerradura, por nombrar algunas, se cuenta con la ecuación de Percus-Yevick o con HNC (Hypernetted-Chain), entre otras. La ecuación de Orstain-Zernike tiene la forma

$$h(r_{12}) = c(r_{12}) + \rho \int c\,(r_1 3)\, h(r_{23}) d\vec{r_3}. \tag{1.19}$$

La Eq. (1.19) nos dice que la correlación total entre dos componentes individuales del sistema depende de la contribución de una correlación directa entre las dos componentes más una correlación indirecta propagada a través de componentes intermedios. Otra cantidad importante es el factor de estructura estático $S(k)$, que se obtiene analíticamente a través de la función de correlación total a partir de una transformada de Fourier de la función de correlación $h(r)$,

$$S(\vec{k}) = 1 + \rho \hat{h}(\vec{k}). \tag{1.20}$$

Se puede definir también de la siguiente manera

$$S(\vec{k}) = \langle \delta n(\vec{k}) \delta n(-\vec{k}) \rangle, \tag{1.21}$$

donde

$$\delta n(\vec{k}) = n(\vec{k}) - \langle n(\vec{k}) \rangle, \tag{1.22}$$

donde $n(\vec{k})$ es la tranformada de Fourier de la densidad microscópica[24]; es decir, el factor

de estructura estático nos brinda información acerca de la correlación en las variaciones de densidad en un mismo tiempo. Se puede obtener, además, como mediciones de intensidad en experimentos de dispersión de neutrones o de dispersión de luz, de acuerdo a la escala de los componentes del sistema a estudiar [21, 24].

Así como tenemos $S(k)$ al mismo tiempo t, podemos tener una función de la correlación a tiempos distintos. Denominada como función de dispersión intermedia, se trata de una propiedad dinámica y se define como

$$F(\vec{k}, \tau) = \langle \delta n(\vec{k}, t_0) \delta n(-\vec{k}, t_0 + \tau) \rangle, \tag{1.23}$$

donde τ es el tiempo de correlación. Cuando $\tau \to 0$, esta función se convierte en la $S(k)$. Otra cantidad que es importante mencionar es el desplazamiento cuadrático medio, $\langle \Delta \vec{r}^2(t) \rangle$, que nos provee información de cuánta distancia en promedio se ha desplazado una partícula que está inmersa en un medio fluido. Se define de la siguiente manera

$$\langle \Delta \vec{r}^2(t) \rangle = \frac{1}{N^2} \sum_{n=1}^{N} |\vec{r_n}(t) - \vec{r_n}(t=0)|^2, \tag{1.24}$$

es decir, se trata de una varianza de $\vec{r}$ al tiempo t respecto de la posición inicial $r(t = 0)$. Existen más cantidades dinámicas que es imperativo mencionar. Nos referimos a la viscosidad η y a los módulos dinámicos (módulo de relajación $G'(\omega)$, y módulo viscoso $G''(\omega)$). Estás cantidades son de especial interés, ya que a partir de ellas diseñamos dos modelos de Machine Learning. Sin entrar a detalle (se discutirán más adelante) se trata de un modelo para predecir viscosidades a partir de $S(k)$ y otro para realizar clasificaciones automáticas de $G'(\omega)$ y $G''(\omega)$. La viscosidad es un valor que indica qué tanta resistencia a fluir tiene un material [25], si este es elevado se comportará como la miel, si es bajo se comportará como el agua. Se puede calcular a partir de la teoría SCGLE de la siguiente

manera

$$\eta_0 = 1 + \int_0^\infty \Delta G(\tau) d\tau, \tag{1.25}$$

donde $\Delta G(\tau)$ es la función de relajación (shear-stress), la cual se define como

$$\Delta G(\tau) = \frac{k_B T}{60\pi^2} \int_0^\infty dk k^4 \left[\frac{1}{S(k)}\left(\frac{dS(k)}{dk}\right)\right]^2 \left[\frac{F(k,\tau)}{S(k)}\right]^2. \tag{1.26}$$

Los módulos dinámicos de relajación y viscoso contienen información de la aplicación de fuerzas cortantes y de tensión de manera oscilatoria con frecuencia ω, que se emplea en técnicas reométricas para estudiar el comportamiento de materiales ante estas interacciones. El lector interesado en la descripción de estas cantidades reológicas puede referirse a [25].

Se define a la viscosidad de corte como

$$\eta(\omega) = \eta_s + \Delta\eta(\omega), \tag{1.27}$$

donde η_s es la viscosidad del solvente y

$$\Delta\eta(\omega) = \int \frac{d^3k}{(2\pi)^3} \exp(i\omega\tau)\Delta G(\tau), \tag{1.28}$$

la cual es la transformada de Fourier de $\Delta G(\tau)$ y es una cantidad compleja, es decir, $\Delta\eta(\omega) = \Delta\eta'(\omega) - i\Delta\eta''(\omega)$. Nótese que en el límite de $\omega \to 0$, $\eta_0 \equiv \lim_{\omega\to 0} \eta(\omega)$, se recupera la viscosidad de cero corte (Eq. 1.25); además, definimos a la viscosidad de cero corte reescalada $\eta_r = \eta_0/\eta_s$, es decir, la viscosidad de cero corte sobre la viscosidad del solvente. Entonces, definimos a los módulos viscoelásticos como $G(\omega) = i\omega\eta(\omega)$ lo cual

nos indica que

$$G'(\omega) = \Delta\eta''(\omega), \tag{1.29}$$

$$G''(\omega) = \omega\eta_s + \Delta\eta'(\omega). \tag{1.30}$$

Con estos elementos que hemos mencionado, se puede realizar una descripción de los materiales amorfos fuera de equilibrio termodinámico. La teoría SCGLE se encarga de "relacionar" las propiedades estáticas con las propiedades dinámicas, siendo las estáticas los insumos que reciben las ecuaciones y las dinámicas las cantidades que se pueden obtener a partir de ella. A continuación, se discuten las ideas detrás de las ecuaciones fundamentales de la teoría SCGLE.

1.2.2. Ecuaciones Fundamentales

Como hemos dicho antes, las ecuaciones de la teoría SCGLE se encargan de tomar insumos estructurales y arrojar propiedades dinámicas, de acuerdo a algún sistema modelado con algún potencial de interacción (como lo pueden ser de esferas duras, WCA, pozo cuadrado, entre otros). Dichos sistemas, por lo general, son coloides (que son sistemas constituidos por dos o más fases, una denominada como solvente y las demás denominadas como solutos, las cuáles pueden ser constituyentes cuyas dimensiones oscilan entre los 10^{-9}m y los 10^{-3}m). En el siglo XIX con el descubrimiento del movimiento browniano por Robert Brown [21] surgió la necesidad de explicar la naturaleza de este fenómeno. En 1905, Einstein en su trabajo [26] propone una explicación teórica al movimiento browniano, donde postula que es debido a choques aleatorios de las moléculas del solvente lo que origina el movimiento, desarrollando, además, una forma de predecir el movimiento de una partícula trazadora. Más tarde, Langevin postuló la ecuación conocida como "ecuación

de Langevin" (1.31). Esta ecuación describe el movimiento de la "partícula browniana" como el resultado de una contribución de una fricción efectiva debida a las partículas del solvente, más una contribución de una fuerza efectiva$\vec{f}(t)$ debida a los choques con las demás partículas [27].

$$m\frac{d\vec{v}(t)}{dt} = -\zeta\vec{v}(t) + \vec{f}_0. \tag{1.31}$$

En la Eq. (1.31), m es la masa de la partícula de interés, ζ funciona como una "memoria" y actúa como un factor que da lugar a la fricción del medio. El término $\vec{f}(t)$ tiene la particularidad de ser una fuerza estocástica gaussiana, estacionaria y markoviana. Además, $\vec{f}(t)$ cumple las siguientes condiciones, $\bar{f}_0 = 0$ es decir, que el promedio sea cero; $\overline{f(t)f(t+\tau)} = 2\gamma\delta(t-t')$, es decir que sea delta correlacionado.

Mas adelante, en 1953 [28, 29] Onsager y Machlup desarrollaron la teoría de fluctuaciones de Onsager-Machlup. En dicho trabajo toman las ideas de Langevin para describir el movimiento browniano y las usan para describir otro tipo de sistemas termodinámicos considerando las fluctuaciones de propiedades termidinámicas reversibles, de forma que se puede escribir un equivalente a la ecuación de Langevin como

$$\frac{d\delta\vec{a}(t)}{dt} = H\delta\vec{a}(t) + \vec{f}(t), \tag{1.32}$$

donde $a_i(t)$ es un vector de propiedades termodinámicas con N componentes aleatorias, tal que $(i = 1, 2, 3, ..., N)$ y $\delta a_i(t) \equiv a_i(t) - \bar{a}_i^{ss}$, donde "ss" significa estados estacionarios; H es una matriz de relajación de $N \times N$ y $\vec{f}(t)$ es una fuerza aleatoria que cumple las mismas características que en (1.31). De forma más general, la Eq. (1.32) se puede escribir de forma que se tome en cuenta la evolución en el tiempo de la memoria

$$\frac{d\delta\vec{a}(t)}{dt} = \int_0^t H(t-t')\delta\vec{a}(t')dt' + \vec{F}(t). \tag{1.33}$$

Para llegar a la denominada ecuación generalizada de Langevin, mencionemos antes una expansión de la Eq. (1.31). En aquel caso sólo se consideraba una sola partícula inmersa en un solvente, ahora tratamos con una partícula de muchas inmersas en el solvente. En este caso escribimos la ecuación de Langevin de la siguiente forma

$$m\frac{d\vec{v_T}(t)}{dt} = -\zeta\vec{v_T}(t) + \vec{f_0}(t) + \vec{F}(t). \tag{1.34}$$

donde los primeros dos términos a la derecha de la igualdad corresponden al solvente y el término $\vec{F}(t)$ que corresponde a una fuerza total debida a la presencia de las partículas; $\vec{v_T}$ es la velocidad de la partícula trazadora [30]. $\vec{F}(t)$ tiene la forma

$$\vec{F}(t) = \int_0^t \Delta\zeta(t-t')\vec{v}_T(t')dt' + \vec{F}_0(t), \tag{1.35}$$

donde $\Delta\zeta(t)$ corresponde a una fricción efectiva debida a la presencia de las partículas que hemos agregado al sistema y $\vec{F}_0(t)$ cumple con las condiciones $\overline{F_o(t)} = 0$ y $\overline{F_0(t)F_0(t')} = k_BT\Delta\zeta(t-t')$. Con esto podemos escribir la ecuación generalizada de Langevin

$$m\frac{d\vec{v_T}(t)}{dt} = -\zeta_0\vec{v_T} + \vec{f_0}(t) - \int_0^t \Delta\zeta(t-t')\vec{v_T}(t')dt' + \vec{F}_0(t). \tag{1.36}$$

SCGLE

Ahora presentamos las ecuaciones correspondientes a la teoría Self-Consistent Generalized Langevin Equation (SCGLE). Fue desarrollada en el año 2000 por L. Yeomans y M. M. Noyola [31], que como mencionamos, su propósito es el de predecir propiedades dinámicas de un sistema coloidal a partir de propiedades estructurales estáticas. Se

pueden escribir como

$$\Delta\zeta^*(\tau) = \frac{\Delta\zeta(\tau)}{\zeta_0} = \frac{D^0}{3(2\pi)^2 n}\int d^3k k^2 \left[\frac{S(k)-1}{S(k)}\right]^2 F(k,\tau)F_s(k,\tau), \tag{1.37}$$

donde D^0 es el coeficiente de difusión; n es la densidad; $S(\vec{k})$ es el factor de estructura; $F(k,\tau)$ es la función de dispersión intermedia y $F_s(k,\tau)$ es la función de autocorrelación intermedia. Esta ecuación contiene 3 incógnitas, a saber: $\Delta\zeta^*(\tau)$, $F(k,\tau)$ y $F_s(k,\tau)$. Por lo tanto, para poder resolver esta ecuación, necesitamos otras dos, las cuales son

$$\tilde{F}(z) = \frac{S(k)}{z + \frac{k^2 D^0 S^{-1}(k)}{1+\lambda(k)\Delta\zeta^*(\tau)}}, \tag{1.38}$$

$$\tilde{F}_s(z) = \frac{1}{z + \frac{k^2 D^0}{1+\lambda(k)\Delta\zeta^*(\tau)}}. \tag{1.39}$$

Este conjunto de ecuaciones que dependen unas de otras forman un conjunto autoconsistente de ecuaciones integro-diferenciables que se resuelven iterativamente. Las Eqs. (1.38) y (1.39) están en el espacio de Fourier. Además

$$\lambda(k) = \frac{1}{1+(k/k_c)^2} \tag{1.40}$$

donde k_c es el único parámetro de ajuste y tiene un vaor de $k_c = 1.305d$.

