APELLIDOS:	PL:
NOMBRE:	DNI:

ESCUELA DE INGENIERÍA INFORMÁTICA

SISTEMAS INTELIGENTES

Examen Final de Teoría. Martes 26 de junio de 2020.

I. Búsqueda

El famoso problema MAX_CUT se puede enunciar del siguiente modo: dado un grafo no dirigido G =(V,E) con costes positivos en los arcos, se trata de calcular una partición del conjunto de nodos V en dos subconjuntos V_1 y V_2 (es decir $V = V_1 \cup V_2$ y $V_1 \cap V_2 = \emptyset$). Si denotamos por E_1 al subconjunto de arcos de E que conectan pares de nodos de V_1 (análogamente E_2), y por Coste(E_1) la suma de los costes de los arcos de E_1 (análogamente Coste(E_2)), el objetivo es encontrar la partición (V_1,V_2) que minimice el valor de Coste(E_1)+ Coste(E_2).

En esta pregunta se pide resolver el problema MAX_CUT utilizando búsqueda en espacios de estados, concretamente el algoritmo A*, y con un algoritmo genético (AG). Mas concretamente:

Solución con A*

1.- [1 punto] Describir de forma precisa los estados y las reglas, es decir el espacio de búsqueda, de forma genérica, indicando si es un árbol o un grafo (tened en cuenta que puede haber reglas de coste 0 siempre y cuando no se generen ciclos en el espacio de búsqueda de coste 0, o caminos de longitud infinita y coste 0).

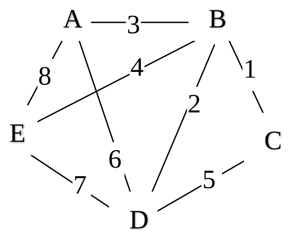
RESP: Dado que una solución del problema viene dada por un par de conjuntos (V_1,V_2) tales que V_1 y V_2 son disjuntos y contienen a todos los nodos del grafo, por ejemplo el par ({A,B,E} {C,D}) para el grafo del apartado siguiente, lo natural parece representar los estados por pares de subconjuntos de nodos (C_1, C_2) tales que C_1 y C_2 sean disjuntos y que su unión contenga un subconjunto de nodos del grafo. Así, el par ({A},{E,D}) sería en principio un estado válido.

Los sucesores de un estado (C_1, C_2) pueden venir dados, en principio, por todas las opciones de añadir una ciudad no incluida en $C_1 \cup C_2$ a C_1 y a C_2. Así, el estado ({A},{E,D}) tendría, entre otros, como sucesores a los estados ({A,C},{E,D}) y ({A},{E,D,C}). El coste de cada regla sería la suma de los costes de las conexiones entre la nueva ciudad y todas las ciudades ya incluidas en C_1, si la nueva ciudad se incluye en C_1, (o en C_2 si se incluye en este conjunto). En el ejemplo anterior, los costes serían: 0 (coste entre A y C) y 5 (0 coste entre C y E, coste 5 entre C y D). El estado inicial podría ser el par ($\emptyset$,$\emptyset$). La solución óptima en este ejemplo es ({A,C,E},{B,D}) y tiene un coste C*=10.

