

Procesamiento de Lenguaje Natural

Apunte de Clases (Borrador)

Felipe Bravo Márquez

Felipe Bravo Márquez

Ilustración Portada por Paulette Filla

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN, UNIVERSIDAD DE CHILE

GITHUB.COM/DCCUCHILE/CC6205

Apuntes de clases del curso de Procesamiento de Lenguaje Natural de la Universidad de Chile.

El formato del apunte fue tomado del template de Jasmine Hao.

Borrador, 17 de enero de 2024



Índice general

0.1 Ejemplos de Problemas de Clasificación	4
0.2 Clasificador Naïve Bayes	6
0.2.1 Suposiciones de Independencia del Bayes Ingenuo Multinomial	7
0.2.2 Clasificador Bayes Ingenuo Multinomial	7
0.2.3 Aplicación de los clasificadores Naive Bayes multinomiales a la clasificación de texto	8
0.2.4 Problemas al multiplicar muchas probabilidades	8
0.2.5 Aprendizaje del modelo Naive Bayes multinomial	8
0.2.6 Estimación de parámetros	9
0.2.7 Probabilidades cero y el problema de las palabras no vistas	9
0.2.8 Suavizado Laplaciano (Add-1) para Naïve Bayes	9
0.2.9 Naïve Bayes multinomial: aprendizaje	10
0.2.10 Palabras desconocidas	10
0.3 Ejemplo	10
0.4 Naive Bayes como modelo de lenguaje	11
0.5 Evaluación	12
0.5.1 La Matriz de Confusión 2x2	12
0.5.2 Evaluación: Exactitud	12
0.5.3 Evaluación: Precisión y Recall	13
0.5.4 ¿Por qué Precisión y Recall?	13
0.5.5 Una Medida Combinada: Medida F	13
0.5.6 Conjuntos de Prueba de Desarrollo ("Devsets")	14
0.5.7 Validación Cruzada: Múltiples Divisiones	14
0.6 Conjuntos de entrenamiento, prueba y validación	15
0.6.1 Matriz de Confusión para clasificación de 3 clases	16

0.1 Evaluación

Una parte muy relevante en la construcción de cualquier sistema de PLN es su evaluación. Por lo general muchos distintos métodos y modelos son comparados en el proceso de entrenamiento para poder decidir qué modelo será puesto en producción. Para

- Consideraremos solo tareas de clasificación de texto binario.
- Imagina que eres el CEO de Delicious Pie Company.
- Quieres saber lo que la gente está diciendo sobre tus pasteles.
- Por lo tanto, construyes un detector de tweets de "Delicious Pie" con las siguientes clases:
 - Clase positiva: tweets sobre Delicious Pie Co.
 - Clase negativa: todos los demás tweets.

0.1.1 La Matriz de Confusión 2x2

	Sistema Positivo	Sistema Negativo
Oro Positivo	Verdadero Positivo (VP)	Falso Negativo (FN)
Oro Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

Recall (también conocido como **Sensibilidad** o **Tasa de Verdaderos Positivos**):

$$\text{Recall} = \frac{VP}{VP + FN}$$

Precisión:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Exactitud:

$$\text{Exactitud} = \frac{VP + VN}{VP + FP + VN + FN}$$

0.1.2 Evaluación: Exactitud

¿Por qué no usamos la exactitud como nuestra métrica?

Imagina que vimos 1 millón de tweets:

- 100 de ellos hablaban sobre Delicious Pie Co.
- 999,900 hablaban de otra cosa.

Podríamos construir un clasificador tonto que simplemente etiquete todos los tweets como "no sobre pasteles":

- ¡¡¡Obtendría una exactitud del 99.99 %!!! ¡¡¡Wow!!!
- ¡Pero sería inútil! ¡No devuelve los comentarios que estamos buscando!

Por eso usamos precisión y recall en su lugar.

0.1.3 Evaluación: Precisión y Recall

Precisión mide el porcentaje de elementos que el sistema detectó (es decir, los elementos que el sistema etiquetó como positivos) que son realmente positivos (según las etiquetas de oro humanas).

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Recall mide el porcentaje de elementos que el sistema identificó correctamente de todos los elementos que deberían haber sido identificados.

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

0.1.4 ¿Por qué Precisión y Recall?

Considera nuestro clasificador tonto de pasteles que simplemente etiqueta nada como "sobre pasteles".