1.2.3. Insumos

En esta sección, describimos los potenciales de interacción que utilizamos en combinación con las ecuaciones de la teoría SCGLE para generar datos de factores de estructura con el objetivo de utilizarlos en los modelos de Machine Learning.

Sistema HS

El potencial de interacción más simple entre las componenentes de un fluido es aquel conocido como sistema de esferas duras (hard sphere, HS). La característica principal de este sistema es que se considera que los componentes son impenetrables entre sí, y en consecuencia, el factor dominante es una fuerte repulsión a muy cortas distancias; esto es, de acuerdo al tamaño de los componentes. Se define de la siguiente manera

$$u(r) = \begin{cases} \infty & , \text{si } r < d \\ \\ 0 & , \text{si } r > d \end{cases}, \tag{1.41}$$

donde d es el diámetro de la partícula. El perfil de este potencial se puede ver en la figura 1.3. Para un sistema como el de HS, sólo contamos con un parámetro de control, a saber la fracción de volumen ϕ. La fracción de volumen nos indica la parte ocupada por las partículas del total del volumen que contiene el sistema. Debido a que se trata de esferas que no pueden traslaparse y a que se considera que el volumen que contiene al sistema se llena de acuerdo a *random close packaging*, se cuenta con un límite de llenado, dicho límite se alcanza con una fracción de volumen $\phi = 0.64$.

Este es el primer potencial que utilizamos para obtener datos, debido a que se trata del sistema más simple [24].

Sistema WCA

Otro potencial que utilizamos es el denominado como WCA, que es un acrónimo para Weeks-Chandler-Anderson potential. Se utiliza para estudiar fluidos simples y se define

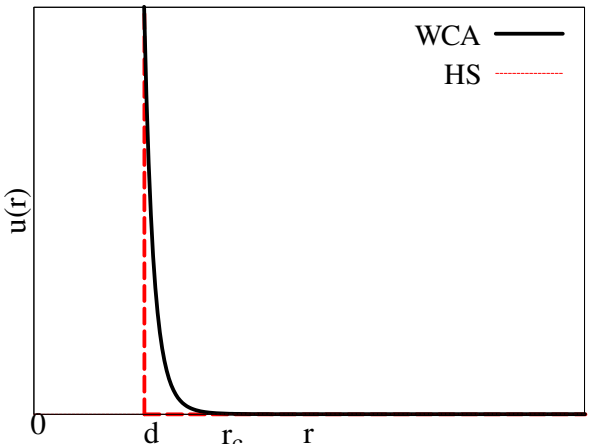


Figura 1.3: En rojo se muestra el potencial de esfera dura. Debido a que en $r = d$, el diámetro de las partículas, el potencial es infinito, se forma una "pared impenetrable" de potencial. En negro se muestra el potencial de WCA. A diferencia del de esfera dura, este potencial presenta un aumento suave según r se acerca a d por la derecha, donde se convierte en una "pared".

de la siguiente manera

$$u(r) = \begin{cases} 4\epsilon\left[\left(\frac{d}{r}\right)^{12} - \left(\frac{d}{r}\right)^{6}\right] + \epsilon & \text{, para } r < r_c \\ 0 & \text{, para } r > r_c \end{cases}, \tag{1.42}$$

donde d es el diámetro de las componentes, ϵ es un término de repulsión y $r_c = 2^{1/6}d$. Este potencial se caracteriza por ser usado para modelar esferas suaves. A diferencia de HS donde la parte repulsiva actúa como una pared impenetrable, este potencial presenta una región $(d < r < r_c)$, fuera de un núcleo duro de diámetro d, que presenta una repulsión débil a las orillas de partículas, se puede entender como una región de material suave. Además, contamos con dos parámetros de control, ϕ, la fracción de volumen y ,T, la temperatura. El perfil que toma este potencial se ve en la figura 1.3 [32].

Sistema SW

Por último, respecto de los sistemas que se utilizaron durante el presente trabajo de Tesis, haremos mención del potencial de pozo cuadrado (Square Well). El perfil de este potencial se puede ver en la figura 1.4.

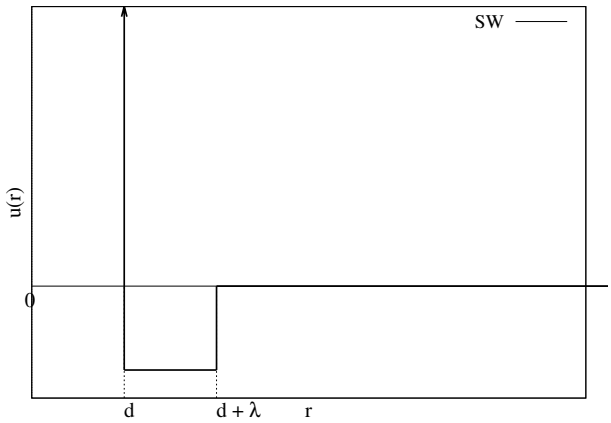


Figura 1.4: Potencial de esfera dura más pozo cuadrado. El perfil de pozo cuadrado se refiere a un rango $(d, d+\lambda)$ donde está presente una atracción de valor constante, que para $r > d + \lambda$ desaparece.

Y se define de la siguiente manera

$$u(r) = \begin{cases} \infty & \text{, para } r < d \\ -u_0 & \text{, para } d > r > \lambda \\ 0 & \text{, para } r > \lambda \end{cases} , \tag{1.43}$$

donde d es el diámetro de las componentes, λ es el ancho del pozo y u_0 es la magnitud de la parte atractiva del potencial. Este potencial se utiliza para modelar líquidos utilizando esferas duras tal que, además, sientan una atracción en un cierto rango λ. Se cuenta con tres parámetros de control: ϕ la fracción de volumen, T la temperatura y λ el ancho del pozo [24].

1.2.4. Criterio de Arresto

La primer exploración que realizamos en este trabajo, utilizando Machine Learning en combinación con las ecuaciones de la SCGLE, es la de predecir si un sistema se encuentra en arresto dinámico o no, a partir del factor de estructura, $S(k)$. Para ello, aprovechamos el poder predictivo de la teoría SCGLE, en particular para el sistema de esferas duras (que es el más sencillo), y definimos un criterio para determinar si el sistema esta arrestado o no. Como método principal de determinación, utilizamos la función de auto-correlación intermedia $F_s(\vec{k},\tau)$, que vemos en la figura 1.5 para diferentes fracciones de volumen. $F_s(\vec{k},\tau) = \langle \delta n(\vec{k},t_0)\delta n(-\vec{k},t_0+\tau)\rangle$, nos indica qué tan correlacionada está una partícula en una posición después de un tiempo $t_0+\tau$, respecto de su posición original. Cuando $\phi = 0.57, 0.58$ la manera en que la magnitud de la correlación decae presenta un estancamiento, esto quiere decir que la partícula mantiene durante un tiempo información (a través de la correlación) del punto de inicio, pero eventualmente llega a 0, significando que pierde toda "memoria" de su posición inicial. Dicho en otras palabras, la partícula ha fluido. Cuando $\phi = 0.59, 0.60$, en cambio, la magnitud de la correlación nunca decae (al menos hasta magnitudes en tiempo de 10^6). Es decir, la partícula nunca pierde información sobre su origen, debido a que permanece estancada en un lugar donde la correlación se vuelve constante, en otras palabras está arrestada. Esta transición ocurre entre $\phi = 0.58$ y $\phi = 0.59$, por lo tanto nuestro criterio, para esferas duras, será que si $\phi > 0.58$ entonces el sistema está arrestado.

Otra forma de determinar si un sistema está arrestado es de acuerdo a el desplazamiento cuadrático medio, $\langle \Delta r^2 \rangle$, en combinación con la longitud de localización γ. La longitud de localización es la distancia entre el mínimo de $\langle \Delta r^2 \rangle$ y consigo misma a tiempos asintóticos, ver Fig. 1.6. El criterio de γ dice que, cuando γ toma un valor finito entonces el sistema es un estado arrestado. En caso contrario, si γ toma valores del orden

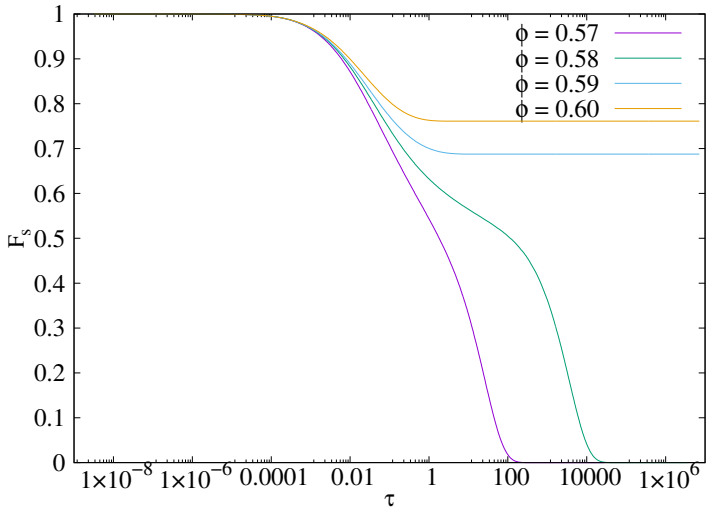


Figura 1.5: Función de auto-correlación, sistema HS, para diferentes fracciones de volumen: $\phi = 0.57$ en morado; $\phi = 0.58$ en verde; $\phi = 0.59$ en azul; y $\phi = 0.60$ en naranja. El eje x es el tiempo de correlación.

de 10^{12}, es decir valores que se acercan a infinito, entonces se dice que el sistema no esta arrestado.

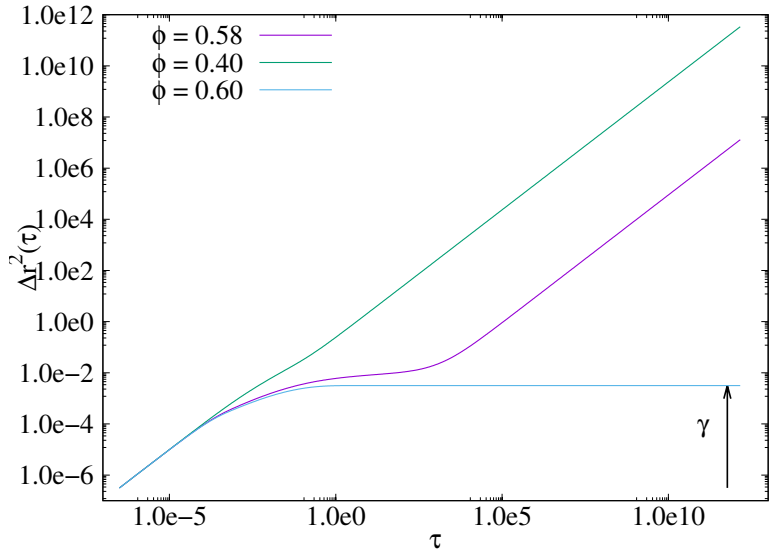


Figura 1.6: Desplazamiento cuadrático medio para un sistema de esferas duras. Se presentan tres casos, para $\phi = 0.40$; $\phi = 0.58$; y $\phi = 0.60$. Es decir, un caso no arrestado, otro cerca de la tracisición y el último arrestado.

Capítulo 2

Resultados Clasificación

En este capítulo, y en los dos siguientes, se discutirán los resultados más relevantes obtenidos durante el desarrollo del presente trabajo. En primer lugar, se presentan los hallazgos referentes a la clasificación de factores de estructura estáticos, $S(k)$, de acuerdo a si corresponden a un estado arrestado o a un estado no arrestado. Se utilizaron dos modelos de Machine Learning, *Linear Classification* y *SVC*. Además, se presentan diagramas de arresto construidos con las predicciones de los modelos.