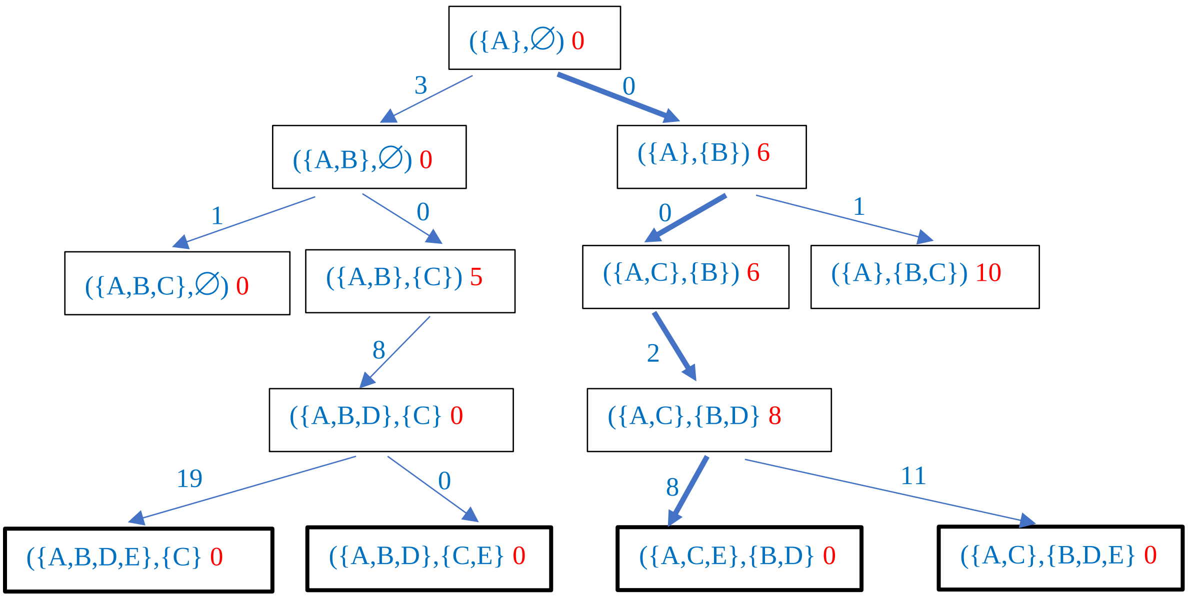
Con estas ideas, el espacio de búsqueda sería un grafo acíclico pero con muchas conexiones ya que es evidente que en general habrá muchos caminos para llegar desde el nodo inicial a un nodo intermedio. Pero todos estos caminos tienen el mismo coste; con lo que, si fuésemos capaces de definir un espacio que solamente contenga uno de estos caminos, el espacio sería mucho más reducido y representaría las mismas soluciones, es decir sería completo.

Para ello, lo que podemos hacer es establecer un orden lineal entre las ciudades (de forma similar a cómo se establece un orden de filas en el problema de las N-reinas), e ir desarrollando el espacio de búsqueda por niveles: para los sucesores del inicial se considera solo la primera ciudad, para los sucesores de estos la segunda y así sucesivamente. Así tendremos un espacio de búsqueda en forma de árbol binario de N+1 niveles, si hay N ciudades. Al igual que en las N-reinas, podemos eliminar simetrías si partimos de un estado inicial en el que la primera ciudad está en el conjunto C_1, así nunca puede estar en C_2, y la altura del árbol se reduce en 1.

2.- [0,5 puntos] Dibujar una parte representativa del espacio de búsqueda para el problema dado por el siguiente grafo



RESP: Una parte significativa del espacio de búsqueda debe contener al menos una solución y un conjunto de nodos intermedios en varios niveles del árbol. Consideramos el orden A,B,C,D,E.



3.- [1 punto] Diseñar un heurístico razonable, a ser posible utilizando el método de la relajación del problema. Indicar las propiedades del heurístico, y dar sus valores para la fracción del espacio de búsqueda dibujado anteriormente. PISTA: una forma de relajar el problema es prescindir de algunos de los arcos.

RESP: En este caso, dado el subproblema planteado por un estado (C_1, C_2), se puede relajar el problema prescindiendo de los arcos que unen ciudades no incluidas en $C_1 \cup C_2$. De esta forma el problema relajado se resuelve de forma óptima en tiempo polinomial sin más que recorrer las ciudades no incluidas en $C_1 \cup C_2$, y para cada ciudad c calculamos Coste(c, C_1) como la suma de los costes entre c y todas las ciudades de C_1, análogamente Coste(c, C_1). Así, en el problema relajado incluimos la ciudad c en el subconjunto con menor coste, y así obtenemos el coste óptimo del problema relajado.
El heurístico es consistente al haber sido diseñado por el método de la relajación del problema.
Los valores de este heurístico para cada uno de los nodos del espacio anterior son los números que aparecen en color rojo. Nótese, que Coste(c, ∅) = 0, ya que no hay arcos entre c y las ciudades de ∅.
Ejemplo: h[({A,B},{C})] = min(Coste(D,{A,B}),Coste(D,{C})) + min(Coste(E,{A,B}),Coste(E,{C})) =

min(6+2,5) + min(8+4,0) = 5+0 = 5 < h*[({A,B},{C})] = 8.

Solución con AG

4.- [0,5 puntos] Dar una codificación y un algoritmo de decodificación adecuados. Indicar si con este esquema de codificación/decodificación es posible obtener cualquier solución del problema.

