Support Vector Machine aplicado a *dataset seeds*

Pedro Pablo Silva Antilef[1 [0000−1111−2222−3333]]

Universidad de Santiago de Chile, Facultad de ingeniería, Departamento de ingeniería informática, Estación Central, Santiago de Chile pedro.silva@usach.cl

Abstract. En este informe se muestra un análisis del *dataset seeds*, una base de datos con información geométrica de 3 tipos de semillas de trigo. La idea principal es generar un modelo de clasificación de aprendizaje de máquinas llamado *Support Vector Machine*. Para esto se realizan diferentes experimentos, donde se buscará variar los parámetros del modelo, así como también los atributos de la base de datos basado en conocimiento extraído de este trabajo como de trabajos anteriores. Los resultados son excepcionales, obteniendo una clasificación perfecta, es decir mejores resultados a los encontrados en experiencias pasadas como por ejemplo *Random Forest* o el mismo artículo que diseño el *dataset*. También es posible realizar un análisis de la linealidad y no linealidad del fenómeno estudiado utilizando la optimización de los hiperparámetros con el método de grilla, encontrando que el comportamiento del fenómeno sigue más bien una tendencia lineal.

Keywords: Support Vector Machine · Machine Learning · Data Minning.

1 Introducción y estado del arte

El reconocimiento y autentificación son tareas esenciales para garantizar la calidad de las semillas en la cadena productiva industrial, con énfasis en el proceso de certificación. Estas tareas se siguen haciendo de manera manual en varias industrias, por lo que la aplicación de métodos computacionales y automatizados podría significar una gran ayuda para diferentes empresas [1].

El reconocimiento de variedades también es una tarea crítica para limitar cualitativa y cuantitativamente las posibles pérdidas en el campo de cultivo, así como también para predecir la cosecha productiva asincrónicas [2].

Tal como se mencionó en la primera instancia de estos informes, se presenta un estado del arte sobre aplicaciones sobre el mismo dataset, donde podemos encontrar [3], donde se utiliza este el algoritmo de cluster gradiente completo o CGCA por sus siglas en ingles (*Complete Gradient Clustering Algorithm*), con el que busca definir grupos que permitan representar las características de los granos de trigo, en base a la distribución de los datos, ya que el CGCA necesita calcular la estimación de densidad de kernel [5]. Cada cluster es caracterizado por un máximo local de la estimación de densidad de kernel. Con esto, las regiones

con una mayor densidad de objetos son definidos como un grupo o cluster. Cada dato es asignado a cada cluster utilizando el método de gradiente ascendente.

Por otro lado existen varios acercamiento en el último tiempo que se han desarrollado con técnicas de redes neuronales profundas, mejorando los resultados, utilizando una gran cantidad de datos, pudiendo entrenar estos modelos de manera mucho más robusta [1][4].

2 Datos

El *dataset* utilizado es demoninado *seeds*, fue creado por el Instituto de Agrofísica de la Academia Polaca de Ciencias en Lublin, pero extraídos desde el repositorio *UCI Machine Learning Repository* de la Universidad de California Irvine [**?**]. Contiene 210 registros donde se describen 7 características geométricas del núcleo de 3 variedades de semillas de trigo: Kama, Rosa y Canadian, 70 elementos para cada variedad. Para esto se utilizaron técnicas de rayos X débil. Las variables contenidas en el *dataset* son: Área A, perímetro P, compactibilidad $C = 4\pi A/P^2$, largo del núcleo *l_kernel*, ancho del núcleo *w_kernel*, coeficiente de asimetría *assimemtry* y largo de la ranura del núcleo *l_groove*. Todas las variables son numéricas continuas y no existen datos con valores nulos o faltantes.

3 Metodología

Para este informe y como se mencionó anteriormente se utilizará el modelo de *machine learning Support Vector Machine* (SVM), el cual es es un modelo de aprendizaje supervisado, utilizado tanto para regresión como para clasificación, el último caso es el que se aplicará para el *dataset Seeds*, ya que contamos con las etiquetas para cada uno de los registros, es decir el tipo de semilla de trigo.

Este modelo puede ser aplicado en diferentes dimensiones dependiendo del tipo de problema y del comportamiento de los datos a analizar, en este sentido se definen dos tipos de kernel, lineales y no lineales, donde el primero genera una separación de los datos a partir de un hiper plano considerando restricciones blandas y duras. Por otra parte se pueden ajustar los datos a un kernel no lineal, el cual puede ser del tipo radial, polinomial, gaussiano, entre otros. Para el caso del presente informe, se analizarán los kernel del tipo lineal y radial.