2.1. Clasificaciones y diagramas de arresto.

En esta sección discutimos la capacidad de algunos modelos de Machine Learning para realizar clasificaciones sobre alguna característica importante de algún conjunto de datos. En este caso particular, se trata de clasificar factores de estructura estático, $S(k)$, de acuerdo a su estado de arresto. Nuestro conjunto de datos o *DataBase* (DB), serán factores de estructura, $S(k)$, para el modelo físico de esferas duras (HS), de forma que contemos con un rango de $(0.1, 0.64)$ para la fracción de volumen ϕ. Apoyándonos en

el software desarrollado previamente por el grupo de trabajo del Dr. Magdaleno Medina Noyola del Instituto de Física, el cual se menciona en el apéndice B, logramos obtener un total de 2297 factores de estructura, es decir, para 2297 fracciones de volumen distintas. De ese total una parte servirá como conjunto de entrenamiento (344), otra parte como conjunto de validación (976) y otra parte como conjunto de prueba (977). A cada uno de los $S(k)$ se les asignó una etiqueta con dos posibles valores: "yes" para aquellos casos donde el factor de estructura corresponda a un sistema arrestado y "no" para aquellos que correspondan a un caso no arrestado. Para determinar si se trata o no de un sistema arrestado utilizamos el criterio de la función de auto-correlación, mencionado en la sección 1.2.4, Fig. 2.1.

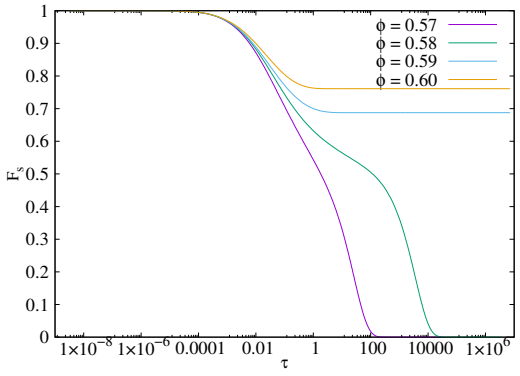


Figura 2.1: Tiempo de evolución τ en el eje x y la función de autocorrelación F_S en el eje y. Se puede observar que la transición ocurre entre $\phi = 0.58$ y $\phi = 0.59$.

En la Fig. 2.1 observamos que ocurre un cambio en el comportamiento de F_s entre $\phi = 0.58$ y $\phi = 0.59$. Explícitamente, la transición ocurre en $\phi = 0.582$. Para valores de ϕ menores, la autocorrelación decae hasta 0, es decir, no es un estado arrestado. Por el contrario, para valores mayores, la autocorrelación forma un plateau, resultando así en un estado arrestado. En conclusión, se etiquetaron los $S(k)$ según su valor de fracción de volumen, si $\phi > 0.582$ entonces se trata de un estado arrestado.

Entonces, para nuestra base de datos contamos con $S(k)$, la fracción de volumen ϕ y una etiqueta para determinar si está o no arrestado. Con estos datos se entrenaron dos modelos de clasificación supervisada: *Logistic Regression* y *SVC*, los cuales se discutieron en la sección 1.1.2. Para ambos, la forma de los datos de entrada consistió en separar los valores contenidos en arreglos vectoriales, tal que por cada uno de esos valores se contará con una característica que eventualmente tomarán los modelos. Por ejemplo, la información de nuestro $S(k)$ está contenida en un vector cuyos elementos corresponden al espaciado en k que se estableció al momento de generar la base de datos. Además, se tendrá la columna de etiquetas. Una porción de la base de datos se puede ver en la tabla 2.1. Se descartó incluir el valor de ϕ de entre las caracteríaticas con el objetivo de obligar a los modelos a encontrar una solución óptima sólo a partir del factor de estructura.

S_0	S_1	S_2	S_3	...	S_{436}	S_{437}	S_{438}	S_{439}
0.162774	0.162893	0.163251	0.16385	...	1.004588	1.004863	1.005087	1.005259
0.0858584	0.0859234	0.0861186	0.0864451	...	1.007391	1.007947	1.0008419	1.008804
0.0362354	0.0362622	0.0363427	0.0364774	...	1.01236	1.013616	1.014726	1.015679
0.031592	0.0316152	0.0316849	0.0318015	...	1.013271	1.014686	1.015944	1.017032
0.00934796	0.00935431	0.00937341	0.00940533	...	1.022123	1.025887	1.029372	1.032541

Cuadro 2.1: Tabla de los datos preparados para ser suministrados a los modelos. Datos correspondientes a factores de estructura, $S(k)$, para un sistema de esferas duras.

Una vez contamos con nuestra partición de entrenamiento, que se conforma de acuerdo a la tabla 2.1, alimentamos a nuestros modelos LR y SVC para el entrenamiento, la columna que contiene las etiquetas (*yes*, *no*), se suministra como un objeto aparte en el entrenamiento, se debe tener cuidado que elemento a elemento de ambos objetos (tabla de datos y columna de etiquetas), sean correspondientes. Concluido el entrenamiento lo validamos poniéndolo a prueba con nuestros conjuntos de validación y prueba.

En las Figs. 2.2 y 2.3 vemos los resultados de predicción de estados arrestados para el conjunto de prueba (para el sistema de esfera dura), utilizando un modelo LR y uno SVC,

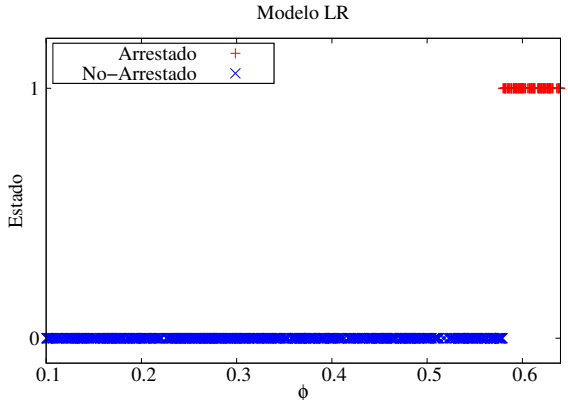


Figura 2.2: Clasificación de factores de estructura del sistema *HS*, usando el conjunto de prueba, por el modelo *Logistic Regression* entrenado.

respectivamente, previamente entrenados. Las predicciones mostradas en 1 y en rojo son estados arrestados; los que están sobre 0 y en azul son estados no arrestados. En el eje x tenemos a la fracción de volumen ϕ. En ambos modelos las predicciones indican que la transición entre estados arrestados y no arrestados está en aproximadamente $\phi \approx 0.58$.

Una vez tenemos un modelo entrenado que es capaz de discernir entre estados que están arrestados y estados que no lo están, nuestra intención es ponerlo a prueba ahora con factores de estructura que no han pasado por el modelo y que, además, vienen de modelos físicos distintos.

2.1.1. WCA

El primer sistema que utilizamos para probar nuestra herramienta fue el llamado WCA (utilizado para modelar esferas suaves), el cual se ha discutido en la sección 1.2.3. Escogimos este sistema debido a que, después del sistema de esfera dura, WCA es el más sencillo. Para la recolección de datos se hizo uso del paquete NESCGLE, escrito en *Julia*, utilizando un módulo especial para calcular la matriz de estabilidad en WCA tal

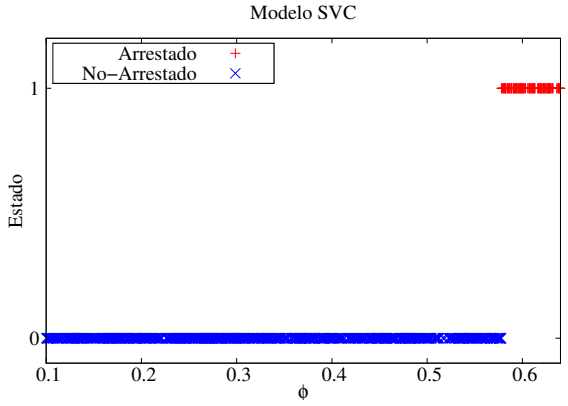


Figura 2.3: Clasificación de factores de estructura del sistema *HS*, usando el conjunto de prueba, por el modelo *SVC* entrenado.

que contamos con dos grados de libertad: ϕ y temperatura T. Sus rangos son $(0.5, 0.9)$ y $(0.1, 10)$, respectivamente. Para realizar las predicciones se utilizaron los modelos de *LR* y *SCV* que fueron entrenados anteriormente con datos de esfera dura 2.1.

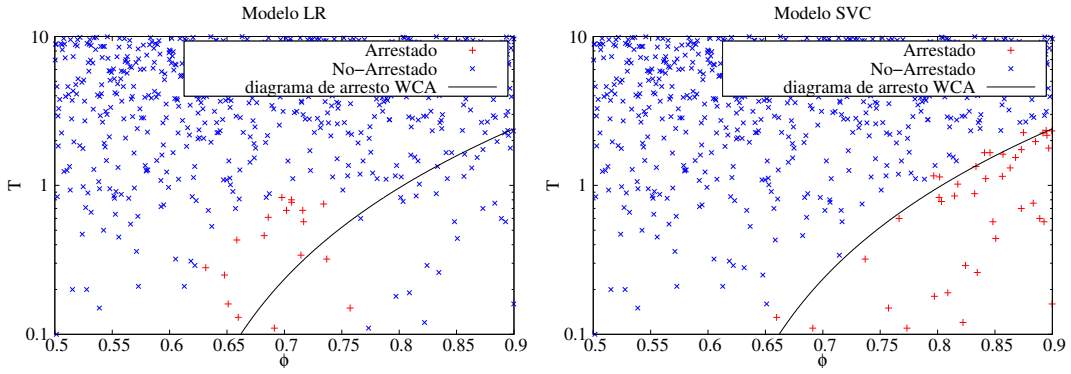


Figura 2.4: Resultados de la clasificación de factores de estructura estáticos del sistema WCA. A la izquierda (a) utilizando *LR*, a la derecha (b) *SVC*.

En la Fig. 2.4 podemos observar los diagramas de arresto correpondientes a las predicciones realizadas por los modelos previamente mencionados. Sobre el eje x tenemos la fracción de volumen ϕ y en el eje y la temperatura T. En azul y con cruces, se presentan

los casos clasificados como no arrestado y en rojo los casos clasificados como arrestados. Sobre cada uno de los diagramas, además, se sobrepone el diagrama de arresto teórico en color negro. En el primer diagrama 2.4 a), correspondiente al modelo *Logistic Regression*, podemos notar que la interfase entre los estados arrestados y no arrestados se desvía del recorrido de la línea de arresto teórica. Podemos decir que, la precisión de este modelo frente a factores de estructura para sistemas WCA es muy pobre y en consecuencia debe ser descartado. En el segundo diagrama, Fig. 2.4 b), vemos los resultados del modelo SVC, los cuales, a diferencia del modelo anterior, tienen mejor desempeño. Como se puede apreciar, los casos entre 0.65 y 0.75 para ϕ, donde los estados arrestados aparecen por encima de la línea teórica. Podemos decir que el modelo SVC se comporta de mejor manera que LR para esta tarea.

2.1.2. Square Well

De acuerdo a los resultados anteriores se consideró que el mejor modelo será el de SVC para el discernimiento entre estados arrestados y no arrestados a partir del factor de estructura.Tomando esto en cuenta, se decidió explorar un poco más allá la validez del entrenamiento con sólo ejemplos de HS. Para ello, con ayuda de la paquetería NESCGLE se obtuvieron datos para $S(k)$ pero ahora para el modelo físico de esfera dura más pozo cuadrado (SW), que es un sistema muy estudiado por el grupo y es de interés en el modelado de laponitas [23]. La interacción SW se discutió brevemente en la sección 1.2.3. Para este sistema, de nuevo contamos con dos grados de libertad, ϕ y T, para los cuales se considerarón los siguientes rangos de valores $(0.01, 0.45)$ y $(0.1, 1.8)$, respectivamente, el ancho para el pozo es de $\lambda = 1.5$. Entonces, para este ejercicio sólo se hizo uso del mejor modelo, SVC. Los resultados se pueden ver en la Fig. 2.5. En color negro, se muestra la línea de arresto teórica. Sabemos que para el sistema SW existe una curva spinodal.

Dicha curva spinodal, que las teorías físicas que tratan los problemas en equilibrio no son capaces de acceder en esa región, no les es posible realizar predicciones en dicha región y numéricamente se obtienen resultados que carecen de sentido [21]. Esta curva se presenta en color verde.

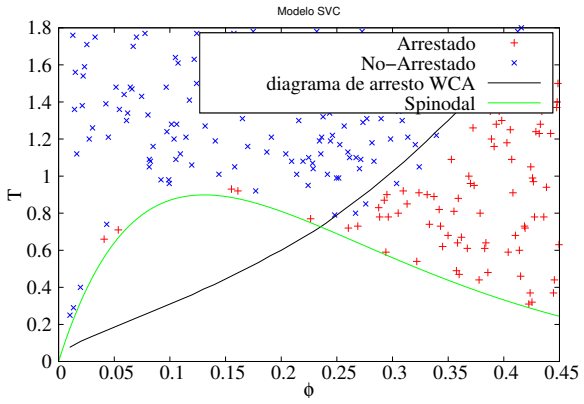


Figura 2.5: Resultados de la clasificaciónde factores de estructura del sistema SW. Se eliminaron los casos debajo de la curva spinodal. En negro se muestra la línea de arresto teórica y en verde la curva spinodal.

