Universitat Politècnica de Catalunya

APA

Grado en Ingeniería Informática

Estudio sobre temperatura crítica en materiales superconductores

Autores:
Víctor Giménez Ábalos,
Guillem Ferrer Nicolas

Profesor:
Luis Antonio Belanche

Q1 Curso 2018/2019

Índice

1. Descripción del trabajo y objetivos

Los materiales superconductores son aquellos cuyas cualidades físicas permiten alcanzar resistencia eléctrica 0 a baja temperatura. Cada material (ya sea aleación o compuesto) tiene un umbral de superconductividad, normalmente expresado en grados Kelvin, tal que si el material se encuentra a temperatura inferior a éste, el material no ofrecerá resistencia al paso de corriente eléctrica. Este valor se denomina "temperatura crítica". [2]

Las propiedades que llevan a un material a ser superconductor todavía no han sido totalmente precisadas por modelos físicos, si bien ha habido modelos bastante certeros que datan de los años 50 (Ginzburg & Landau, 1950 y Bardeen, Cooper & Schrieffer, 1957). En concreto, la teoría central actual es la aportada por BCS, también llamada "Teoría de pares de Cooper", que ganó el Nobel en 1972.

Sin embargo, generalizaciones de ésta pueden ser controvertidas cuando se aplican a superconductores poco convencionales, como por ejemplo el oxipnictido de samario, cuya temperatura crítica real es superior a la predicha por BCS.[3]

Nuestro objetivo es encontrar un modelo (preferiblemente sencillo) para predecir la temperatura crítica a partir de los datos que tengamos disponibles o podamos encontrar sobre un material concreto. Esto nos podría permitir predecir la temperatura crítica de un superconductor hipotético aún no creado para ahorrar recursos científicos en su creación, entender mejor la importancia de las cualidades de un material para su superconductividad o, en el mejor de los casos, obtener datos sobre los cuales mejorar la precisión de la Teoría de pares de Cooper [4].

La información que hemos encontrado para ello está derivada de parejas de compuesto químico (en formato texto) y temperatura crítica obtenidas empíricamente. Hablaremos de la derivación en la próxima sección.

2. Artículos y estudios previos

Nuestro trabajo se basa en un artículo previo realizado por Kam Hamidieh, de la universidad de Pennsylvania[1].

Para poder tratar con los pares experimentales, Hamadieh consiguió los datos que usaremos en el estudio (masa atómica, primera energía de ionización...) a partir de los valores individuales de cada elemento que forma el compuesto. En concreto, a partir de las proporciones del material en el compuesto y de los valores concretos de cada elemento para la variable a estudiar, obtiene valores intermedios (uno por elemento), de los cuales puede extraer la media, la media ponderada según proporción (tanto para media aritmética como geométrica), entropía, desviación estándar, rango etc.

Este procedimiento aplicado a cada variable y para cada superconductor de los cuales tenemos datos de temperatura crítica nos da una tabla de 82 variables (80 sacadas mediante el proceso anterior, temperatura crítica y número de elementos que forman el compuesto).

A partir de estos datos, Hamadieh aplica dos métodos de regresión: regresión linear múltiple y XGBoost.

Regresión linear múltiple le permite alcanzar un R^2 de 0.74 y entorno a 17.6 K de error RMSE

para la muestra OOB.

XGBoost, al ser más sofisticado, le permite alcanzar R^2 0.92 y RMSE de 9.5 K para la muestra OOB tras optimizar los hiper-parámetros del método.

3. Exploración de los datos

Debido a la naturaleza del origen de nuestros datos, el apartado de preprocesado de éstos es bastante sencillo. Por un lado, no tenemos NAs, ni valores incoherentes o anómalos. Además, no nos interesa tratar outliers, ya que al contrario que en otros estudios donde pueden ser errores que distorsionan los datos, precisamente encontrar materiales poco convencionales y predecirlos correctamente es vital para nuestro estudio.

Los datos son todos numéricos de por sí y no requerirán de *one-hot-encoding* ni ningún otro método similar. Además, no hemos encontrado ninguna forma lógica de extraer más datos que los que se encuentran en el dataset, así que no hacemos extracción de atributos.