Este modelo cuenta con parámetros para los kernel, en el caso del kernel lineal se busca ajustar el parámetro C, mientras que cuando se utiliza un kernel radial, se ajustan los parámetros C y *Gamma*. Estos parámetros no sólo son útiles para obtener un mejor modelo en términos del error de clasificación, si no que también entregan información importante respecto a la "naturaleza" de los datos. Para el caso del parámetro C, un valor alto significará que se ajusta el modelo a los datos de una manera "más lineal", mientras que un valor menor tendrá un comportamiento en mayor medida no lineal sobre los datos. Lo contrario sucede con el *Gamma*.

Para analizar el modelo sobre el *dataset Seeds*, se realizarán dos experimentos uno utilizando el *dataset* completo y otro con las variables más importantes (en

base a un algoritmos de selección de características), ambos utilizando el kernel lineal y kernel radial. Para encontrar los valores de los parámetros mencionados anteriormente, se utiliza el método de optimización de hiperparámetros de grilla o cuadrícula, en el que simplemente se construye un modelo con cada una de las combinaciones de los valores proporcionados, evaluando cada uno de los modelos y seleccionando aquel con el mejor error de clasificación.

Para la selección de características se utilizó el *package Weka*, para el software estadístico R. obteniendo los siguientes resultados.

Ranking	**Variable**	**InfoGain**
1	area	1.2447613
2	perimeter	1.2040836
3	w_kernel	1.0151603
4	l_kernel	0.9367747
5	l_groove	0.8150603
6	compactness	0.3251864
7	assymetry	0.2382071

Table 1: Importancia de características según *package Weka*

En base a esto y en base al análisis de correlación presentado en instancias anteriores, se decide por utilizar para el experimento 2, sólo los atributos Area, l_kernel, l_groove, despreciando el atributo *perimeter*, debido a que tiene una correlación cercana a 1 con el atributo *area*, por lo que podría estar entregando información redundante. Por lo tanto los experimentos quedan de la siguiente manera:

1. Considerando la totalidad de los atributos del *dataset*.
   (a) kernel lineal
   (b) kernel radial
2. Considerando sólo los atributos Area, l_kernel, l_groove.
   (a) kernel lineal
   (b) kernel radial

4 Resultados

En esta sección se muestran los resultados obtenidos de los experimentos detallados en la sección anterior.

Para el caso del experimento 1.(a), se realizan un total de 10 combinaciones, considerando una función de costo $c = \{2^{-3}, ..., 2^{6}\}$, es decir, valores entre 0.125 y 64. Los resultados de las combinaciones se pueden encontrar en la tabla 2, donde se muestra el mejor modelo, el cual utilizó una función de costo de 16, obteniendo un error de 0.04761905. En este caso el modelo con costo 16 y 32 tienen el mismo resultado, por el principio de parsimonia, se selecciona el modelo con un menor

costo. Para el caso del experimento 1.(b), la optimización de hiperparámetros generó un total de 1690 conjugaciones, utilizando como funciones de costo y gamma $c = \{2^{-3}, ..., 2^{6}\}$ y $gamma = \{2^{-4}, ..., 2^{8}\}$ respectivamente. Con estos límites, se encontró que el mejor modelo era aquel con un *gamma* de 0.25 y un costo de 64, con un error de 0.03809524.

cost	error
0.125	0.07619048
0.250	0.07142857
0.500	0.06666667
1.000	0.06190476
2.000	0.05714286
4.000	0.06666667
8.000	0.07142857
16.000	**0.04761905**
32.000	0.04761905
64.000	0.05238095

Table 2: Modelos evaluados por el método de grilla para experimento 1.(a)

A continuación de muestran las matrices de confusión para cada uno de los mejores modelos obtenidos por el método de la grilla. En base a esta matriz se calcula el índice de Kappa, el cual mide el error en función de la clasificación de clases del modelo, obteniendo un índice de 0.979 y 1.00 para los experimentos 1.(a) y 1.(b) respectivamente.

	type		
pred	**1**	**2**	**3**
1	69	0	2
2	0	70	0
3	1	0	68

Table 3: Matriz de confusión para experimento 1.(a)

	type		
pred	**1**	**2**	**3**
1	70	0	0
2	0	70	0
3	0	0	70

Table 4: Matriz de confusión para experimento 1.(b)

Para el experimento 2.(a), se realizan un total de 12 combinaciones, con una función de costo $c = \{2^{-3}, ..., 2^{8}\}$, es decir, valores entre 0.125 y 256. Los resultados de los modelos se muestran en la tabla 5, donde el mejor modelo utilizó un costo de 128, con un error de 0.04761905. Para el caso del experimento 2.(b), la optimización de hiperparámetros generó un total de 1690 conjugaciones, utilizando como funciones de costo y gamma $c = \{2^{-3}, ..., 2^{6}\}$ y $gamma = \{2^{-4}, ..., 2^{8}\}$ respectivamente. Con estos límites, se encontró que el mejor modelo era aquel con un *gamma* de 0.125 y un costo de 32, con un error de 0.03809524 .