Se aprecia que, fuera de la región spinodal, la interfaz entre estados arrestados y no arrestados parece seguir la tendencia de la curva teórica, sin embargo, no siguen la misma trayectoria sino que difieren como curvas paralelas. Esto parece indicarnos que aunque el modelo puede reconocer estados arrestados en este sistema. Cerca de la frontera, definida por la curva de arresto, tiene problemas para discernir entre ambas opciones. Por lo tanto, podemos decir que el modelo es capaz de clasificar correctamente los factores de estructura hasta cierto porcentaje, habiendo sido entrenado sólo con ejemplos de *HS*. En consecuencia, es necesario considerar diferentes tipos de características para entrenar los modelos, o bien, entrenar con datos de SW para obtener mayor calidad de resultados de SW.

Capítulo 3

Resultados Regresión

En el presente capítulo, se presentan los resultados correspondientes a el uso de los modelos de regresión. En específico, a *Lasso Regression* y *Ridge Regression*. Donde a partir del factor de estructura estático calculamos el valor de la viscosidad, η. Además, utilizamos los resultados anteriores para obtener un diagrama de arresto con viscosidad.

3.1. Regresión y viscosidad.

En esta sección consideramos los modelos que mejor comportamiento alcanzaron al predecir valores de viscosidad η. Inspirados por el resultado de la sección anterior, donde un modelo entrenado a partir de datos del sistema *HS* era capaz de arrojar resultados satisfactorios para otros sistemas como *WCA* o *SW*, buscamos entrenar un modelo que se especialice en predecir valores de viscosidad, a partir de datos de un sistema simple como (*HS*). Pretendemos con ese mismo modelo lograr obtener valores acertados de viscosidad para sistemas más complejos. Para ello, como se trata de un entrenamiento supervisado, además de los factores de estructura estáticos $S(k)$, es necesario contar con los correspon-

dientes valores de viscosidad η para cada instancia. Con este fin se utilizó un algoritmo para obtener dichos valores, se puede encontrar en el apéndice B.

Se generaron un total de 500 factores de estructura estáticos con su correspondiente valor de η para un rango en ϕ de $(0, 0.58)$. Como vimos anteriormente, en *HS* cuando estamos en la región donde $\phi > 0.58$ nos encontramos con estados arrestados, por lo tanto, para estos casos la viscosidad η es tan grande que se puede considerar $\eta \to \infty$ y tenemos la certeza que así será para todos. Entonces, de nuestros 500 factores de estructura obtendremos tres conjuntos: conjunto de entrenamiento, conjunto de validación y conjunto de prueba. Nuestra nueva base de datos, en conclusión, contará con los $S(k)$, ϕ y η, para el sistema *HS*. Como modelos, se consideraron los llamados *LASSO Regression* y *RIDGE Regression* debido a que, como se menciona en la sección 1.1.1 donde se discuten con más detalle, estos modelos penalizan las características que son de menor relevancia para el entrenamiento. Así como en la sección 3.1, la forma de entrada de los datos a los modelos es la misma, a excepción de que ahora en lugar de tener una columna con etiquetas tendremos una columna con los valores correspondientes de η. Podemos ver una parte de esta base de datos en la tabla 3.1.

S_0	S_1	S_2	S_3	...	S_{468}	S_{469}	S_{470}	$S_{471}v$
0.0695394	0.069592	0.0697499	0.0700141	...	0.992411	0.991863	0.991399	0.991022
0.0137658	0.0137754	0.0138043	0.0138525	...	0.98216	0.979977	0.978009	0.97627
0.103387	0.103465	0.103699	0.104091	...	0.994195	0.99383	0.993528	0.993293
0.00970192	0.00970853	0.00972839	0.00976161	...	0.97938	0.976552	0.97398	0.971683
0.0136147	0.0136242	0.0136527	0.0137004	...	0.982074	0.979873	0.977888	0.976134

Cuadro 3.1: Primeras 5 filas de la base de datos de entrada para entrenamiento de los modelos *LASSO* y *RIDGE* para predicción de viscosidad. Usando factores de estructura, $S(k)$, correspondientes a un sistema de esfera dura.

Una vez entrenados los modelos, se realizaron las correspondientes predicciones para los conjuntos de prueba en cada modelo. Los resultados se pueden apreciar en la Fig 3.1.

Podemos ver los resultados para la predicción de viscosidad de los modelos entrenados en color rojo, y en azul se muestran los valores de viscosidad teóricos calculados a partir de la teoría NESCGLE. En 3.1 a) vemos el caso para regresión *LASSO*; en el eje y se tiene $log_{10}(\eta)$. Vale la pena señalar que por cada punto rojo (es decir, el valor $log_{10}(\eta)$ para una ϕ específica) existe el valor teórico en color azul de $log_{10}(\eta)$ para la misma ϕ. Podemos ver que para estructuras cuyo valor de ϕ es menor a 0.5, los valores arrojados por el modelo coinciden de buena manera con los valores teóricos. Para estructuras con ϕ mayores a ese valor comienzan a presentarse inconsistencias de los valores predichos respecto a los teóricos.

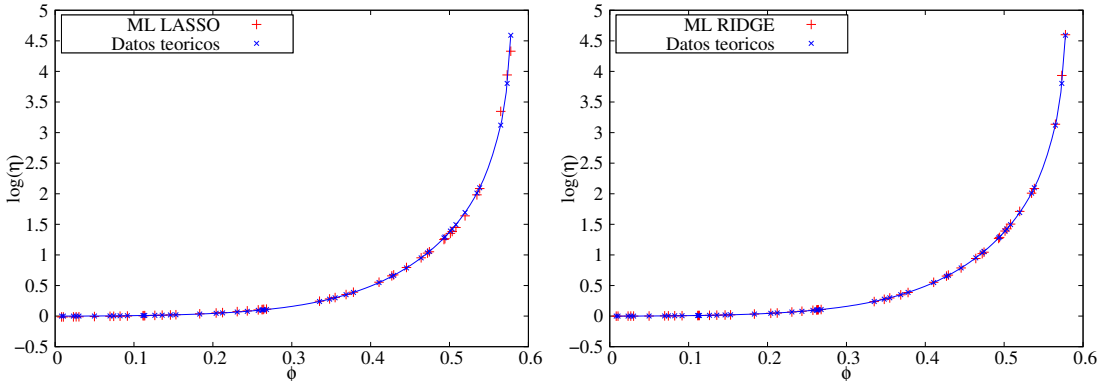


Figura 3.1: Resultados de predicción de viscosidad para el sistema *HS*, a la izquierda (a) utilizando *LASSO* alcanzando un *accuracy score* de 0.99774 y a la derecha (b) RIDGE con un *accuracy score* de 0.9995.

En la Fig. 3.1 b), correspondiente al modelo de regresión *RIDGE*, vemos que, a diferencia del rendimiento del modelo anterior, obtenemos resultados buenos a lo largo de todo el rango de valores en ϕ. Podemos concluir que parece ser un mejor modelo para esta tarea en específico. Sin embargo, un aspecto donde este modelo es inferior al anterior es en la penalización de características que no son relevantes para el entrenamiento, ya que *LASSO* es capaz de anular los pesos de las características irrelevantes, mientras que

RIDGE sólo les asigna un peso cercano a cero.

Ya que ambos modelos tienen resultados satisfactorios para la predicción de $log_{10}(\eta)$, usaremos ambos al ponerlos a prueba con factores de estructura de sistemas diferentes al de *HS*. Para la obtención de datos en esta prueba, tomamos en cuenta los sistemas *SW* y *WCA*, así como lo hicimos para los diagramas de arresto. Entonces sometimos los datos nuevos a los modelos entrenados para realizar predicciones al valor de la viscosidad a partir de sólo el factor de estructura. Los resultados se pueden ver en las Figs. 3.2 y 3.3 para *WCA* y *SW*, respectivamente. Debido a que tanto en *WCA* y *SW* podemos controlar dos parámetros. Para visualizar los resultados fue necesario hacer un barrido sobre alguno de esos parámetros dejando fijo al otro. En este caso, dejamos fija a la temperatura. Como antes, en rojo se presentan los valores predichos por los modelos (a la izquierda *LASSO* y a la derecha *RIDGE*) y en azul los valores teóricos. El modelo *LASSO* arrojó un *accuracy score* de -2.2286, mientras que el modelo *RIDGE* arrojó -135.1079 para su *accuracy score* para los datos de WCA. Para los datos de SW los *accuracy score* de los modelos *LASSO* y *RIDGE* fueron -2.5106 y -29.8788, respectivamente. *Accuracy score* se detalla en el apéndice A.4, se trata de un valor que cuantifica la cantidad de predicciones correctas, donde el valor perfecto es 1.

Podemos observar que, al haber entrenado un modelo de regresión utilizando datos de *HS*, cuando lo ponemos a prueba con datos de otros sistemas, el comportamiento es desastroso. El estado óptimo al que llega el modelo entrenado no es suficiente cuando se trabaja con datos de otro sistema, como lo son *WCA* y *SW*. Los resultados anteriormente presentados nos parecen indicar que la predicción de cantidades, como la viscosidad, es lo suficientemente compleja como para refutar nuestra hipótesis de obtenerlas con modelos entrenados con datos de sistemas más sencillos. A continuación probamos el rendimiento

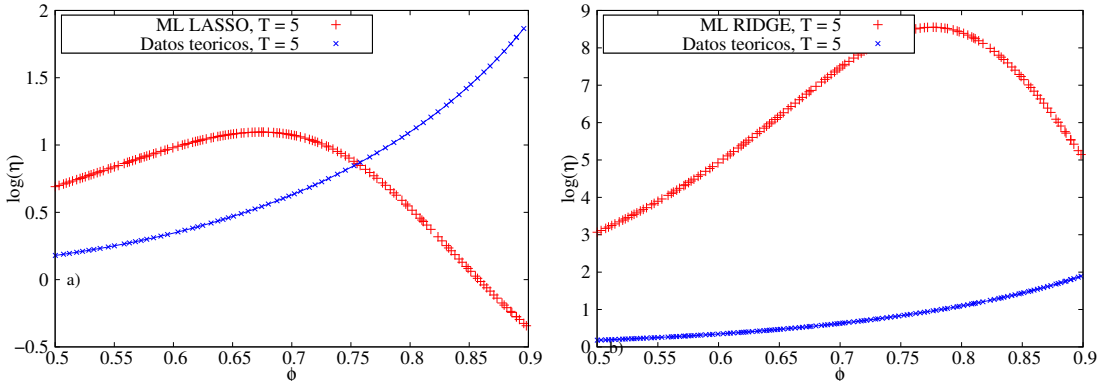


Figura 3.2: Resultados de la predicción de factores de estructura del sistema *WCA* utilizando modelos entrenados con *HS*. A la izquierda (a) utilizando *LASSO* y a la derecha *RIDGE*. En azul se muestran los datos teóricos y en rojo las predicciones de los modelos.

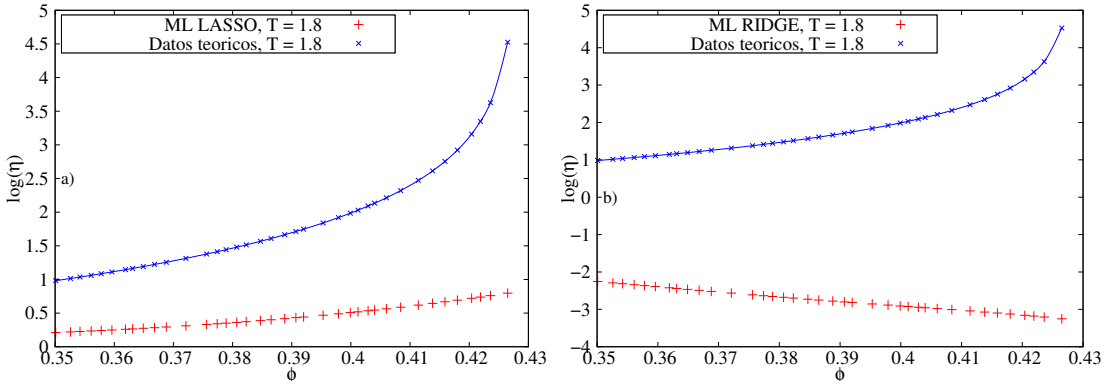


Figura 3.3: Resultados de la predicción de factores de estructura del sistema *SW* utilizando modelos entrenados con *HS*. A la izquierda (a) utilizando *LASSO* y a la derecha *RIDGE*. En azul se muestran los datos teóricos y en rojo las predicciones de los modelos

de modelos entrenados con algún sistema para predicciones en ese mismo sistema.