Inicialmente, nos propusimos tratar con ambos datasets, tanto el de valores físicos como el de proporciones químicas. Sin embargo, después de realizar unos primeros test, nos percatamos que debido a su distribución (mayoritariamente todos los valores químicos valen 0), nos distorsionaban muchísimo los datos (hasta llegar a 30000 de NMSE). Entendemos que esto se debe a que lo que aprendían las técnicas como SVM era que, si hay un elemento concreto, había que darle más valor a la temperatura crítica, pero como no todos los elementos estarán en el fragmento de entrenamiento, para cada elemento nuevo los modelos fallarán miserablemente. Igualmente, adjuntamos el código original al proyecto como *preprocessing_old.Rmd*.

En consecuencia, nuestro preprocesado termina en un simple paso de transformación logarítmica y estandarización.

- Examinamos las distribuciones de cada atributo físico y nos percatamos de que alguno podría requerir de transformación logarítmica.
- Estandarizamos datos.

En más detalle, cuando hablamos de decidir si aplicamos una transformación logarítmica nos referimos a que, en examinar el histograma de la variable, hay algunas que parecen asimétricas (desplazadas a la izquierda, específicamente) y leptocúrticas. Nos aseguramos de qué variables deberían ser transformadas mediante un filtrado según si ambas curtosis y asimetría estadística (*Skewness*) se encuentran por encima de 2. A las variables que les suceda esto les aplicaremos un logaritmo, pero para asegurar que no haya errores matemáticos si el mínimo de la variable es 0, comprobaremos esto y le aplicaremos un $log(x + 1)$ en caso de que lo sea.

Las variables que requieren de logaritmos son las siguientes:

```
log: "mean_Density","wtd_mean_Density","mean_FusionHeat","wtd_mean_FusionHeat",
     "gmean_FusionHeat","wtd_gmean_FusionHeat","gmean_ThermalConductivity",
     "wtd_gmean_ThermalConductivity"
log1: "wtd_range_atomic_mass","wtd_range_Density","range_FusionHeat",
      "wtd_range_FusionHeat","std_FusionHeat","wtd_std_FusionHeat"
```

Cuando estandarizamos los datos lo único que haremos será restarle la media aritmética a cada variable y dividirla por su desviación estándar. Este proceso también lo realizamos para la variable target, lo cual parece que podría hacer nuestros métodos inútiles (ya que no estamos prediciendo la temperatura crítica sino una transformación de ésta), pero como la transformación es perfectamente reversible podremos recuperar los valores reales con facilidad.

Entorno a procesos de clustering, por la naturaleza de regresión del problema, consideramos que no son eficaces y no los realizamos.

Sin embargo, para entender mejor nuestros datos, sí que realizamos un PCR (regresión sobre componentes principales) para entender mejor la importancia de los datos:

