Clasificación

Rubén Francisco Manrique
rf.manrique@uniandes.edu.co

Contenido

1. Formalización del problema de clasificación
2. Clasificadores generativos vs discriminativos
3. Naive Bayes: Simple pero muy usado
4. Regresión logística
    - Componentes del aprendizaje (función de perdida y algoritmo de optimización).
    - Sobreajuste y regularización.

Clasificación de texto

- Asignar categorías, tópicos, géneros.
- Detección de spam.
- Identificación de autoría.
- Identificación de edad/genero.
- Identificación de lenguaje
- Análisis de sentimientos.

Clasificación de texto

- Entrada:
  - Un documento d
  - Un conjunto fijo de clases $C = \{c_1, c_2, \dots, c_j\}$.

- Salida:
  - Una predicción de clase $c \in C$.

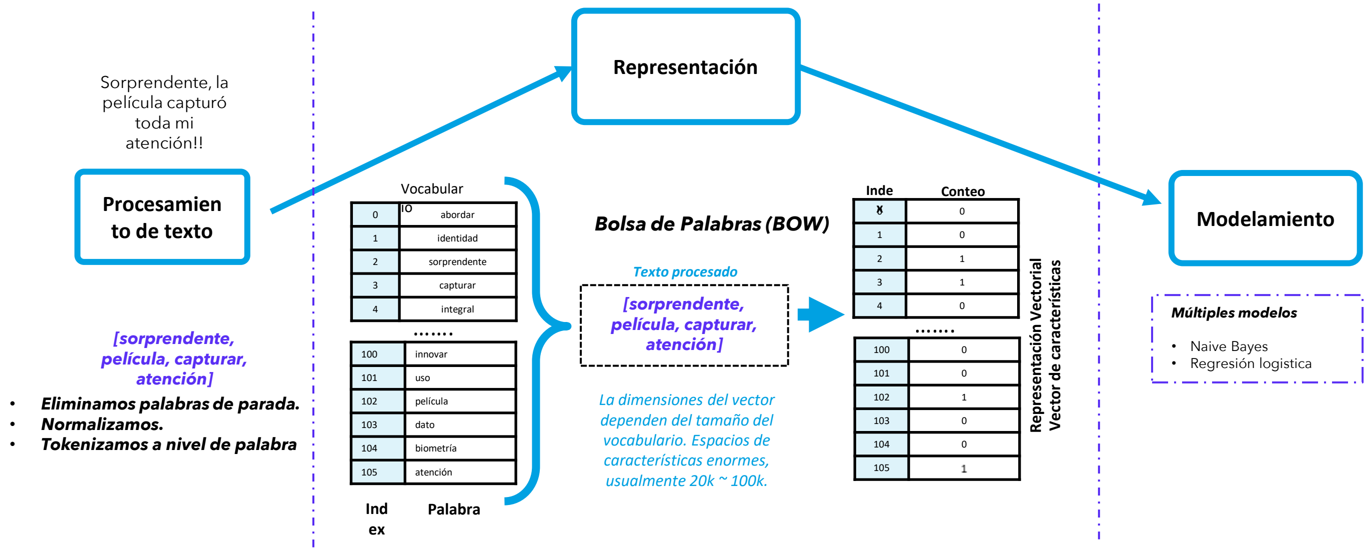
Clasificación de texto

Método Supervisado de Aprendizaje de Máquina

- Entrada:
  - Un documento d
  - Un conjunto fijo de clases $C = \{c_1, c_2, \dots, c_j\}$.
  - Un conjunto de entrenamiento m etiquetado manualmente $\{(d_1, c_1), (d_2, c_2), \dots, (d_m, c_m)\}$

- Salida:
  - Una función de clasificación $\gamma: d \to c$.