3.1.1. WCA

Como primer caso consideramos *WCA*. Para los datos que usaremos, reciclamos los anteriormente obtenidos, pero en esta ocasión divididos en tres conjuntos, tal que el de

entrenamiento cuenta con 254 ejemplos. En esta instancia queremos tener, en lugar de cantidad de ejemplos para el entrenamiento, presencia a lo largo de los rangos considerados. En consecuencia, el número de casos que tendremos para realizar el entrenamiento será considerablemente menor a entrenamientos anteriores. De nueva cuenta se utilizaron los mismos modelos de regresión mencionados con anterioridad. En la Fig. 3.4 vemos los resultados de ambos modelos para las predicciones con la base de datos de *WCA* utilizada en la sección 3.1. En este ejercicio, los modelos *LASSO* y *RIDGE* alcanzaron un *accuracy score* de 0.9699 y 0.9915, respectivamente.

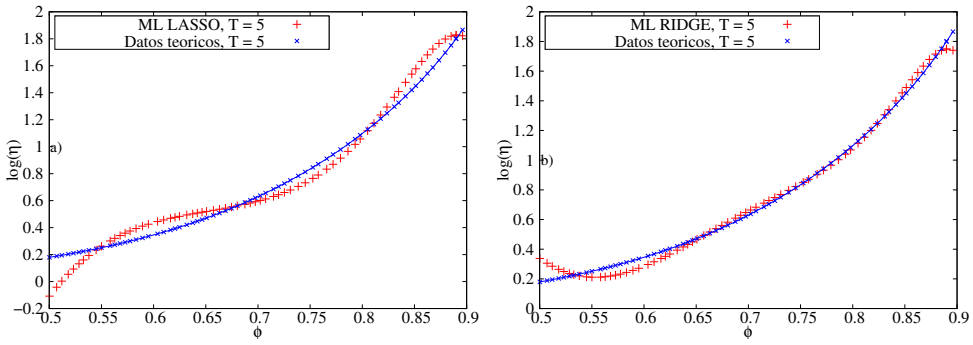


Figura 3.4: Resultados de predicción de viscosidad para el sistema *WCA* cuando se entrenaron los modelos con datos del mismo sistema *WCA*. A la izquierda (a) utilizando *LASSO* y a la derecha (b) *RIDGE*. En azul se presentan los datos teóricos y en rojo las predicciones.

Es evidente que, realizando el entrenamiento con datos de *WCA*, el comportamiento para datos de prueba por el modelo que siguen perteneciendo al sistema *WCA* se mejoró notablemente, siendo el valor para $accuracy = 0.96995$ para *LASSO* y $accuracy = 0.99154$ para *RIDGE*.

3.1.2. Square Well

Repetimos la prueba, en esta ocasicón para datos de *SW*. De nueva cuenta, hacemos uso de los datos previamente generados, dividiéndolos en esta ocasión en los conjuntos de entrenamiento, validación y prueba. Como en el caso de *WCA*, estamos interesados en tener datos a lo largo de los rangos de libertad ϕ y T, con el objetivo de tener representación en todos los lugares, sin importar la cantidad de datos. En consecuencia, contamos con menor cantidad de datos de entrenamiento. Como antes, utilizamos los mismos modelos (*LASSO* y *RIDGE*). Los resultados se pueden ver en la Fig. 3.5. Para este caso particular, dejamos fuera los casos que estuvieran por debajo de la línea espinodal , ya que se trata de estructuras las cuales contienen inconsistencias y no son de utilidad para los modelos. Los *accuracy score* alcanzados por los modelos *LASSO* y *RIDGE* fueron 0.9617 y 0.9921. En la Fig. 3.5 en azul vemos los valores teóricos de $log_{10}(\eta)$ y en rojo las predicciones de los modelos, por *LASSO* (a) y por *RIDGE* (b). Podemos ver que las predicciones para estructuras de SW de ambos modelos, al haber sido entrenados con datos del mismo sistema (SW), arrojaron resultados sobre la curva teórica, en particular, el modelo *RIDGE* obtuvo mejores resultados que *LASSO*.

3.2. Diagramas de arresto y viscosidad.

Dentro de las capacidades predictivas de la NESCGLE se encuentra la capacidad de calcular cantidades como la viscosidad como función del tiempo, así como, determinar si el sistema se encuentra arrestado o no. Para realizar los cálculos correspondientes, las computadoras toman cantidades considerables de tiempo. Con ayuda de los modelos entrenados presentados previamente es posible obtener resultados considerablemente similares a los teóricos en este respecto. No hay que perder de vista el hecho de que los

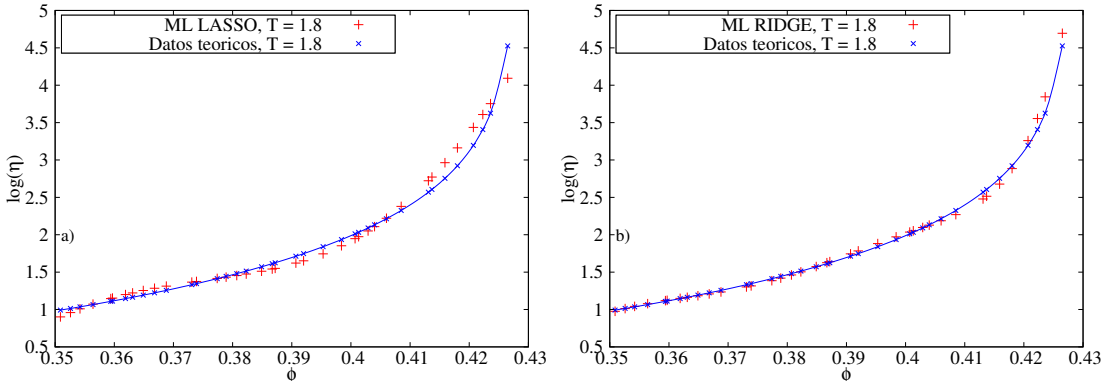


Figura 3.5: Resultados de predicción de viscosidad para el sistema SW cuando se entrenaron los modelos con datos del mismo sistema SW. A la izquierda (a) utilizando $LASSO$ y a la derecha (b) $RIDGE$. En azul se presentan los datos teóricos y en rojo las predicciones.

entrenamientos a nuestros modelos se han hecho con datos de la teoría en equilibrio, esto es importante ya que ahora lo pondremos a prueba con datos que dependen del tiempo, sin embargo,usaremos los casos asintóticos, que se corresponden con los casos de equilibrio.

En la Fig. 3.6, vemos el diagrama de arresto con valores de viscosidad correspondientes a los cálculos teóricos donde se utilizó la teoría NESCGLE [25]. Tenemos en el eje y a la temperatura en escala logarítmica, y en el eje x a la fracción de volumen ϕ, y en escala de colores se tiene la viscosidad η, que será una de las cantidades a las cuales haremos predicción así como a lo que refiere si se encuentra arrestado o no. Para obtener las predicciones, generamos datos de factores de estructura para WCA en los siguientes rangos para T y ϕ: $(10^{-5}, 10)$ y $(0.56, 0.7)$, que corresponden con los ejes de la Fig. 3.6. El próximo paso consiste en pasar por los modelos entrenados a todos los factores de estructura con los que contamos, se registran los resultados de clasificación y viscosidad y se grafican. El resultado lo podemos apreciar en la Fig. 3.7

Podemos apreciar una gran similitud entre las dos imágenes, siendo que en la parte

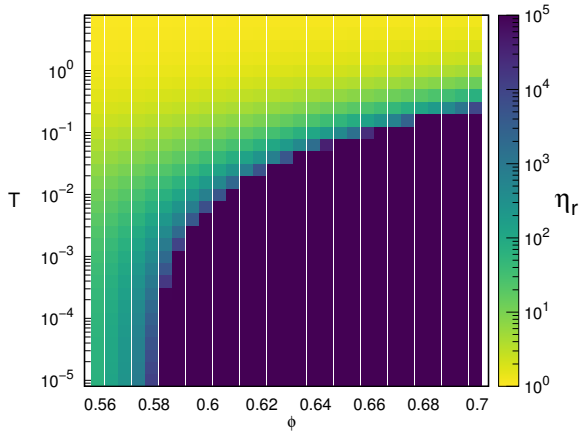


Figura 3.6: Diagrama de arresto (ϕ, T, η) donde la dimensión de viscosidad corresponde a la barra de colores. Resultados teóricos.

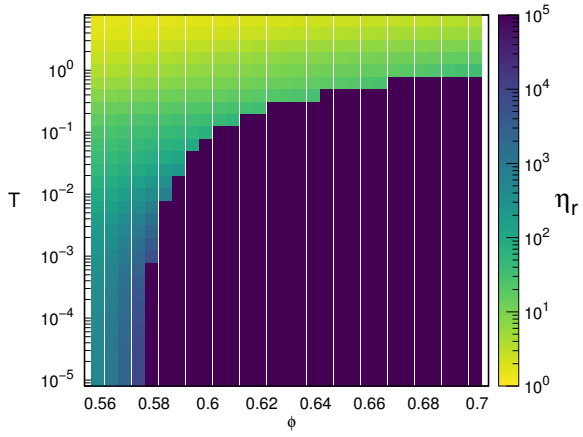


Figura 3.7: Resultados del diagrama de arresto con viscosidades, utilizando el clasificador de *SVC* y el modelo de regresión RIDGE, los cuales fueron entrenados previamente, como se explica en secciones anteriores.

donde se forman la frontera de arresto los valores predichos parecen estar por encima de donde se ubican los teóricos. Fuera de esta diferencia, los modelos han hecho un buen trabajo clasificando y asignando valores de viscosidad, incluso cuando fueron entrenados con datos de teorías de equilibrio.

Capítulo 4

Resultados Clasificación No-supervisada

En este capítulo, se hace un análisis utilizando modelos de entrenamiento no supervisado para realizar una clasificación de los módulos $G'(\omega)$ y $G''(\omega)$ de acuerdo a su comportamiento. Además se hace un análisis al compararlo con una clasificación manual. Finalmente, se presenta una mejora para el procedimiento de la clasificación supervisada, presentada en el capítulo 2, a partir de cantidades estadísticas del factor de estructura estático.

4.1. Aprendizaje No-supervisado.

Para explorar el alcance del aprendizaje no-supervisado (el cual discutimos brevemente en la sección 1.1.3), en esta sección, se presentan los resultados de una clasificación en regiones en el plano (ϕ, T) tales que los módulos dinámicos $G'(\omega)$ y $G''(\omega)$ se comporten de manera específica en cada una de esas regiones. Esta clasificación se presume ser autónoma

de cualquier intervención humana.

Los datos que fueron necesarios para esta parte consistieron en el cálculo numérico de las cantidades $G'(\omega)$ y $G''(\omega)$. Para ello se utilizaron scripts previamente desrrollados por el grupo de investigación. Se calcularon a partir de *quenches* donde se realizaron enfriamientos isobáricos de una temperatura inicial $T_i = 0.01$ a una temperatura final del rango $[0.5, 7.0]$, para diferentes valores de ϕ en el rango $[0.55, 0.9]$. A partir de $G'(\omega)$ y $G''(\omega)$ se obtuvo la cantidad $\tan(\delta)$, su definición es

$$\tan(\delta) = \frac{G'(\omega)}{G''(\omega)}. \tag{4.1}$$

Además, se consideraron como características de entrenamiento el número de veces que las curvas correspondientes a $G'(\omega)$ y $G''(\omega)$, en el plano $(\omega, G(\omega))$, tienen cruces entre ellas (x_G); además del número de veces que la curva $\tan(\delta)$, en el plano $(\omega, f(\omega))$, cruza por el 1 de las ordenadas (x_1). En la Fig. 4.1 a) podemos ver un caso particular cuando los módulos $G'(\omega)$ y $G''(\omega)$ se cruzan una vez, y por consiguiente, la curva $\tan(\delta)$ cruza una vez el 1 en Fig. 4.1 b).