- Exactitud = 99.99 % (etiqueta correctamente la mayoría de los tweets como no relacionados con pasteles)
 - Recall = 0 (no detecta ninguno de los 100 tweets relacionados con pasteles)
- La precisión y el recall, a diferencia de la exactitud, enfatizan los verdaderos positivos:
- Se centran en encontrar las cosas que se supone que debemos buscar.

0.1.5 Una Medida Combinada: Medida F

La medida F es un número único que combina la precisión (P) y el recall (R), definida como:

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

La medida F, definida con el parámetro β , pondera diferencialmente la importancia del recall y la precisión.

- $\beta > 1$ favorece al recall
- $\beta < 1$ favorece a la precisión

Cuando $\beta = 1$, la precisión y el recall son iguales, y tenemos la medida F_1 equilibrada:

$$F_1 = \frac{2PR}{P+R}$$

0.1.6 Conjuntos de Prueba de Desarrollo ("Devsets")

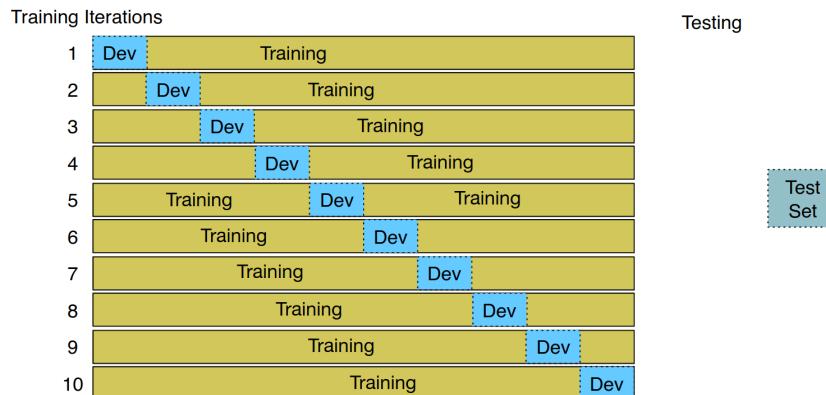
- Para evitar el sobreajuste y proporcionar una estimación más conservadora del rendimiento, comúnmente utilizamos un enfoque de tres conjuntos: conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba.



- **Conjunto de entrenamiento:** Se utiliza para entrenar el modelo.
- **Conjunto de desarrollo:** Se utiliza para ajustar el modelo y seleccionar los mejores hiperparámetros.
- **Conjunto de prueba:** Se utiliza para informar el rendimiento final del modelo.
- Este enfoque garantiza que el modelo no esté ajustado específicamente al conjunto de prueba, evitando el sobreajuste.
- Sin embargo, crea una paradoja: queremos la mayor cantidad de datos posible para el entrenamiento, pero también para el conjunto de desarrollo.
- ¿Cómo dividimos los datos?

0.1.7 Validación Cruzada: Múltiples Divisiones

- La validación cruzada nos permite utilizar todos nuestros datos para el entrenamiento y la prueba sin tener un conjunto de entrenamiento, conjunto de desarrollo y conjunto de prueba fijos.
- Elegimos un número k y dividimos nuestros datos en k subconjuntos disjuntos llamados pliegues.
- En cada iteración, uno de los pliegues se selecciona como conjunto de prueba mientras que los $k - 1$ pliegues restantes se utilizan para entrenar el clasificador.
- Calculamos la tasa de error en el conjunto de prueba y repetimos este proceso k veces.
- Finalmente, promediamos las tasas de error de estas k ejecuciones para obtener una tasa de error promedio.
- Por ejemplo, la validación cruzada de 10 pliegues implica entrenar 10 modelos con el 90 % de los datos y probar cada modelo por separado.
- Las tasas de error resultantes se promedian para obtener la estimación final del rendimiento.
- Sin embargo, la validación cruzada requiere que todo el corpus sea ciego, lo que impide examinar los datos para sugerir características o comprender el comportamiento del sistema.
- Para abordar esto, se crea un conjunto de entrenamiento y un conjunto de prueba fijos, y se realiza la validación cruzada de 10 pliegues dentro del conjunto de entrenamiento.
- La tasa de error se calcula convencionalmente en el conjunto de prueba.