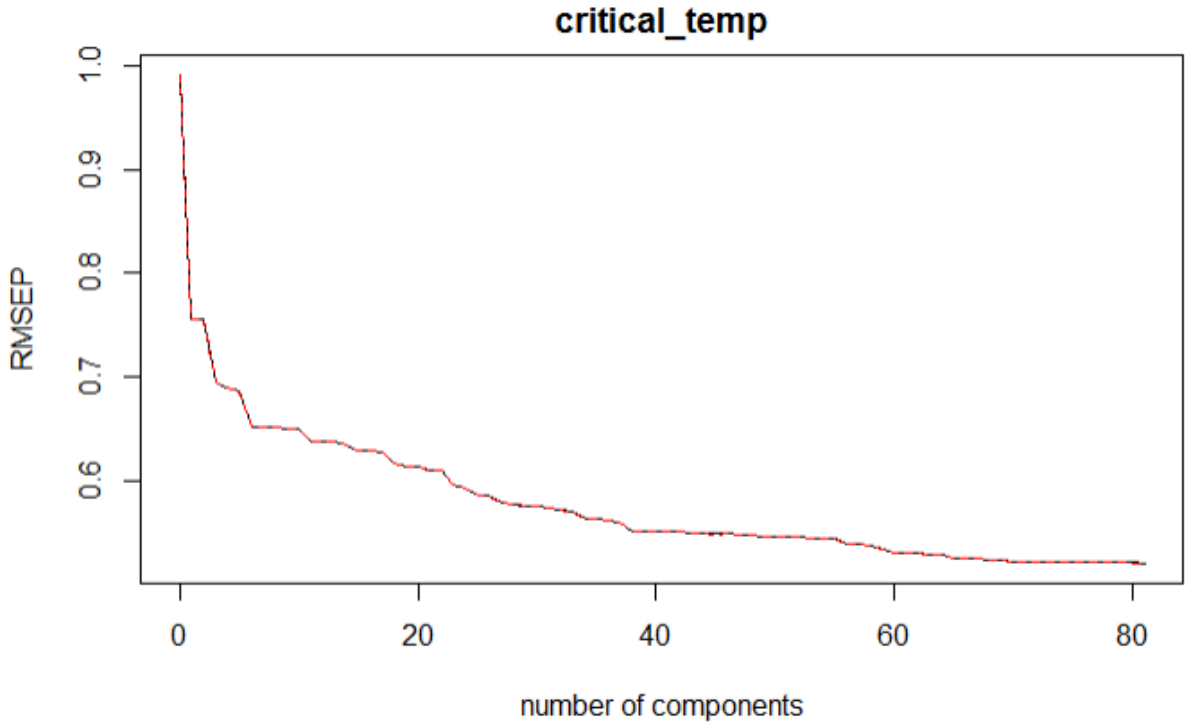

Figura 1: RMSE con respecto del número de variables analizado

Al realizar este análisis, nos percatamos de que la importancia de las variables a partir de la cuadragésima no reducen tanto el error. Sin embargo, entorno a la sexagésima vuelve a darse una ligera mejora.

Hemos realizado experimentos tanto considerando el conjunto de datos al completo como una versión reducida según el PCR. Específicamente, realizamos el estudio sobre el método de KNN y el de una SVM, pero en los resultados finales en ambos casos apuntan a que usar todos los datos nos ofrece un NMSE notablemente mejor.

4. Protocolos de validación

En cuanto al remuestreo de los datos para validar y sacar conclusiones de los modelos que hagamos, tomamos el siguiente modelo.

Dejaremos un 30 % de los datos (muestreados aleatoriamente pues no es relevante la ordenación del dataset en este caso) fuera del conjunto de aprendizaje, y lo usaremos para puntuar el modelo elegido cuando lo tengamos. Sólo entonces podremos mirar este conjunto de test.

Mientras tanto, para poder comparar los diferentes modelos que vamos a evaluar, usaremos la técnica de validación cruzada con $K = 10$. Dado nuestro tamaño de conjunto de aprendizaje, pensamos que es suficiente para tener una medida de error precisa sin estar excesivo tiempo entrenando y evaluando modelos, ya que algunos (como SVM) tardan demasiado como para usar técnicas como LOOCV.

Sin embargo, hemos tenido que realizar un pequeño truco a la hora de realizar la validación, en concreto para ejecutarla sobre SVMs. Para tratar de optimizar los hiperparámetros de éstas, en lugar de realizar una validación cruzada, realizamos otra partición sobre el conjunto de aprendizaje, dejando entorno a 10 % o 20 % para entrenamiento, y el resto para evaluar el error, y así poder aproximar los hiperparámetros.

Si bien el error aproximado por esta técnica no es el real, pequeñas pruebas (no incluidas en el trabajo final), nos llevan a la conclusión de que, aunque el error no es el mismo, los hiperparámetros sí que están muy cerca de los óptimos.

Por último, y para terminar con el protocolo de validación, como métrica de error usaremos el NMSE. La razón de que lo hagamos así es que se trata de un error comprensible dado el hecho de que hemos normalizado la columna a predecir (aunque se puede recuperar el error real fácilmente) y que nos indica qué porcentaje de la varianza no hemos conseguido explicar.