cost	**error**
0.125	0.07619048
0.250	0.08095238
0.500	0.07142857
1.000	0.07142857
2.000	0.05714286
4.000	0.06190476
8.000	0.06190476
16.000	0.06666667
32.000	0.05714286
64.000	0.05238095
128.000	**0.04761905**
256.000	0.04761905

Table 5: Modelos evaluados por el método de grilla para experimento 2.(a)

Las matrices de confusión para el experimento 2, se muestran a continuación. A partir de estas se puede calcular el índice de Kappa, con un valor de 0.964 y 0.957 para el experimento 2.(a) y 2.(b) respectivamente.

	type		
pred	**1**	**2**	**3**
1	65	0	2
2	1	70	0
3	4	0	70

Table 6: Matriz de confusión para experimento 2.(a)

	type		
pred	**1**	**2**	**3**
1	64	0	0
2	1	70	0
3	5	0	70

Table 7: Matriz de confusión para experimento 2.(b)

5 Discusión y conclusiones

Al analizar los resultados obtenidos y detallados en la sección anterior, podemos observar que si bien todos los experimentos tienen un alto rendimiento en cuanto a la clasificación, se pueden apreciar unas pequeñas diferencias entre los mejores modelos encontrados por la optimización de hiperparámetros. Lo primero que podemos observar es que utilizando el *dataset* completo se obtuvieron mejores resultados que utilizando un *dataset* con la selección de características. Si bien estas diferencias son mínimas o inexistentes desde el punto de vista del error obtenido por el método de grilla (10 carpetas de validación cruzada), desde el punto de vista del índice de kappa y el análisis de la matriz de confusión, sí se pueden apreciar mayores diferencias (0.01 si se compara el índice Kappa), obteniendo un 100% de efectividad en el experimento 1.(b), donde se predijeron todos los registros del *dataset*. Por ejemplo en el experimento 1, se clasificaron de manera incorrecta 3 y 0 registros para el experimento (a) y (b) respectivamente,

mientras que para el experimento 2, se clasificaron incorrectamente 7 y 6 para los experimentos (a) y (b) respectivamente.

Analizando los resultados desde el punto de vista de los parámetros que entregaban los mejores resultados, se puede apreciar que los valores de C siempre fueron los más bajos respecto al rango de búsqueda utilizado, y lo contrario con el parámetro *gamma*, donde siempre se seleccionaron los valores más bajos. De lo anterior es posible inferir que el fenómeno estudiado tiende a tener un comportamiento más bien lineal, según lo estudiado en la teoría. Esto también se puede relacionar con los tan buenos resultados obtenidos, los que podrían ser considerados sospechosos, pero que una herramienta tan poderosa como las SVM, es capaz de clasificar con un rendimiento excepcional.

Comparando los resultados respecto a las experiencias anteriores, por ejemplo con el laboratorio de *Random Forest*, los resultados con SVm son mucho mejores, ya que en dicha experiencia, se obtuvo un Kappa de 0.921. También se obtuvieron mejores resultados respecto al artículo que diseñó el *dataset* [3], donde se utilizó el algoritmo de cluster gradiente completo o CGCA por sus siglas en ingles (*Complete Gradient Clustering Algorithm*), donde si bien, no se trata de un modelo de clasificación, el clustering agrupo de manera errónea muchos más registros que los clasificados en esta experiencia.

References

1. Karim Laabassi: Wheat varieties identification based on a deep learning approach. Journal of the Saudi Society of Agricultural Sciences (2021)
2. Taheri-Garavand, A., Nasiri, A., Fanourakis, D., Fatahi, S., Omid, M., & Nikoloudakis, N.: Automated In Situ Seed Variety Identification via Deep Learning: A Case Study in Chickpea. Journal of the Saudi Society of Agricultural Sciences Plants (Basel, Switzerland), 10(7), 1406 (2021). https://doi.org/10.3390/plants10071406
3. Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. : Complete gradient clustering algorithm for features analysis of x-ray images. In Information technologies in biomedicine. (pp. 15-24). Springer, Berlin, Heidelberg. (2010)
4. Zhou Lei, Zhang Chu, Taha Mohamed Farag, Wei Xinhua, He Yong, Qiu Zhengjun, Liu Yufei: CWheat Kernel Variety Identification Based on a Large Near-Infrared Spectral Dataset and a Novel Deep Learning-Based Feature Selection Method. Frontiers in Plant Science. (2020).
5. Medium Estimación de densidad de kernel, https://medium.com/@garzonsergio/m%C3%A9todos-de-estimaci%C3%B3n-de-densidad-de-kernel-de-odf-a-ebsd-b4a143dc9eee. Last accessed 12 Abr 2022 classifier/". Rescatado el 10 de Mayo de 2020.