Clasificación: pipeline enfoque aprendizaje de máquina



Necesito muchos datos etiquetados

Clasificadores generativos y discriminatorios

Clasificadores generativos

- Construye un modelo de lo que hay en la imagen de un gato
  - Sabe sobre bigotes, orejas, ojos.
  - Asigna una probabilidad a cualquier imagen:
  - **¿Qué tan felina es esta imagen?**
- También se construye un modelo de perros.

- Ahora dada una nueva imagen:
  - **Ejecute ambos modelos y vea cuál se ajusta mejor**

Clasificadores discriminativos

- Discriminar: “IDENTIFICAR DIFERENCIAS”

- Mira!, ¡los perros tienen collar!, y los gatos no. Ignoremos todo lo demás.

Clasificadores generativos y discriminatorios

- Naïve Bayes es un clasificador generativo.

VS

- La regresión logística es un clasificador discriminativo.

Naïve Bayes

Clasificador Naive Bayes

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

Clasificador Naive Bayes

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

$$c_{MAP} = argmax_{c \in C} \, P(x_1, x_2, x_3, x_4 \ldots x_n | c) P(c)$$

En el modelo de bolsa de palabras el documento d es representado por un conjunto de términos $x_1, x_2, x_3, x_4 \ldots x_n$. Dado que estamos en ML aprendizaje supervisados estos términos se suelen denominar “features/caracteristicas”

Clasificador Naive Bayes Multinomial

$$c_{MAP} = argmax_{c \, \epsilon \, C} \, P(x_1, x_2, x_3, x_4 \ldots x_n | c) P(c)$$

Difícil de estimar, solo si se tiene un conjunto enorme de entrenamiento.

Que tan frecuente ocurre esta clase?

Estimación MLE, frecuencias relativas en el corpus.

Clasificador Naive Bayes Multinomial

$$c_{MAP} = argmax_{c \in C} \, P(x_1, x_2, x_3, x_4 \ldots x_n|c)P(c)$$

- Asunción BOW: Posición no interesa.
- Independencia condicional: Asume que la probabilidad de cada feature/termino $P(x_i|c_j)$ es independiente dada una clase c.

$$P(x_1, x_2, x_3, x_4 \ldots x_n|c) = P(x_1|c)P(x_2|c)P(x_2|c) \ldots P(x_n|c)$$

Clasificador Naive Bayes Multinomial

$$c_{MAP} = argmax_{c \, \epsilon \, C} \, P(x_1, x_2, x_3, x_4 \dots x_n | c) P(c)$$

$$c_{NB} = argmax_{c \, \epsilon \, C} \, P(c) \prod_{x \epsilon \, X} P(x|c)$$

Clasificador Naive Bayes Multinomial: Entrenamiento

- A la fecha hemos aprendido a estimar probabilidades usando el principio MLE.
  - Simplemente use las frecuencias en los datos.

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Número de veces que la palabra w_j aparece

Entre todas las palabras en documentos de la clase c_j

¡Problema con MLE!, si el mismo de siempre!

- Qué pasa si aparece una palabra no vista en el conjunto de datos.

- Suavizado Suma-1 (Laplace)

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c)}{\sum_{w \in V} \left(count(w, c)\right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Clasificador Multinomial Naive Bayes

Reseña
(d)

?

Negativo
(c_1)

Corpus Documentos Negativos

D1. Terrible **película**
D2. **Espantosa**
D3. Predecible, **capturaron** al villano
D4. **Sorprendente**, no me gusto ni la música

Positivo
(c_2)

Corpus Documentos Positivos

D1. **Sorprendente** y maravillosa
D2. Espectacular, **sorprendente** tienen que ver esta **película**
D3. **Capturó** toda mi atención, es **sorprendente**

Probabilidades a priori

$$P(c_1) = \frac{4}{7} \approx 0{,}57$$

$$P(c_2) = \frac{3}{7} \approx 0{,}43$$

Vocabulario: **[sorprendente, película, capturar, espantosa]**

Paso 1: Calcular probabilidades condicionales

Corpus Documentos Negativos
(solo se preservan palabras que estén en el vocabulario)

Negativo (c_1)