RESP: Dado que se trata de distribuir los elementos de un conjunto entre dos subconjuntos, cualquier partición del conjunto de nodos en dos subconjuntos es una solución candidata. De esta forma podemos utilizar una codificación binaria de longitud N (número de nodos). La forma de decodificar un cromosoma puede ser la siguiente: consideramos un orden de los nodos, por ejemplo A,B,C,D,E en el grafo anterior. Así, un 0 en la posición de un nodo indica que éste pertenece a V_1 y un 1 indica que pertenece a V_2. Por ejemplo, el cromosoma que representa la solución óptima (ver espacio de búsqueda) será (0 1 0 1 0).
En este problema, cada solución candidata tienen una única representación y cualquier solución se puede representar mediante un cromosoma.

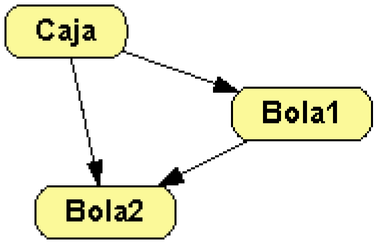
5.- [0,5 puntos] Definir operadores de cruce y mutación adecuados teniendo en cuenta el esquema de codificación/decodificación y el problema MAX_CUT.

RESP: En este caso, la información relevante de un cromosoma es el hecho de que tenga un 0 o un 1 en una determinada posición. El hecho de que haya dos o más 0s o 1s consecutivos no es relevante. Por esa razón, si queremos diseñar un operador de cruce que traslade propiedades de los padres a los hijos, debemos pensar en un operador que conserve 0s y/o 1s. Una idea puede ser la siguiente: Si los dos padres tienen el mismo valor en una posición, el hijo toma ese mismo valor en la posición, si los padres tienen valores distintos, entonces en el hijo se elige de forma aleatoria entre 0 y 1.

II.- Representación

6.- [1,5 puntos] Una de las pruebas de un concurso consiste en que el concursante tiene que elegir entre dos cajas, y solo una de ellas tiene el premio. Para elegir, puede sacar sin mirar dos bolas de forma consecutiva y sin reemplazo de solo una de las cajas. La caja con premio contiene 6 bolas rojas y 4 negras, mientras que la caja sin premio contiene 3 bolas rojas y 7 negras. Modela este problema mediante una red bayesiana, con su estructura y sus probabilidades condicionadas. Calcula, en base a la red creada, cuál será la probabilidad de haber sacado bolas de la caja buena si se han sacado en un caso primero una bola negra y después una roja, y en otro caso primero una bola roja y después una negra.

Solución: El grafo de la red bayesiana será como se muestra en la figura a la derecha.



La caja elegida para sacar las bolas influye en la probabilidad del color de las bolas y también el color de la bola sacada en primer lugar influye en la segunda porque habrá una bola menos del color sacado. Las probabilidades de cada nodo, Caja, Bola1 y Bola2, son:

Buena	0.5
Mala	0.5

Caja	Buena	Mala
Roja	0.6	0.3
Negra	0.4	0.7

Caja	Buena	Buena	Mala	Mala
Bola1	Roja	Negra	Roja	Negra
Roja	0.56	0.67	0.22	0.33
Negra	0.44	0.33	0.78	0.67

La probabilidad para la caja buena al sacar negra y luego roja es:

$$P(Caja = B|Bola1 = N, Bola2 = R)$$
$$= \frac{P(Caja = B) \cdot P(Bola1 = N|Caja = B) \cdot P(Bola2 = R|Caja = B, Bola1 = N)}{P(Bola1 = N, Bola2 = R)}$$
$$= \frac{0.5 \cdot 0.4 \cdot 0{,}67}{0.25} = 0.54$$

La probabilidad para la caja buena al sacar roja y luego negra es:

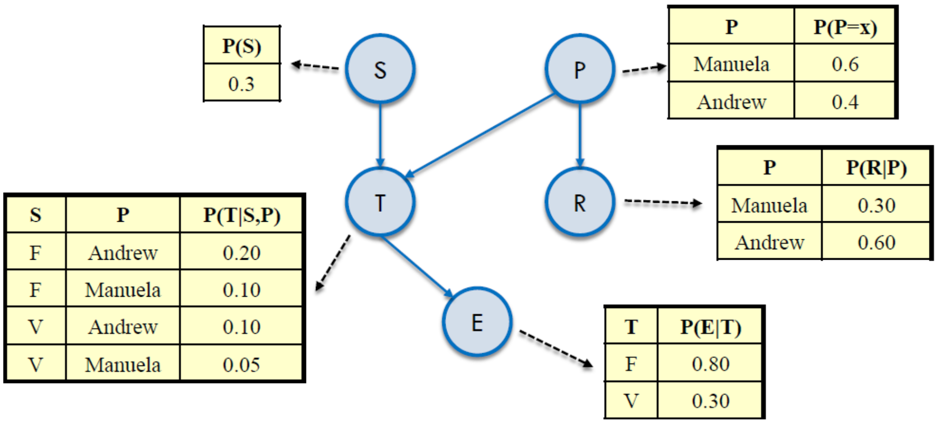
$$P(Caja = B|Bola1 = R, Bola2 = N)$$
$$= \frac{P(Caja = B) \cdot P(Bola1 = R|Caja = B) \cdot P(Bola2 = N|Caja = B, Bola1 = R)}{P(Bola1 = R, Bola2 = N)}$$
$$= \frac{0.5 \cdot 0.6 \cdot 0.44}{0.25} = 0.53$$

Para estos cálculos hemos necesitado las probabilidades conjuntas de *Bola1* y *Bola2*:

$$P(Bola1 = N, Bola2 = R)$$
$$= \sum_{Caja} P(Caja) \cdot P(Bola1 = N|Caja) \cdot P(Bola2 = R|Caja, Bola1 = N)$$
$$= 0.5 \cdot 0.4 \cdot 0.67 + 0.5 \cdot 0.7 \cdot 0.33 = 0.25$$

$$P(Bola1 = R, Bola2 = N)$$
$$= \sum_{Caja} P(Caja) \cdot P(Bola1 = R|Caja) \cdot P(Bola2 = N|Caja, Bola1 = R)$$
$$= 0.5 \cdot 0.6 \cdot 0.44 + 0.5 \cdot 0.3 \cdot 0.78 = 0.25$$

7.- [1,5 puntos] Dada la siguiente red bayesiana y usando las muestras que vienen debajo, obtén $P(E|S, R)$ mediante el método de inferencia por muestreo con ponderación de la verosimilitud.



Muestra	**P**	**R**	**S**	**T**	**E**
1	Manuela	V	V	F	V
2	Manuela	V	V	F	F
3	Manuela	V	V	F	V
4	Andrew	V	V	F	F
5	Andrew	V	V	F	F
6	Manuela	V	V	F	V
7	Andrew	V	V	F	V
8	Andrew	V	V	V	V
9	Manuela	V	V	F	V
10	Manuela	V	V	F	V

Solución: Estas muestras ya se han generado forzando los valores de la evidencia, es decir verdadero para R y S. Antes de poder contar las combinaciones hay que calcular el peso de cada una de ellas. Como *S* no tiene padres tomamos 0.3 siempre, pero como *R* depende de *P* habrá que mirar el valor del padre en cada configuración para completar el peso.

Muestra	Peso
1	0.3*0.3=0.09
2	0.3*0.3=0.09
3	0.3*0.3=0.09
4	0.3*0.6=0.18
5	0.3*0.6=0.18
6	0.3*0.3=0.09
7	0.3*0.6=0.18
8	0.3*0.6=0.18
9	0.3*0.3=0.09
10	0.3*0.3=0.09

Solo las muestras 1,3 y 6-10 son compatibles con *E* siendo verdadero, entonces $N_S = 0.09 \cdot 5 + 0.18 \cdot 2 = 0.81$. Por otro lado $N_C = 0.09 \cdot 6 + 0.18 \cdot 4 = 1.26$. Entonces, la estimación de $P(E|S,R) = \frac{N_S}{N_C} = \frac{0.81}{1.26} = 0.64$

III.- Aprendizaje

8.- **[0.5 puntos]** Explica qué es un vector soporte, cuál es su posición relativa con respecto al hiperplano de separación y por qué son importantes en SVM.