Todos los tipos de datos mencionados anteriormente se puede ver agrupado "en crudo" (es decir, antes de realizar un procesamiento de datos) en la tabla 4.1. En la primer columna se encuentran los vectores ω, que son el mismo para todos los casos; en la columna 8 se encuentra la temperatura final del enfriamiento.

Con el objetivo de realizar una comparación a los resultados que se obtengan de los modelos, realizamos una clasificación "manual". Sabiendo que al cumplirse ciertas condiciones, algún caso pertenecerá a una región o a otra. Esas condiciones se corresponden con el comportamiento de las curvas $G'(\omega)$ y $G''(\omega)$. El resultado se ve en la Fig. 4.2.

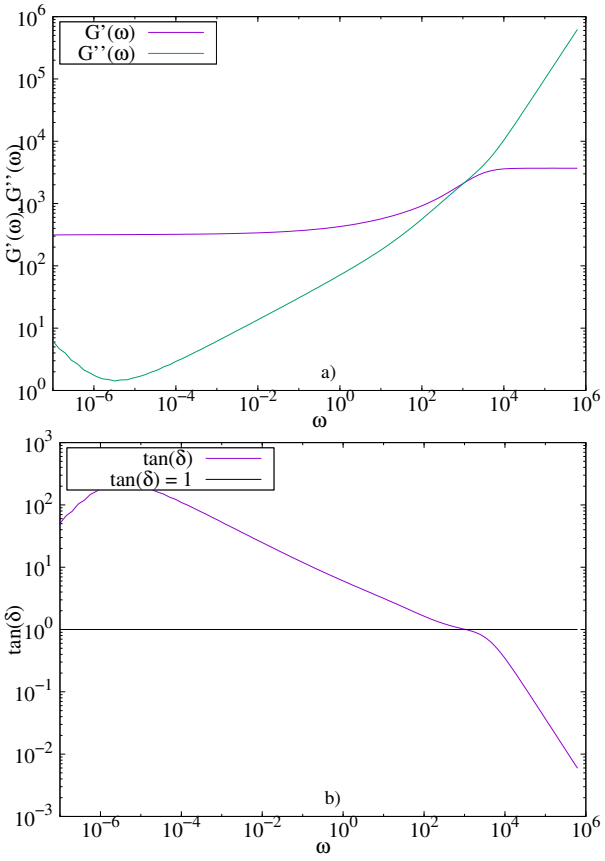


Figura 4.1: . En a) los módulos dinámicos $G'(\omega)$ y $G''(\omega)$, en este caso particular se cruzan entre ellos una vez. En b) la curva $\tan(d)$, corresponde a los módulos dinámicos en a) y cruza una vez por 1.

Para la entrada de datos al modelo que utilizaremos (*K-means clustering*) se escogieron como caracteísticas el arreglo $\tan(\delta)$, x_1 y x_G. Podemos ver una parte en la tabla 4.2. Sin embargo, antes de poder alimentar al modelo, es necesario realizar una serie de transformaciones que beneficiarán los resultados. Primero, se realiza un escalado para los datos, tal que conservemos las relaciones entre ellos obteniendo rangos manejables para los cálculos que realizan los algoritmos. Para llevar a cabo este escalado, utilizamos un escalador ya implementado en la paquetería *Scikit-Learn*. En seguida, los datos escalados

ω	$G'(\omega)$	$G''(\omega)$	$\tan(\delta)$	x_1	x_G	ϕ	T_f
ω_1	$G'_1(\omega_1)$	$G''_1(\omega_1)$	$\tan(\delta_1)$	0	0	0.55	0.5
ω_2	$G'_2(\omega_2)$	$G''_2(\omega_2)$	$\tan(\delta_2)$	0	0	0.6	0.5
ω_3	$G'_3(\omega_3)$	$G''_3(\omega_3)$	$\tan(\delta_3)$	2	2	0.65	0.5
ω_4	$G'_4(\omega_4)$	$G''_4(\omega_4)$	$\tan(\delta_4)$	2	2	0.7	0.5
ω_5	$G'_5(\omega_5)$	$G''_5(\omega_5)$	$\tan(\delta_5)$	1	1	0.75	0.5

Cuadro 4.1: Primeras 5 filas de la base de datos en crudo para el entrenamiento no supervisado de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$. Las columnas 2, 3 y 4 contienen el arreglo de datos correspondiente que representamos con el símbolo $G'_N(\omega_1)$, $G''_N(\omega_1)$, $\tan(\delta_N)$, según sea el caso.

$\tan_0(\delta)$	$\tan_1(\delta)$	$\tan_2(\delta)$	...	$\tan_8 4(\delta)$	$\tan_8 5(\delta)$	x_1	x_G
8.20879e-10	1.1609e-09	1.64176e-09	...	0.000512797	0.000362377	0	0
3.15285e-09	4.4588e-09	6.30569e-09	...	0.00130591	0.000922843	0	0
6.80796e-09	9.62791e-09	1.36159e-08	...	0.00285909	0.00202044	0	0
7.63395	10.0088	12.8079	...	0.024489	0.0173074	3	3
4.40902e-09	6.23529e-09	8.81804e-09	...	0.00172714	0.00122052	0	0

Cuadro 4.2: Primeras 5 filas de la base de datos conformada por $tan(\delta)$ sin procesamiento dimensional.

son procesados para reducir su dimensionalidad, esto es, a la cantidad total de características, la cual la podemos ver en la tabla 4.2, se dice reducir su dimensionalidad al proceso de reducir el total de características, manteniendo la información contenida lo más intacta posible. Hacer esto no sólo nos ayuda para obtener resultados con mayor rapidez, sino que nos ayuda a visualizar los *clusters* que se determinarán en el entrenamiento no supervisado.

De igual manera, utilizamos un objeto desarrollado por *Scikit-Learn* basado en el algoritmo *PCA* (que se discute en el apéndice A.2). Como primer paso, inicializamos el objeto sin especificar el número de dimensiones objetivo a reducir, esto se hace para,

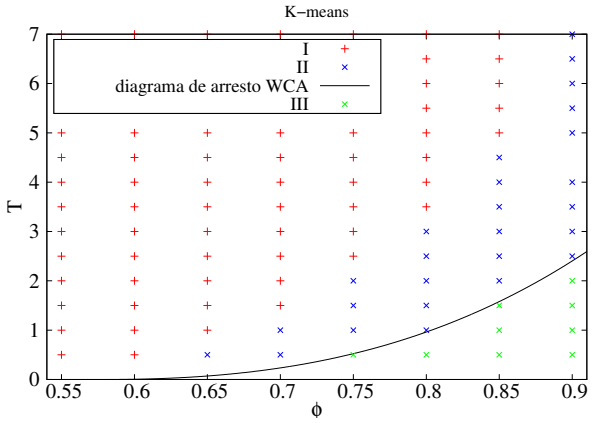


Figura 4.2: Para comparar los resultados del modelo *K-means*, se realizó una clasificación manual de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$. En un script en Python se programaron las condiciones siguientes: si el número de veces que los módulos dinámicos se cruzan entre sí es 2 entonces pertenece a la región III, si se cruzan 1 vez, entonces pertenece a la región II; si ni hay cruces, entonces pertenece a I.

precisamente, determinar ése número de dimensiones. Obtenemos una gráfica como la de la Fig. 4.3. En el eje y se aprecia la cantidad de información, donde 1.0 significa que para un determinado número de dimensiones (eje x) la información total contenida en nuestros datos escalados no se altera. De esta forma, determinamos que el número de dimensiones ideal será de 9. En seguida, se inicializa de nuevo un objeto *PCA* indicándole el número de dimensiones que requerimos. Una vez suministramos los datos escalados, obtenemos nuestros datos con 9 dimensiones, o dicho de otra forma, 9 características, y serán estos datos los que finalmente pasen al modelo final.

4.1.1. K-means.

Los detalles de este algoritmo se discuten en la sección 1.1.3. En el momento de inicializar el modelo, especificamos el número de *clusters* k, en el que se dividirán los datos. Al usar *k-means* es necesario saber de antemano el número de regiones con los que

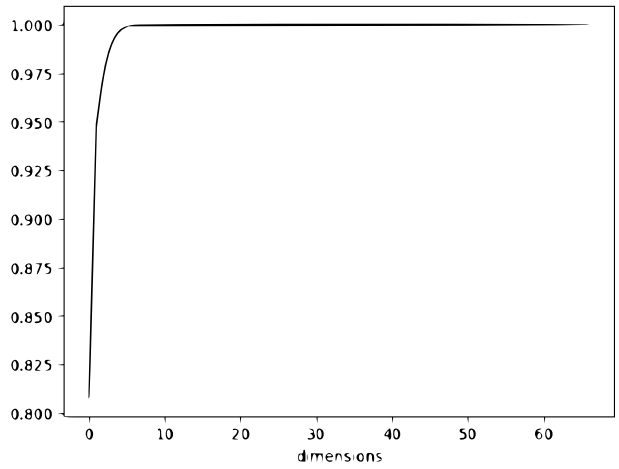


Figura 4.3: Gráfica sobre la cantidad de "información" contenida en los datos, el número mayor de dimensiones corresponde a las características con las que contamos originalmente tabla 4.2. Según ese número disminuye lo hace a su vez el número de características. Podemos ver que cuando se reduce de 87 características a 9 la cantidad de "información" se mantiene en 1.0.

se cuenta. Sólo resta entrenar el modelo y realizar predicciones sobre sus clasificaciones a los datos. Los resultados se presentan en la Fig. 4.4. Si comparamos las Figs. 4.2 y 4.4 podemos apreciar que el modelo hizo un trabajo excelente para hacer predicciones a los datos en los tres conjuntos de entrenamiento, validación y prueba. Entonces podemos decir que contamos con un método de clasificación no supervisado que funciona de manera eficiente, siempre y cuando sepamos el número de regiones. El próximo paso a tomar en esta dirección sería encontrar un método para clasificar conjuntos de datos de los cuáles no sabemos con exactitud el número de grupos que puede formar.

4.2. Un método diferente

En esta sección, planteamos y discutimos los resultados de implementar un nuevo método para la clasificación de factores de estructura, $S(k)$, tanto provenientes del sistema

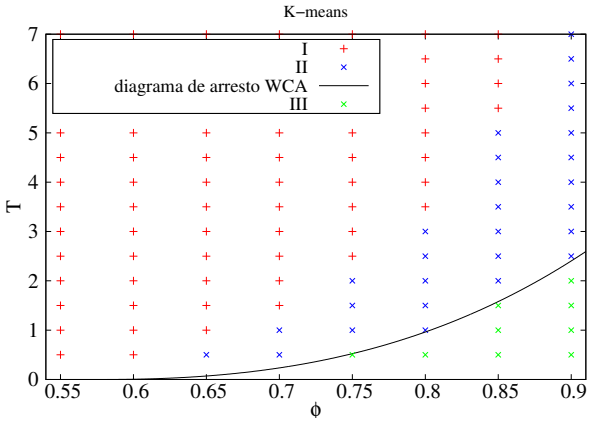


Figura 4.4: Resultados clasificación no supervisada de los módulos dinámicos $G'(\omega)$ y $G''(\omega)$, usando un modelo de *K-means*.

HS como del sistema *WCA*. A partir de los factores de estructura que utilizamos en el entrenamiento de la sección 2.1 para los modelos de *Logistic Regression* y *SVC* podemos proponer el siguiente tratamiento para obtener un número mucho menor de características de entrenamiento. A partir de los valores S_1, S_2, ... S_k que conforman el vector $S(k)$ calculamos el promedio, la desviación estándar, el skweness, kurtosis, el mínimo y el máximo, que son cantidades estadísticas bien conocidas. Con estas junto a ϕ obtenemos finalmente nuestro nuevo conjunto de características, siendo que en el ejercicio anterior (sección 2.1 tabla 2.1) se tenían 440, y ahora sólo 7. El próximo paso es entrenar un nuevo modelo (lo hacemos con *LR* y *SVC*, los mismos que hemos utilizado) con datos del sistema *HS*, el más sencillo, y evaluarlo con datos nuevos, ya sea del mismo sistema o de algún otro sistema.