D1. película
D2. espantosa
D3. capturar
D4. sorprendente

Sorprendente
$P(\text{“Sorprendente”}|c_1) = 1/4$

Película
$P(\text{“Película”}|c_1) = 1/4$

Espantosa
$P(\text{“Espantosa”}|c_1) = 1/4$

Capturar
$P(\text{“Capturar”}|c_1) = 1/4$

Corpus Documentos Positivos
(solo se preservan palabras que estén en el vocabulario)

Positivo (c_2)

D1. sorprendente
D2. sorprendente, película
D3. capturar, sorprendente

Sorprendente
$P(\text{“Sorprendente”}|c_2) = 3/5$

Película
$P(\text{“Película”}|c_2) = 1/5$

Capturar
$P(\text{“Película”}|c_2) = 1/5$

Vocabulario: [sorprendente, película, capturar, espantosa]

Paso 2: Estimar la clase con la mayor probabilidad a posteriori (I)

Negativo (c_1)

$$\begin{cases} P(c_1) = 4/7 \\ \boldsymbol{P(\text{“}Película\text{”}|c_1) = 1/4} \\ P(\text{“}Espantosa\text{”}|c_1) = 1/4 \\ P(\text{“}Capturar\text{”}|c_1) = 1/4 \\ P(\text{“}Sorpredente\text{”}|c_1) = 1/4 \end{cases}$$

Positivo (c_2)

$$\begin{cases} P(c_2) = 3/7 \\ \boldsymbol{P(\text{“}Película\text{”}|c_2) = 1/5} \\ P(\text{“}Sorpredente\text{”}|c_2) = 3/5 \\ P(\text{“}Película\text{”}|c_2) = 1/5 \\ P(\text{“}Espantosa\text{”}|c_2) = 0 \end{cases}$$

d: Que película tan sorprendente!!

d: [película,sorpredente]

$$c = \underset{c \in \{c_1, c_2\}}{\operatorname{argmax}} P(c|d)$$

Teorema de Bayes

$$c = \underset{c \in \{c_1, c_2\}}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

$$\boxed{c = \underset{c \in \{c_1, c_2\}}{\operatorname{argmax}} P(d|c)P(c)}$$

$\boxed{P(d|c)}$?

$$\boxed{P(película, sorpredente|c)}$$

1. Representación de bolsa de palabras (el orden no importa)
2. Ingenuamente asume que las ocurrencias de las palabras son eventos independientes (independencia condicional).

$$\boxed{c = \underset{c \in \{c_1, c_2\}}{\operatorname{argmax}} P(película|c)P(sorprendente|c)P(c)}$$

Fórmula Final

Vocabulario: **[sorprendente, película, capturar, espantosa]**

Paso 2: Estimar la clase con la mayor probabilidad a posteriori (II)

Negativo (c_1)

$P(c_1) = 4/7$

$\boldsymbol{P(\text{“}Película\text{”}|c_1) = 1/4}$

$P(\text{“}Espantosa\text{”}|c_1) = 1/4$

$P(\text{“}Capturar\text{”}|c_1) = 1/4$

$P(\text{“}Sorpredente\text{”}|c_1) = 1/4$

Positivo (c_2)

$P(c_2) = 3/7$

$\boldsymbol{P(\text{“}Película\text{”}|c_2) = 1/5}$

$P(\text{“}Sorpredente\text{”}|c_2) = 3/5$

$P(\text{“}Película\text{”}|c_2) = 1/5$

$P(\text{“}Espantosa\text{”}|c_2) = 0$

d: ¡¡Que película tan sorprendente!!

d: [película,sorpredente]

$$\boxed{c = \underset{c \in \{c_1, c_2\}}{\operatorname{argmax}} P(película|c)P(sorprendente|c)P(c)}$$

Negativo (c_1)

$$P(película|c_1)P(sorprendente|c_1)P(c_1) = \left(\frac{1}{4}\right)\left(\frac{1}{4}\right)\left(\frac{4}{7}\right) \approx 0{,}0357$$

Positivo (c_2)

$$P(película|c_2)P(sorprendente|c_2)P(c_2) = \left(\frac{1}{5}\right)\left(\frac{3}{5}\right)\left(\frac{3}{7}\right) \approx 0{,}0515$$

Vocabulario: [sorprendente, película, capturar, espantosa]

Regresión logística

Regresión logística

- Importante herramienta analítica en ciencias naturales y sociales.
- Frecuentemente usado como baseline en aprendizaje automático supervisado.
- Es también la base de rede neuronales.