5. Métodos lineales y cuadráticos

5.1. Ridge Regression

En el método de ridge regression requerimos de optimizar el parámetro lambda, lo cual haremos a través de optimizar el GCV. Hacemos esto vía dos búsquedas, una primera en un rango grande de lambdas seguido de una con un rango más estrecho alrededor del mejor valor obtenido en la primera.

Al final obtuvimos un valor de 0.1 en la primera y de 0.106 en la segunda.

El error final de entrenamiento con lambda 0.106 es un 27.3 %.

Además, al mirar el coeficiente R2 del modelo (99.99818) vemos que nuestro problema muy probablemente es lineal y por tanto este tipo de métodos seguramente nos darán mejores resultados que los no lineales.

5.2. KNN

El método de K-nearest-neighbours es bastante sencillo y sólo requiere de optimizar el parámetro K. Para ello usamos la validación cruzada mencionada con anterioridad para obtener el siguiente gráfico:

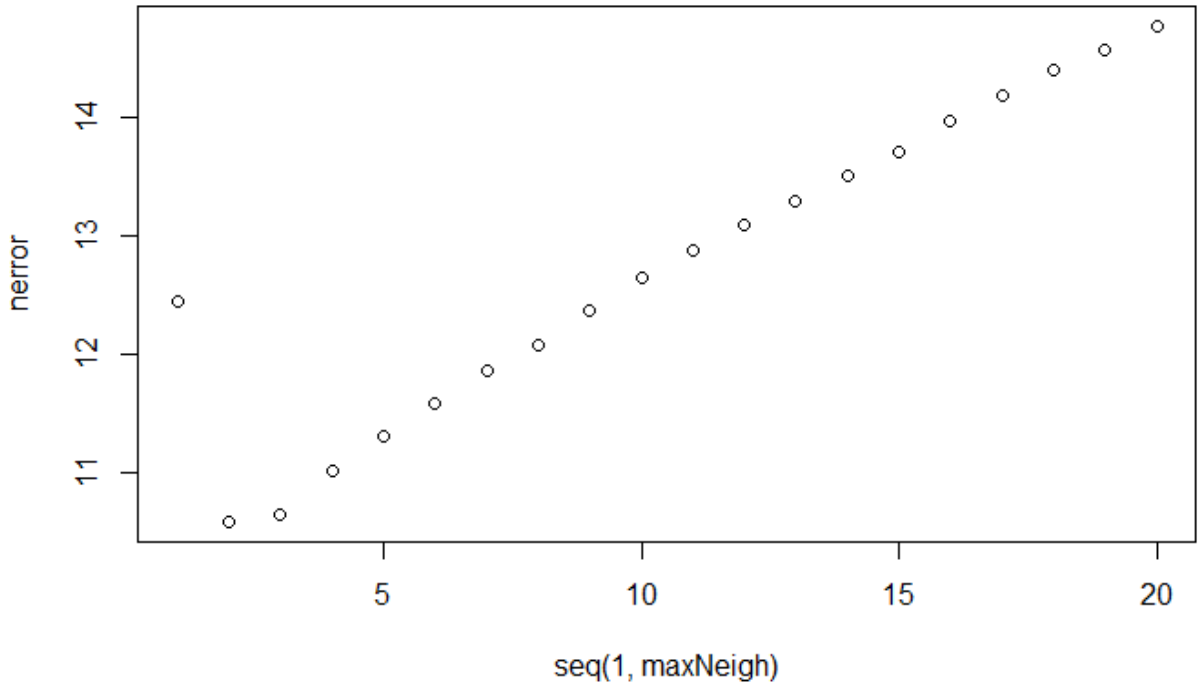

Figura 2: NMSE con respecto del número K de vecinos considerados

Vemos que el error mínimo se encuentra cuando consideramos dos vecinos, siendo mucho menor que considerando un solo elemento y aparentemente mejor que los siguientes. Nos hubiera gustado también poder modificar el peso de los vecinos en función de la distancia para ver si esto ayudaba a mejorar el error, pero no hemos encontrado modo de hacerlo con las librerías de R.

Nuestro error final es pues aquél del modelo con $K = 2$, es un 10.5823 %. Explicamos un 89.4 % de la variancia de la temperatura crítica con este método.