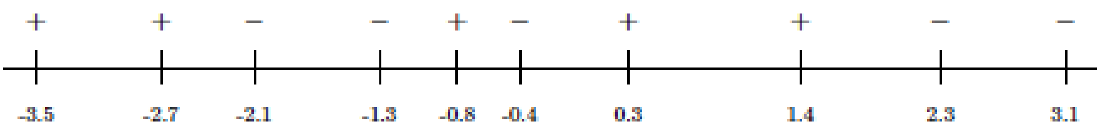
Los vectores soporte son ejemplos de entrenamiento que están a una distancia mínima del hiperplano de separación. Son importantes porque son lo que permiten realizar la clasificación.

9. [0.5 puntos] Dado el siguiente conjunto de entrenamiento, responde razonadamente si mediante el método C4.5 es posible construir un árbol de decisión que obtenga un porcentaje de acierto del 100% para ese conjunto. Si la respuesta es verdadera dibuja el árbol. Si es falsa explica la razón por la que es falsa.

A	B	C	Y
0	1	0	Yes
1	0	1	Yes
0	0	0	No
1	0	1	No
0	1	1	No
1	1	0	Yes

El ejemplo 2 y el 4 toman los mismos valores para las variables A, B y C pero pertenecen a distinta clase, por lo que no es posible obtener un árbol que en ese conjunto tenga un 100% de acierto.

10.- [1.25 puntos] Considera los siguientes ejemplos con un solo atributo real clasificados en dos clases (+,-) y el clasificador KNN, con K=3.



x	-3.5	-2.7	-2.1	-1.3	-0.8	-0.4	0.3	1.4	2.3	3.1
y	+	+	−	−	+	−	+	+	−	−

Suponiendo que entrenamos este clasificador con Validación cruzada de 5 cajas, **da una posible partición (solo una)** y calcula el error cometido en **UNA** iteración de la validación cruzada.

Hay tanta soluciones como combinaciones de 10 ejemplos tomados de 2 en 2. Así que elijo 1.
Caja1: (-3.5,+), (-2.7,+)
Caja2: (-2.1,-), (-1.3,-)
Caja3: (-0.8,+), (-0.4,-)
Caja4: (0.3,+), (1.4,+)
Caja5: (2.3,-), (3.1,-)

Seleccionamos caja 1 vara validar. Entrenamos con los ejemplos de las cajas 2, 3, 4 y 5

	VECINO1	**VECINO2**	**VECINO3**	**PREDICCION**	
(-3.5,+)	(-2.1,-)	(-1.3,-)	(-0.8,+)	**-**	**FALLO**
(-2.7,+)	(-2.1,-)	(-1.3,-)	(-0.8,+)	**-**	**FALLO**

Seleccionamos caja 2 vara validar. Entrenamos con los ejemplos de las cajas 1, 3, 4 y 5

	VECINO1	**VECINO2**	**VECINO3**	**PREDICCION**	
(-2.1,-)	(-3.5,+)	(-2.7,+)	(-0.8,+)	**+**	**FALLO**
(-1.3,-)	(-0.8,+)	(-0.4,-)	(0.3, +)	**+**	**FALLO**

Seleccionamos caja 3 vara validar. Entrenamos con los ejemplos de las cajas 1, 2, 4 y 5

	VECINO1	**VECINO2**	**VECINO3**	**PREDICCION**	
(-0.8,+)	(-1.3,-)	(0.3, +)	(-2.1,-)	**-**	**FALLO**
(-0.4, +)	(-1.3,-)	(0.3,+)	(-2.1, +)	**+**	**ACIERTO**

Seleccionamos caja 4 vara validar. Entrenamos con los ejemplos de las cajas 1, 2, 3 y 5

	VECINO1	VECINO2	VECINO3	PREDICCION	
(0.3,+)	(-0.4, +)	(-0.8,+)	(-1.3,-)	+	ACIERTO
(1.4, +)	(2.3,-)	(-0.4,-)	(3.1, -)	-	FALLO

Seleccionamos caja 5 vara validar. Entrenamos con los ejemplos de las cajas 1, 2, 3 y 4