El nuevo formato de entrada para los modelos luce como en la tabla 4.3. Así como ocurrió en la sección 2.1, las predicciones hechas para datos del sistema en el que fueron entrenados (en este caso *HS*) arrojaron excelentes resultados. Como era de esperar, el

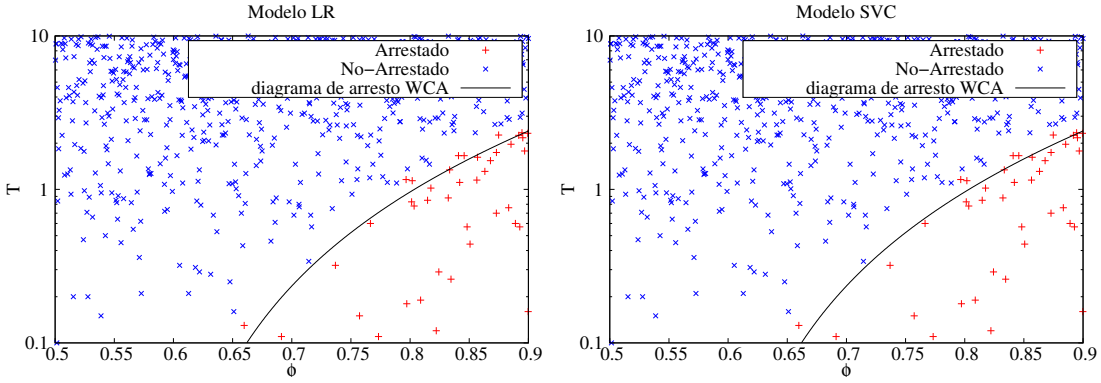


Figura 4.5: Resultados de clasificación de estados arrestados (rojo) y no arrestados (azul) por los modelos *LR* a la izquierda y *SVC* a la derecha. Utilizando como características de entrenamiento las presentadas en la tabla 4.3.

valor de accuracy que alcanzaron los modelos fue de 0.995. Así que una vez entrenados hay ponerlos a prueba con datos de otro sistema (*WCA*). Para ello utilizamos los datos de aquella misma sección, sólo que en esta instancia recibirán un tratamiento diferente, tal como los datos de la tabla 4.3. Los resultados se ven en la Fig. 4.5.

ϕ	mean	std	skewness	kurtosis	min	max
0.2351	0.938209	0.222422	-2.507937	5.7115631	0.0	1.311885
0.3170	0.926096	0.270964	-1.938172	4.036098	0.0	1.556252
0.4222	0.914175	0.339819	-0.817726	3.229412	0.0	2.131564
0.4382	0.912637	0.352544	-0.589593	0.3.414201	0.0	2.267477
0.5698	0.902087	0.515023	2.143044	14.047307	0.0	4.422049

Cuadro 4.3: Primeras 5 filas de la base de datos con sólo 7 características de entrenamiento. Utilizando cantidades estadísticas del factor de estructura.

Es evidente que cuando utilizamos este nuevo formato de 7 características los resultados muestran mejor rendimiento respecto de cuando proporcionábamos sólo el factor de estructura. Se observa que ambos modelos, LR y SVC, arrojaron resultados que parecen ser idénticos. De la figura 4.5, vemos que en la zona donde se dibuja la línea de

arresto teórica existen algunos casos arrestados que están por encima, sin embargo no hay ninguno no-arrestado que esté por debajo. Además, logramos eliminar la protuberancia de casos arrestados que se formaba entre $\phi = 0.65$ y $\phi = 0.75$. Otra de las ventajas de reducir la "dimensionalidad" se refiere a la capacidad de visualizar los datos en un espacio diferente de características. Por ejemplo, en la Fig. 4.6 se muestra el gráfico de las características "kurtosis" vs. "mean" (a), "kurtosis" vs "ϕ" (b) y "std" vs "mean" (c). Para cada una de estas gráficas los puntos rojos representan (como ha sido hasta ahora) factores de estructura arrestados y los puntos azules los no arrestados. Podemos apreciar que en estos espacios se forma una frontera lineal entre ambos estados. Por ejemplo, para el caso de la Fig. 4.6(a) para casos ,$S(k)$, que la kurtosis sea mayor que ≈ 15, se corresponderán con un estado arrestado. En la Fig. 4.6(c), podemos intuir una frontera, entre casos arrestados y no arrestados, como una recta con pendiente negativa para el espacio $(mean, std)$. Por otra parte, se puede reducir la base de datos de 7 dimensiones a una de sólo dos, utilizando el algoritmo de PCA, y graficar las relaciones en este caso. Esta reducción adicional nos forma dos nuevas características, por conveniencia, llamaremos a esas nuevas características x_2 y x_1. El resultado se ve en la Fig. 4.7. Como podemos ver, en este espacio de nuevas características (x_2, x_1), los casos no arrestados (en azul) se encuentran agrupados entre $x_2 \approx 0$ y $x_2 \approx 15$, en cambio, los casos arrestados se dispersan unos de otros, en especial, cuando x_2 está en valores más a la derecha.

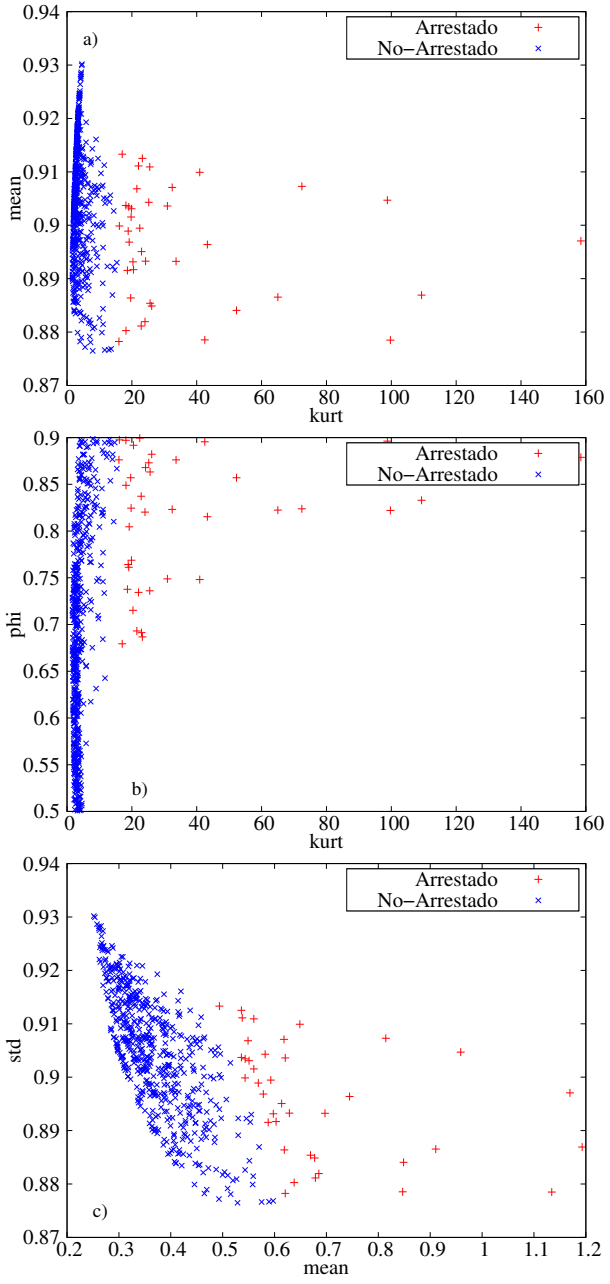


Figura 4.6: Gráficas para diferentes espacios de características, a saber: "kurtosis vs mean" (a); "kurtosis vs ϕ" (b); y "std vs mean" (b). En azul se muestran casos clasificados como no-arrestados, en rojo los arrestados.

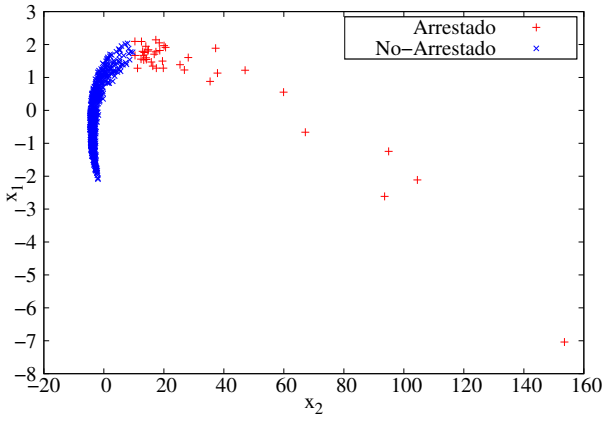


Figura 4.7: Gráfica donde se muestra el resultado de reducir la dimensionalidad de las características de 9 a 2, las nuevas características se nombran "x_1" y "x_2".

Capítulo 5

Conclusiones y Perspectivas

En este capítulo, se presentan las conclusiones generales del presente trabajo de tesis. El reto principal al que nos enfrentamos en este trabajo fue el de aprender a usar las herramientas de Machine Learning desde la completa ignorancia, ya que el grupo de trabajo no había utilizado antes este tipo de herramientas. Recordemos que el objetivo principal de este trabajo era introducir herramientas de Machine Learning en el manejo y análisis de datos provenientes de la teoría SCGLE. Sobre este respecto, concluimos que, tomando en cuenta las herramientas que fueron utilizadas, su uso es plausible en combinación con el tipo de datos que se obtienen a partir de la teoría SCGLE. En la mayoría de los casos, los resultados de precisión en predicciones fueron mayores a 0.9. Por otro lado, concluimos que, gracias al origen de los datos con los que trabajamos, los modelos de Machine Learning alcanzaron valores óptimos a sus parámetros en poco tiempo y con pocos datos de entrenamiento, debido a que los datos vienen de una teoría bien establecida. Comúnmente, las herramientas de Machine Learning se usan para datos de fuentes aleatorias, en estos casos los modelos alcanzan precisiones no mayores a 0.85 y con cantidades del orden de miles de datos. Con los nuestros alcanzamos precisiones

mayores a 0.9. Podemos concluir también sobre el modelo con más potencial útil a futuro para el grupo de trabajo. Esto es, el modelo de aprendizaje no-supervisado, K-means. Como vimos en el capítulo 4, nuestro modelo de K-means separó el conjunto de datos de $G'(\omega)$ y $G''(\omega)$ en sus respectivos clusters en el espacio termodinámico $\phi - T$, clusters que sabíamos eran 3. Para sistemas diferentes al de WCA, por ejemplo para el sistema SALR, seremos capaces de determinar el número de clusters que se formarán para este o algún otro caso particular.

Como mencionamos, los datos con los que trabajamos corresponden a los generados con la teoría SCGLE. Como siguiente paso, se propone empezar a utilizar datos generados con la teoría NESCGLE. En primer lugar enfocándonos en replicar los resultados obtenidos con datos para sistemas en equilibrio, ahora con datos de no equilibrio a tiempos asintóticos. Una vez llevada a cabo esta prueba, seguirá utilizar datos de no equilibrio a diferentes tiempos de espera, tal que podamos poner a prueba los modelos de Machine Learning en predicciones complejas de procesos estocásticos.

Hasta ahora hemos utilizado un método en el que para trabajar con las herramientas de Scikit-Learn generamos una base de datos local, es decir que se almacena en la memoria de la computadora del usuario. Esta manera de trabajar los datos representa un riesgo de saturar la memoria local del usuario. Por lo tanto, conviene utilizar plataformas que ofrezcan mayor capacidad de almacenamiento y mejor manejo de datos. Afortunadamente, existe una plataforma que nos permite, precisamente, guardar y manejar grandes cantidades de datos. Nos referimos a *Google Cloud Platform*, donde además de permitirnos el almacenamiento en tablas al estilo de SQL, encontramos herramientas como *compute engine* que nos permite utilizar máquinas virtuales; *BigQuery* que es la forma en que se nombra el almacén de datos. De la misma manera, también nos brinda herramientas de Inteligencia Artificial y Machine Learning, como la denominada *VertexAI*, que es una

plataforma dedicada para todo el ciclo de aprendizaje, también nos ofrece la capacidad de utilizar *Gemini*, que es una *Large Language Model* (LLM) desde el código mismo. De esta manera, podemos incorporar los procedimientos donde utilizamos Scikit-Learn que se han desarrollado durante el presente trabajo de tesis, a la plataforma de Google Cloud, aprovechando VertexAI y la capacidad de almacenamiento y consulta de datos.Con esto, pretendemos mejorar el uso de Inteligencia Artificial en conjunto con la teoría NESCGLE.

Apéndice A

Algoritmos.