5.3. SVM cuadrática

En el método de SVM para regresión tenemos dos hiperparámetros a optimizar mínimo, la C y la ϵ. En concreto, para una con kernel cuadrático, solamente optimizaremos estos dos parámetros.

Sin embargo, el parámetro ϵ nos indica la región de margen que la máquina ignorará para hacer el cálculo. Debido tanto a la significación del parámetro como al tiempo que tarda en ejecutarse el modelo, dejaremos éste parámetro constante a 0.1, que equivale a aproximadamente 3 grados Kelvin (recordando la estandarización de la temperatura crítica).

No sólo esto, sino que dada nuestra cantidad de datos nos veremos forzados a trabajar sin validación cruzada, pues cada una de éstas nos llevaba entorno a dos horas de tiempo de ejecución para un único modelo.

El procedimiento que realizaremos será pues un muestreo pequeño para entrenamiento y el resto para calcular el error a la hora de optimizar. Inicialmente pensábamos que la optimización de los parámetros podría salir distorsionada debido a la reducción del conjunto de entrenamiento, pero bajo pequeñas pruebas descubrimos que, si bien el valor de error varía para un mismo valor de C entre validación y validación cruzada, el mínimo sigue estando en el mismo lugar, lo que nos permite hacer la exploración un tiempo más razonable.

Realizamos la exploración a partes, cada vez con conjuntos de entrenamiento mayores para lograr más precisión.