Componentes de un clasificador de aprendizaje automático probabilístico

Dados m pares de entrada/salida $(x^{(i)}, y^{(i)})$:

1. Una representación característica de la entrada (feature representation). Para cada observación de entrada $x^{(i)}$, un vector de características $[x_1, x_2, \dots, x_n]$. La característica i para la entrada $x^{(j)}$ es x_i, o más específicamente $x_i^{(j)}$.
2. Una función de clasificación que computa $\hat{y}$, la salida estimada, vía $p(y|x)$. Ejemplos: función sigmoidal y softmax.
3. Una función objetivo de aprendizaje (función de loss). Ejemplo: cross-entropy-loss.
4. Un algoritmo para optimización de la función objetivo: gradiente descendiente estocástico.

Las dos fases de la regresión logística

Entrenamiento (Training): aprendemos los pesos $\boldsymbol{w}$ y $\boldsymbol{b}$ usando el descenso de gradiente estocástico y el cross-entropy loss.

Evaluación (Test): Dado un ejemplo de prueba $\boldsymbol{x}$, calculamos $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})$ usando los pesos aprendidos $\boldsymbol{w}$ y $\boldsymbol{b}$, y devolvemos la etiqueta ($y = 1$ o $y = 0$) que tenga mayor probabilidad.

Clasificación de texto con regresión logística

Entrada:

Documentos con su respectiva etiqueta. Supongamos un **caso binario**: positivo/negativo, spam/no spam, etc.

$(x^{(i)}, y^{(i)})$ Observaciones/ejemplos de entranamiento

Para cada observación:

- Representamos $x^{(i)}$ por un vector de caracteristicas $[x_1, x_2, \ldots, x_n]$
- Computamos una salida: la predicción de la clase $\hat{y}^{(i)} \in \{0,1\}$

Características (Features) en regresión logística

Para la característica x_i, el peso w_i nos indica que tan importante es x_i:

Suponga por ejemplo las siguientes caracteristicas:

- x_1="documento contiene la palabra 'asombroso'": w_1=+10
- x_2="documento contiene la palabra 'pesima'": w_2=-10
- x_3="documento contiene la palabra 'regular'": w_2=-2

En resumen....para una observación $\boldsymbol{x}^{(\boldsymbol{i})}$

Tenemos:

- Vector de características: $\mathbf{x} = [x_1, x_2, \dots, x_n]$
- Pesos (uno por cada caracteristica): $\mathbf{w} = [w_1, w_2, \dots, w_n]$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$
- Predicción: $\hat{y}^{(i)} \in \{0,1\}$
    - Multinomial: $\hat{y}^{(i)} \in \{0,1,2,3,4,..\}$

¿Como se hace la clasificación?

- Cada característica x_i tiene un peso w_i que nos indica la importancia de x_i.