	VECINO1	VECINO2	VECINO3	PREDICCION	
(2.3,-)	(-0.4, -)	(0.3,+)	(1.4,+)	+	FALLO
(3.1, -)	(-0.4, -)	(0.3,+)	(1.4,+)	+	FALLO

Como se puede observar solo acierta en 2 de los 10 ejemplos

11.- [1.25 puntos] Considera un perceptrón y dos entradas x_1 y x_2 donde el valor de la entrada de sesgo es 1 y los pesos asignados a cada entrada son ($w_0 = 2$, $w_1 = 1$, $w_2 = -1$). La función de activación de la neurona es la función signo, de manera que los ejemplos se clasifican como positivos si el valor en el hiperplano es positivo o cero y como negativo en otro caso.

a) Escribe la ecuación del hiperplano separador inicial e indica cual es la clasificación que daría para el punto (−1, 2).
b) Supongamos que el punto (-1, 0) es de la clase negativa. Dados los pesos anteriores aplica la regla del perceptrón para que se clasifique correctamente y da la ecuación del hiperplano resultante usando una tasa de aprendizaje de 0,5.

La ecuación del hiperplano de separación inicial viene dada por $h(x_1, x_2)=x_1- x_2+2$
Evaluamos esta ecuación en el punto (-1,2), obteniendo h(-1,2)=-1-2+2<0, así que el ejemplo se clasifica como negativo.

h(-1, 0)=-1+2=1 >0. Luego el perceptrón diría que es de la clase positiva.
Actualizamos pesos (el superíndice 1 indica los pesos de la iteración anterior)

$w_1^2 = w_1^1 - 2\eta$ *(-1)=1+1=2*

$w_2^2 = w_2^1 - 2\eta$ *0=-1*

$w_0^2 = w_0^2 - 2\eta$ *1=2-1=1*

$P(x_1,\dots,x_i)$=$\sum_{x_{i+1},\dots,x_n} P(x_1,\dots,x_i,x_{i+1},\dots,x_n)$= $\sum_{x_{i+1}} P(x_1,\dots,x_i,x_{i+1})$		
$P(x\|y) \equiv P(X = x\|Y = y)$ = $\frac{P(x,y)}{P(y)}$	$P(X,Y) = P(X\|Y)P(Y)$ =$P(Y\|X)P(X)$	
$P(Y)$=$\sum_{i=1}^{m} P(Y\|x_i)P(x_i)$)	$P(X_1 = x_1,\dots,X_n = x_n)$ = $\prod_{i=1}^{n} P(X_i = x_i\|X_{i+1} = x_{i+1},\dots,X_n = x_n)$	
$P(X\|Y) = P(X)$	$P(X\|Y,K) = P(X\|K)$	$P(X\|Y,K) \neq P(X\|K)$
$P(A\|MB(A),B) = P(A\|MB(A))$	$P(x_1,\dots,x_n)$ = $\prod_{i=1}^{n} P(X_i = x_i\|Padres(X_i))$	
	$P(X\|Padres(X)) = P(X\|Padres(X), NoDescendientes(X))$	
$E(\boldsymbol{w}) = \frac{1}{2}\sum_d\sum_k (y_{kd} - o_{kd})^2$	$\delta_k = o_k(1-o_k)(y_k - o_k)$ $\delta_h = o_h(1-o_h)\sum_k w_{kh}\delta_k$	$\Delta w_i = \eta(y_d - \boldsymbol{w}\boldsymbol{x}_d)x_{di}$ $g(x) = \frac{1}{1+e^{-x}}$
$dist(\vec{x},\vec{y}) = \left(\sum_i \|x_i - y_i\|^p\right)^{1/p}$	$E(\boldsymbol{w}) = \frac{1}{2}(y_d - \boldsymbol{w}\boldsymbol{x}_d)^2$	$g(\cdot) = \begin{cases} 1 \text{ si } \boldsymbol{wx}>0 \\ -1 \text{ en otro caso} \end{cases}$

$H(X) = -\sum_{i=1}^{k} p_i \, log_2 \, p_i$	$H(X\|Y) = \sum_{y \in Y} p(y) H(X\|Y = y)$	$K(x_i, x_j) = \phi(x_i)\phi(x_j)$ $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{xy} + 1)^d$
$b = \frac{1}{\|VS\|} \sum_{i \in VS} y_i - \boldsymbol{w}\boldsymbol{x_i}$	$\boldsymbol{w} = \sum_{i \in VS} \alpha_i y_i \boldsymbol{x_i}$	$d = \frac{2}{\\|\boldsymbol{w}\\|}$