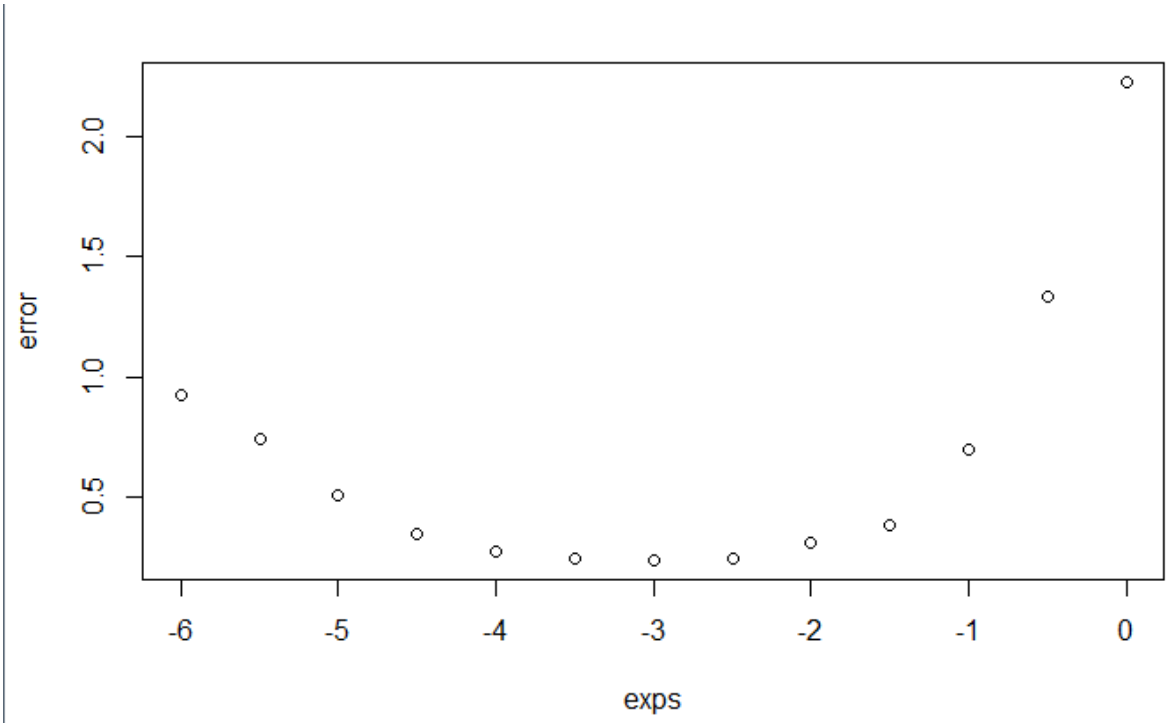

Figura 3: NMSE con respecto del exponente de C ($C = 10^x$) para 10 % de train

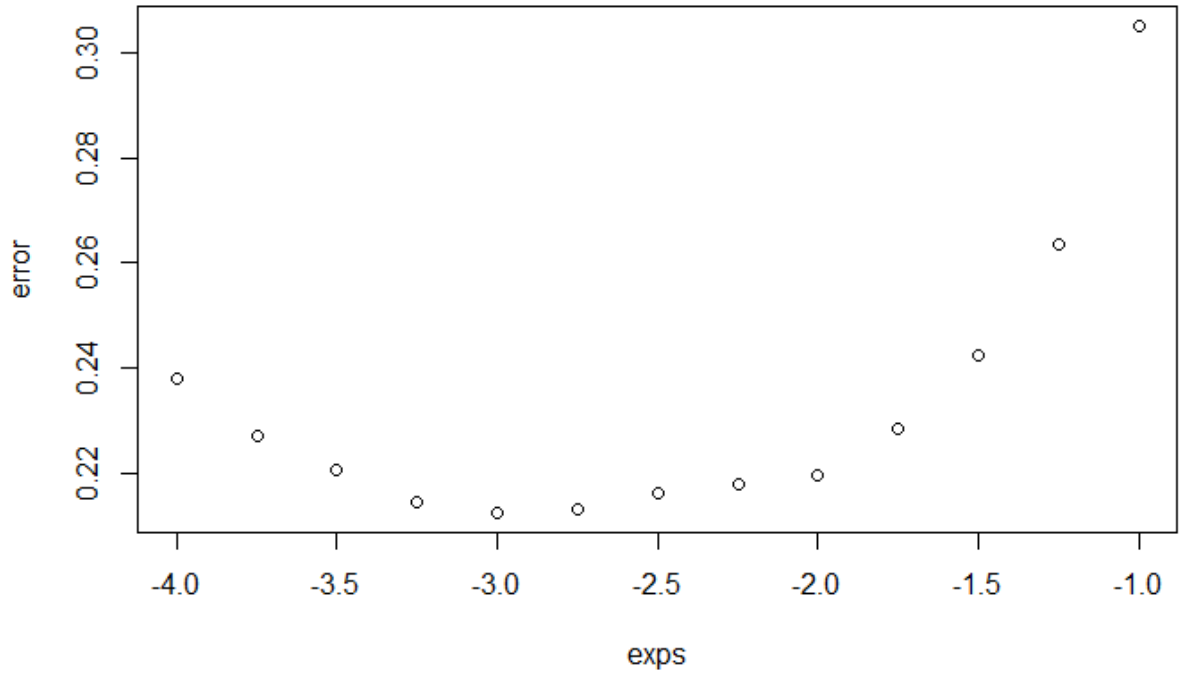

Figura 4: NMSE con respecto del exponente de C ($C = 10^x$) para 20 % de train

Detenemos aquí nuestra exploración, tomando $C = 10^{-3}$ como valor óptimo.

Al realizar la validación cruzada, obtenemos los siguientes resultados:

- **NMSE:** 18.1753 %
- **SV %:** 8502/13396 = 63.5 % (para uno de los modelos, el resto son más o menos lo mismo).

Como podemos ver, se trata de un modelo como mucho aceptable, pero bastante peor que alternativas como KNN.

6. Métodos no lineales

6.1. MLP

Para este método hemos de optimizar el valor de regularization y el número de neuronas.

Para hacer esto primero exploramos un rango grande de valores de regularización fijando un número alto de neuronas (40):

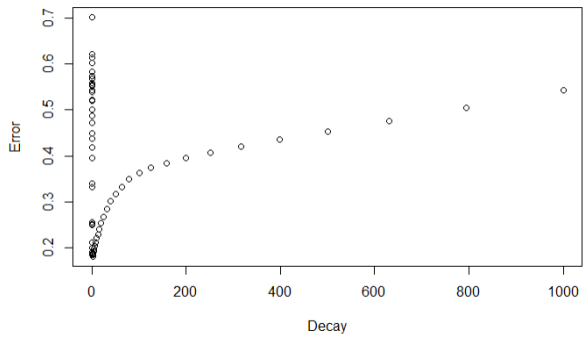

Figura 5: Valores de regularización con respecto a su error

Con lo que encontramos que el mejor valor de regularización es 1.