A.1. Gradient Descent

Este algoritmo es ampliamente utilizado en una gran variedad de problemas que requieren una optimización de parámetros. La idea general es ajustar los parámetros iterativamente para minimizar una función de error [8]. El funcionamiento de *Gradient Descent* es el siguiente. Se tiene el conjunto de características de entrada $\vec{x}$, que a partir de una función hipótesis, $h_{\vec{\theta}}$, se obtiene un resultado deseable y. La función hipótesis tiene un conjunto de parámetros ajustables $\vec{\theta}$. Asumimos que $h_{\vec{\theta}}$ es diferenciable respecto a $\vec{\theta}$. Debido a que inicialmente los parámetros ajustables pueden no ser los óptimos se determina una función de error, la cual la denotaremos como $Err(\vec{x}, \vec{y}; \vec{\theta})$, que nos señanala qué tan alejado está el resultado arrojado a partir del vector de características con el valor deseable y. Entonces *Gradiant Descent* realiza el siguiente cálculo:

$$\vec{\theta}_{t+1} = \vec{\theta}_t - \alpha_t \nabla_{\vec{\theta}} Err(\vec{x}, \vec{y}; \vec{\theta}) \tag{A.1}$$

En esta ecuación $\vec{\theta}_{t+1}$ son los parámetros modificados; α_t es una constante que puede

depender de t. En el caso más simple, α_t no depende de t, y α es sólo una constante, A.1 nos indica que el vector $\vec{\theta_{t+1}}$ al tiempo $t+1$ será el resultado de substraer de $\vec{\theta}$, α veces la pendiente de la función de error. A la constante α se le conoce como *learning rate*, y controla la sensibilidad con que se modifica el vector $\vec{\theta_{t+1}}$.

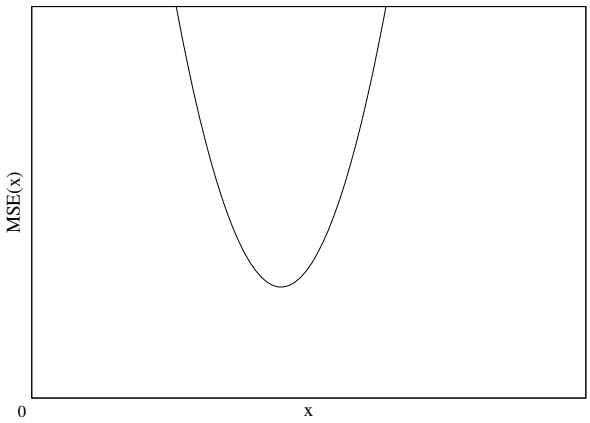


Figura A.1: Perfil típico de la función *Mean Square Error* ($MSE(x)$). Gradient Descent busca alcanzar el mínimo de ésta función modificando el vector de parámetros $\vec{\theta}$ hasta alcanzar el vector óptimo.

Este proceso se realiza iterativamente hasta que el gradiente de la función de error sea 0, con lo que se asegura un mínimo; o hasta que alcance una tolerancia ϵ.

A.2. PCA

PCA es un algoritmo que se utiliza para reducir la dimensionalidad de un conjunto de datos D. PCA (Principal component analysis) analiza datos contenidos en una tabla con variables dependeintes entre sí. Su objetivo principal es el de extraer la información más importante contenida en los datos y expresar dicha información a partir de un conjunto de nuevas variables ortogonales llamadas "componentes principales" (PC) [33] [34], de esta forma, se simplifica la descripción del conjunto. Las componentes principales son

combinaciones lineales de las variables originales. Existe una técnica estándar para obtener las componentes principales de un conjunto, se conoce como *Singular Value Decomposition* (SVD) [8], el cual consiste en descomponer al conjunto D en tres matrices

$$D = U\Sigma V^{\top}, \tag{A.2}$$

donde, V contiene los vectores columna que definen todas las componentes principales. A partir de V se construye la matriz W_d utilizando las primeras d columnas de V (siendo d el número de dimensiones que se desea). Finalmente,

$$D_d = DW_d, \tag{A.3}$$

donde D_d es el nuevo conjunto con dimensiones reducidas a d. El framework scikit-learn, que usamos en este trabajo, cuenta con un objeto denominado PCA() que realiza este procedimiento. El lector interesado puede referirise a [34].

A.3. Standard Scaler

Es un algoritmo sencillo que escala los valores de un conjunto de datos, D, de acuerdo a una unidad de varianza, en este respecto

$$z = \frac{x - v}{s}, \tag{A.4}$$

donde z es el valor escalado, x es el valor antes de ser escalado, v es el valor medio de los datos y s es la desviación estándar [35].

A.4. Accuracy score

Clasificación

Accuracy score es una operación que realiza scikit-learn para determinar la cantidad de predicciones correctas, donde 1.0 es el resultado perfecto, la expresión es la siguiente,

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i), \tag{A.5}$$

donde, $\hat{y}_i$ es el valor i-ésimo (o clase) predicha y y_i es el valor real (o teórico). $1(x)$ es la función indicador [36], este método se utiliza para los modelos de clasificación.

Regresión

Para el case de regresión, se utiliza la siguiente expresión para determinar el *accuracy score*,

$$accuracy(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{A.6}$$

donde $\hat{y}_i$ es el valor predicho, y_i es el valor real (o teórico) correspondiente y $\bar{y}$ es el valor medio de y [36].

Apéndice B

Scripts.

El total de programas escritos por el autor de este trabajo de tesis, ya sea como un jupyter notebook o un simple programa en python, se encuentran reunidos en un repositorio de la plataforma GitHub, el lector interesado puede encontrar todos los programas en la referencia [37].

Para la recolección de datos se utilizó, como se mencionó anteriormente, el repositorio *NESCGLE* que, de igual forma, se encuentra en la plataforma GitHub, el lector interesado puede encontrar la referencia en [38].

Viscosidad

Se utilizó el siguiente script para obtener los valores de viscosidad correspondientes a algún factor de estructura. Se trata de una función que se añade a los programas correspondientes del repositorio [38].

```
1 def eta(tau, DG):
2     int = 0
3     for i in range(2:len(tau)):
```

```
        dx = tau[i] - tau[i-1]
        int += dx*DG[i]
    eta = 3*pi*int
    return eta
```

Listing B.1: Función para determinar la viscosidad a partir de la teoría SCGLE.

Bibliografía

[1] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodriguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, xxx(xxxx):xxx, 2020.

[2] Emanuele Boattini, Marjolein Dijkstra, and Laura Filion. Unsupervised learning for local structure detection in colloidal systems. *The Journal of Chemical Physics*, 151(15):154901, 10 2019.

[3] Ulices Que-Salinas, Pedro E. Ramírez-González, and Alexis Torres-Carbajal. Determination of thermodynamic state variables of liquids from their microscopic structures using an artificial neural network. *Soft Matter*, 17:1975–1984, 2021.

[4] Amun Jarzembski, Zachary T. Piontkowski, Wyatt Hodges, Matthew Bahr, Anthony McDonald, William Delmas, Greg W. Pickrell, and Luke Yates. Rapid subsurface analysis of frequency-domain thermoreflectance images with k-means clustering. *Journal of Applied Physics*, 135(16):165102, 04 2024.

[5] Advanced information. Nobelprize.org. `https://www.nobelprize.org/prizes/physics/2024/advanced-information/`. Fri. 21 Mar 2025.

[6] Popular information. Nobelprize.org. `https://www.nobelprize.org/prizes/physics/2024/popular-information/`. Fri. 21 Mar 2025.

[7] Laurent Sindayigaya. Machine learning algorithms: A review. *Information Systems Journal*, 2022. Article.

[8] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2017.

[9] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, 3rd edition, 1998.

[10] Gülden Kaya Uyanık and Neşe Güler. A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, 106:234–240, 2013. 4th International Conference on New Horizons in Education.

[11] Hongshuo Huang and Amir Barati Farimani. Multimodal learning of heat capacity based on transformers and crystallography pretraining. *Journal of Applied Physics*, 135(16):165104, 04 2024.

[12] Bradley C. Dallin, Atharva S. Kelkar, and Reid C. Van Lehn. Structural features of interfacial water predict the hydrophobicity of chemically heterogeneous surfaces. *Chemical Science*, 14:1308–1319, 2023.

[13] Lijin Zhang, Xiayan Wei, Jiaqi Lu, and Junhao Pan. Lasso regression: From explanation to prediction. *Advances in Psychological Science*, 28(10):1777–1788, 2020.

[14] Emilio Carrizosa and Dolores Romero Morales. Supervised classification and mathematical optimization. *Computers Operations Research*, 40(1):150–165, 2013.

[15] Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Advanced lectures on machine learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *ML Summer Schools 2003*, volume 3176 of *Lecture Notes in Artificial Intelligence*, pages 1–241, Berlin, Heidelberg, 2004. Springer. Revised Lectures from the ML Summer Schools 2003 in Canberra, Australia and Tübingen, Germany.

[16] Dakhaz Mustafa Abdullah and Adnan Mohsin Abdulazeez. Machine learning applications based on svm classification: A review. *Qubahan Academic Journal*.

[17] Charles Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 147–153, Washington DC, 2003.

[18] Shi Na, Liu Xumin, and Guan Yong. An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 63–67. IEEE, 2010.

[19] Herbert B. Callen. *Thermodynamics and an Introduction to Thermostatistics*. John Wiley & Sons, New York, 2nd edition, 1985.

[20] Leopoldo García-Colín Scherer. *Termodinámica: Teoría cinética y termodinámica estadística*. Sección de Obras de Ciencia y Tecnología. Fondo de Cultura Económica, México, 2006.

[21] Donald A. McQuarrie. *Statistical Mechanics*. University Science Books, Sausalito, CA, 2000. Revised edition.

[22] Luis Enrique Sanchez-Diaz, Pedro Ramirez-Gonzalez, and Magdaleno Medina-Noyola. Equilibration and aging of dense soft-sphere glass-forming liquids. *Physical Review E*, 87(5):052306, May 2013.

[23] Jesús Benigno Zepeda-López and Magdaleno Medina-Noyola. Waiting-time dependent non-equilibrium phase diagram of simple glass- and gel-forming liquids. *The Journal of Chemical Physics*, 154(17):174901, 2021.

[24] Jean-Pierre Hansen and Ian R. McDonald. *Theory of Simple Liquids*. Academic Press, London, 3 edition, 2013.

[25] R. Peredo-Ortiz, O. Joaquín-Jaime, L. López-Flores, M. Medina-Noyola, and L. F. Elizondo-Aguilera. Nonequilibrium theory of the linear viscoelasticity of glass and gel forming liquids. *Journal of Rheology*, 69(1):201–222, 2025.

[26] Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Dover Publications, Inc., new dover edition, first published in 1956, is an unabridged and unaltered republication of the translation first published in 1926 edition, 1956. edited with notes by R. Fürth; translated by A. D. Cowper.

[27] Don S. Lemons and Anthony Gythiel. Paul langevin's 1908 paper .ºn the theory of brownian motion"["sur la théorie du mouvement brownien,ç. r. acad. sci. (paris) 146, 530–533 (1908)]. *American Journal of Physics*, 65(11):1079–1081, 1997.

[28] Lars Onsager and S. Machlup. Fluctuations and irreversible processes. *Physical Review*, 91(6):1505–1512, 1953.

[29] S. Machlup and Lars Onsager. Fluctuations and irreversible processes. II. systems with kinetic energy. *Physical Review*, 91(6):1512–1515, 1953.

[30] M. Medina-Noyola and J.L. Del Rio-Correa. The fluctuation–dissipation theorem for non-markov processes and their contractions: The role of the stationarity condition. *Physica A: Statistical Mechanics and its Applications*, 146(3):483–505, 1987.

[31] Laura Yeomans-Reyna and Magdaleno Medina-Noyola. Self-consistent generalized langevin equation for colloid dynamics. *Phys. Rev. E*, 64:066114, Nov 2001.

[32] Alauddin Ahmed and Richard J. Sadus. Phase diagram of the weeks-chandler-andersen potential from very low to high temperatures and pressures. *Phys. Rev. E*, 80:061101, Dec 2009.

[33] Markus Ringnér. What is principal component analysis? *Nature Biotechnology*, 26(3):303–304, mar 2008.

[34] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, jul 2010.

[35] Scikit learn Developers. sklearn.preprocessing.standardscaler, 2023.

[36] Scikit learn Developers. Model evaluation: quantifying the quality of predictions, 2023.

[37] José Angel Sánchez. scgle-ml: Machine learning tools for scgle. `https://github.com/jas-sanchez/scgle-ML`, 2025. Accedido el 9 de abril de 2025.

[38] Riperedo. Nescgle.jl: Julia library for simulating the game of life on nes. `https://github.com/Riperedo/NESCGLE.jl.git`. Accessed: 2024-05-02.