- Idea: suma ponderada de las características más un bias (un hiperplano):

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

$$z = \boldsymbol{w}\boldsymbol{x} + b$$

- Si esta suma *es alta*, decimos $\hat{y} = 1$; si *es baja*, entonces $\hat{y}$=0

Necesitamos una probabilidad

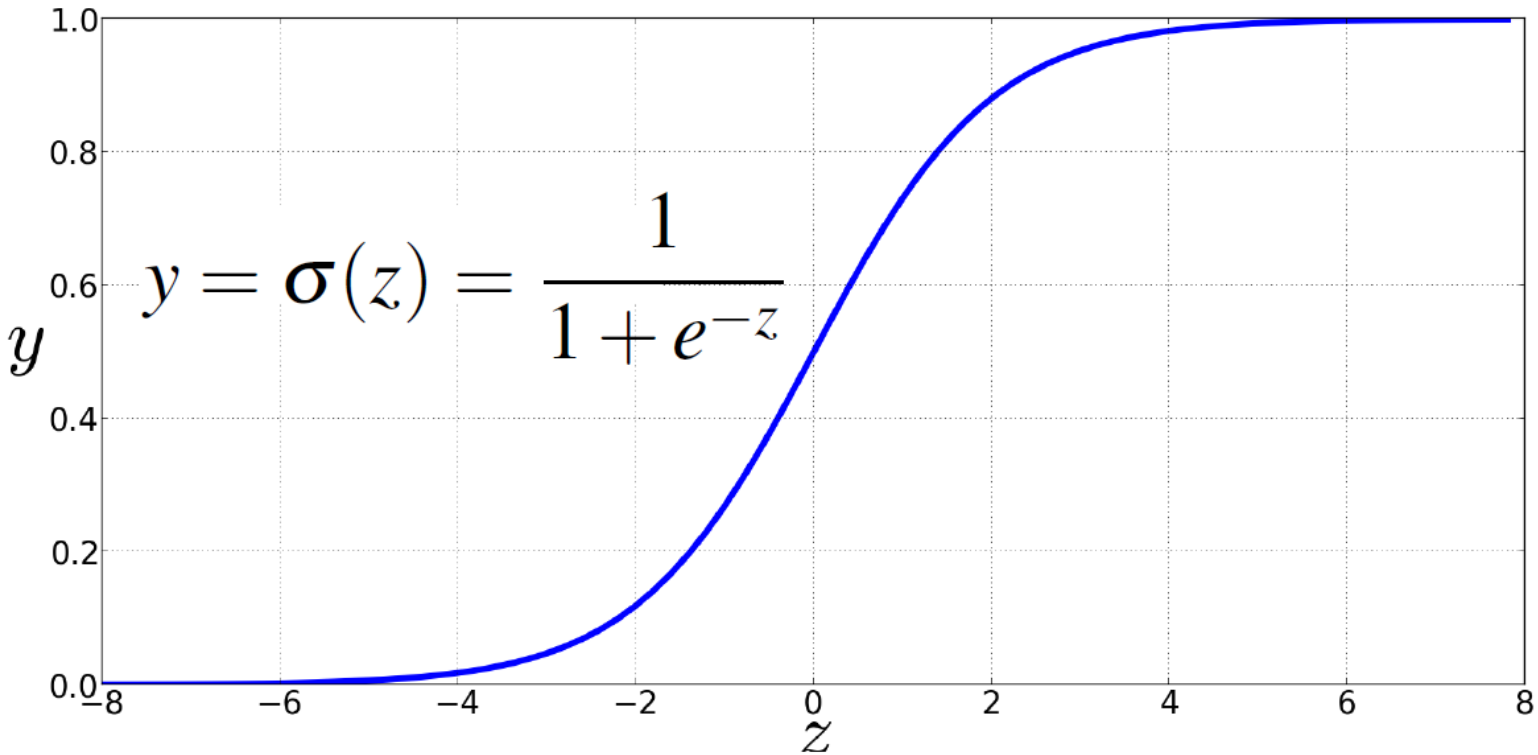
- Decir “suma alta” o “suma baja” es demasiado informal. Por otro lado, buscamos un clasificador que nos arroje una probabilidad:

$$p(y = 1|\boldsymbol{x}; \boldsymbol{w})$$
$$p(y = 0|\boldsymbol{x}; \boldsymbol{w})$$