Seguidamente exploraremos un rango grande de neuronas usando este valor de regularización:

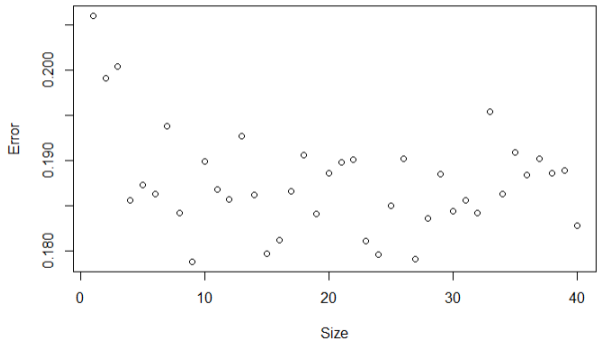

Figura 6: Número de neuronas con respecto a su error

Encontramos que tendremos los mejores resultados usando 9 o 27 neuronas. Dado que dan resultados muy similares utilizaremos 9, dado que es más simple.

Finalmente calculando el error con repeated cross validation del modelo final obtenemos que tiene un 18.3 % de error.

6.2. SVM con Kernel RBF

De la misma forma que con el kernel cuadrático, tendremos que optimizar C, pero ahora también deberíamos optimizar el parámetro σ del kernel.

Para ello, haremos un sencillo *Hill-climbing*, en que dejaremos fijo uno de los dos parámetros para optimizar el otro y luego intercambiar el constante y el fijo, repitiendo hasta que la diferencia de error sea marginal.

Empezamos optimizando C:

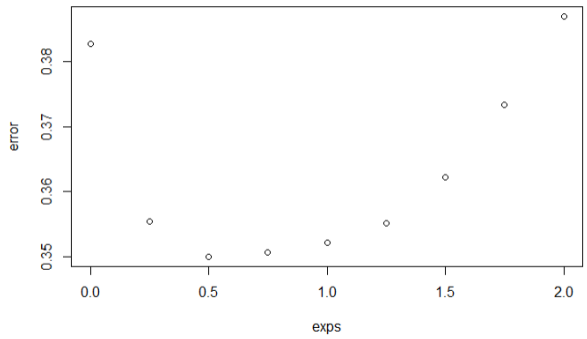

Figura 7: NMSE con respecto del exponente de C ($C = 10^x$) para 20 % de train

Subsecuentemente optimizamos σ, fijado $C = 10^{0,5}$

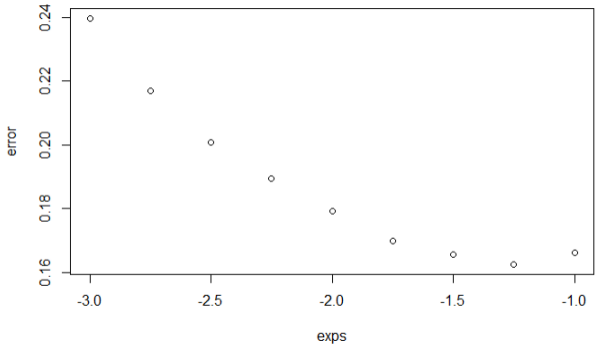

Figura 8: NMSE con respecto del exponente de σ ($\sigma = 10^x$) para 20 % de train

La siguiente optimización sobre C nos lleva a unos hiperparámetros finales de $C = 10^{1,2}$, $\sigma = 10^{-1,25}$, con un error aproximado de 0.165.

Haciendo ahora la validación cruzada de este modelo bajo las condiciones estipuladas en la sección 4.

- **NMSE:** 11.65468 %
- **SV %:** 6860/13396 = 51.21 %.