0.2 Conjuntos de entrenamiento, prueba y validación

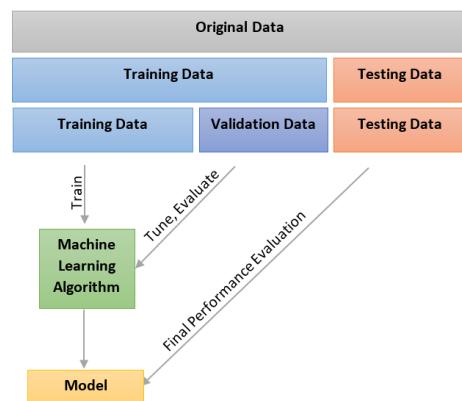
Cuando entrenamos un modelo, nuestro objetivo es producir una función $f(\vec{x})$ que mapee correctamente las entradas \vec{x} a las salidas \hat{y} según lo evidenciado por el conjunto de entrenamiento. La evaluación del rendimiento en los datos de entrenamiento puede ser engañosa, ya que nuestro objetivo es entrenar una función capaz de generalizar a ejemplos no vistos. Una forma común de abordar esto es dividir el conjunto de entrenamiento en subconjuntos de entrenamiento y prueba (80 % y 20 % respectivamente). Se entrena el modelo en el subconjunto de entrenamiento y se calcula la precisión en el subconjunto de prueba.

Sin embargo, este enfoque tiene una limitación. En la práctica, a menudo se entrenan varios modelos, se comparan sus calidades y se selecciona el mejor. Si se selecciona el mejor modelo en función de la precisión en el subconjunto de prueba, se obtendrá una estimación excesivamente optimista de la calidad del modelo. No se sabe si la configuración elegida del clasificador final es

bueno en general o simplemente bueno para los ejemplos particulares en los subconjuntos de prueba.

La metodología aceptada es utilizar una división de tres vías de los datos en conjuntos de entrenamiento, validación (también llamado desarrollo) y prueba¹. Esto proporciona dos conjuntos apartados: un conjunto de validación (también llamado conjunto de desarrollo) y un conjunto de prueba. Todos los experimentos, ajustes, análisis de errores y selección de modelos deben realizarse

basados en el conjunto de validación. Luego, una única ejecución del modelo final sobre el conjunto de prueba proporcionará una buena estimación de su calidad esperada en ejemplos no vistos. Es importante mantener el conjunto de prueba lo más limpio posible, realizando la menor cantidad de experimentos posible en él. Incluso algunos defienden que no se deben mirar siquiera los ejemplos en el conjunto de prueba, para evitar sesgar el diseño del modelo.



0.2.1 Matriz de Confusión para clasificación de 3 clases

			gold labels		
			urgent	normal	spam
system output	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$ $\text{precision}_n = \frac{60}{5+60+50}$ $\text{precision}_s = \frac{200}{3+30+200}$
	normal	5	60	50	
	spam	3	30	200	
			$\text{recall}_u = \frac{8}{8+5+3}$ $\text{recall}_n = \frac{60}{10+60+30}$ $\text{recall}_s = \frac{200}{1+50+200}$		

Cómo combinar métricas binarias (Precisión, Recall, F_1) de más de 2 clases para obtener una métrica única:

- Macro-promedio:
 - Calcular las métricas de rendimiento (Precisión, Recall, F_1) para cada clase individualmente.

¹Un enfoque alternativo es la validación cruzada, pero no se escala bien para entrenar redes neuronales profundas.

¹Fuente: <https://www.codeproject.com/KB/AI/1146582/validation.PNG>

- Promediar las métricas en todas las clases.
- Micro-promedio:
 - Recopilar las decisiones para todas las clases en una matriz de confusión.
 - Calcular la Precisión y el Recall a partir de la matriz de confusión.

		Class 1: Urgent		Class 2: Normal		Class 3: Spam		Pooled	
		true	true	true	true	true	true	true	true
		urgent	not	normal	not	spam	not	yes	no
system	urgent	8	11	system	60	55	system	200	33
	not	8	340		40	212		51	83

$\text{precision} = \frac{8}{8+11} = .42$

 $\text{precision} = \frac{60}{60+55} = .52$

 $\text{precision} = \frac{200}{200+33} = .86$

 $\text{microaverage precision} = \frac{268}{268+99} = .73$

$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$



[McCallum et al., 1998] McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI.

[Mosteller and Wallace, 1963] Mosteller, F. and Wallace, D. L. (1963). Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.