Podemos ver como nuestra SVM usando un kernel de base radial es bastante mejor que el kernel cuadrático. Sin embargo, notamos que hay una alta proporción de vectores soporte.

7. Modelo final

Finalmente nos decantaremos por KNN como modelo por diversas razones:

- Con un 10.5823 % de NMSE, es el más preciso

- No es un modelo complejo, al contrario que otros.
- Nos permitiría añadir más compuestos a posteriori y a medida que se vayan obteniendo medidas.

Al ejecutar el modelo entrenado con todo el conjunto de aprendizaje para predecir el conjunto de test (que hasta ahora no habíamos mirado), obtenemos las siguientes métricas:

```
NMSE = 10.1026%
MSE (sobre datos estandarizados) = 0.10101
RMSE (multiplicados por desviación original) = 10.89 K
```

El más interesante de estos datos es el RMSE. En concreto, mencionamos anteriormente que habíamos estandarizado la columna de temperatura crítica, así que errores como MSE y RMSE solo son fiables para comparar entre sí en su forma actual. Sin embargo, al multiplicar el RMSE por la desviación estándar por la cual dividíamos en el preprocesado, obtenemos el valor real del RMSE en grados Kelvin.

En concreto, el RMSE nos proporciona el dato de que nuestra temperatura predicha se espera en gran mayoría a menos de 10.89 K de la temperatura real de superconducción del material.

8. Resultados y conclusiones

8.1. Análisis de éxitos logrados.

Recapitulando, recordamos que nuestro objetivo es tratar de crear un modelo para predecir la temperatura crítica a partir de datos disponibles. Consideramos que los datos usados son suficientemente sencillos de obtener para materiales nuevos. Además, el error de tan solo 10 grados de diferencia en el mayor número de los casos es suficientemente bueno como para considerar el resultado como buen modelo, si bien no lo es tanto como el obtenido por artículos anteriores.

Dado que los datos originales no solo contienen superconductores convencionales, también consideramos un éxito la predicción más allá de las fórmulas de pares de Cooper, si bien no son interpretables en su forma actual.

El modelo obtenido en sí es muy sencillo ya que se trata de un *2-nearest-neighbours*, lo que nos permite mantener la simplicidad y comprensibilidad de nuestro resultado.

Sin embargo, no hemos logrado extraer el porqué éste modelo funciona tan bien en comparación con los otros, ni vincularlo con el cómo continuar la investigación sobre superconducción, que era el más avanzado de nuestros objetivos.

8.2. Conclusiones científicas y personales

Debido al modelo que ha surgido de entre todos, nos preguntamos si un modelo lineal sería realmente suficiente para expresar el problema de temperatura crítica. Además, también el hecho de que el modelo sencillamente extrapole la temperatura crítica de un material a partir de los dos

más similares (que pueden no compartir ni un solo elemento en común), nos lleva a plantearnos si el problema de temperatura crítica se podría resolver a partir de un conjunto de variables no muy distinto al actual.

8.3. Limitaciones y extensiones posibles.

La mayor limitación del trabajo ha sido la tardanza en tiempo de ejecución de varios de los modelos. Debido a la baja velocidad de nuestras máquinas de computación, hemos tenido que recurrir a algunos trucos no del todo fiables para optimizar parámetros, y no hemos podido explorar algunos modelos en su totalidad debido a esta misma razón.

Como extensión, nos gustaría buscar algún método para realizar KNN pero usando diferentes métricas de distancia y/o peso en el voto, para poder afinar mejor el modelo.

Referencias

[1] Hamidieh, Kam. *A data-driven statistical model for predicting the critical temperature of a superconductor*. Statistics Department, The Wharton School, University of Pennsylvania, arXiv:1803.10260, 2018.

[2] Wikipedia, *Superconductivity* accesible en
`https://en.wikipedia.org/wiki/Superconductivity`

[3] Wikipedia, *Unconventional Superconductors* accesible en
`https://en.wikipedia.org/wiki/Unconventional_superconductor`

[4] Wikipedia, *Cooper Pairs* accessible en
`https://en.wikipedia.org/wiki/Cooper_pair`