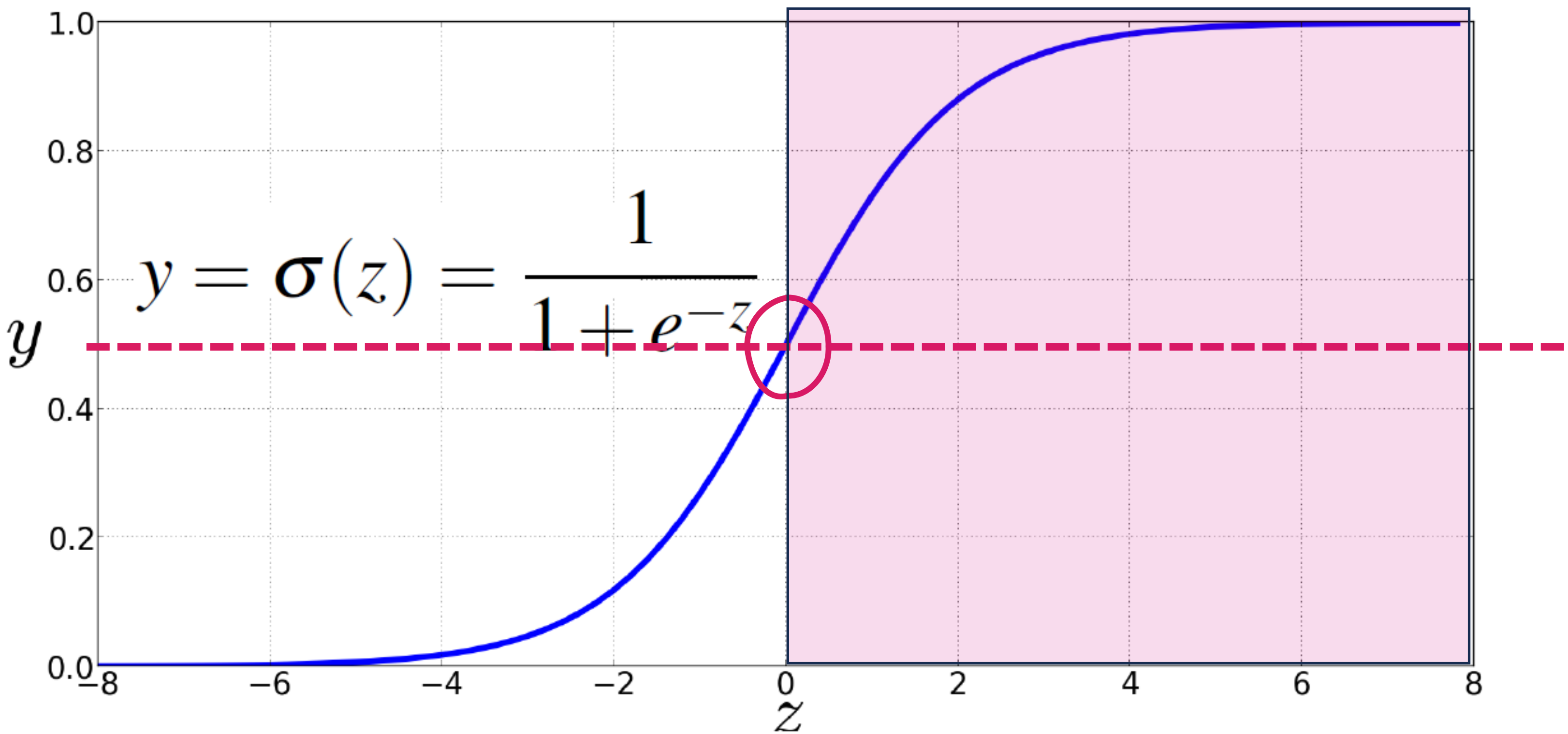
- El problema es que nuestra suma ponderada, $z = \boldsymbol{wx} + b$, es solo un número. Como lo transformamos en una probabilidad?
- **Solución:** use una función para transformar z en un valor que va de 0 a 1.

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Función sigmoidal



De una probabilidad a un clasificador



$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$

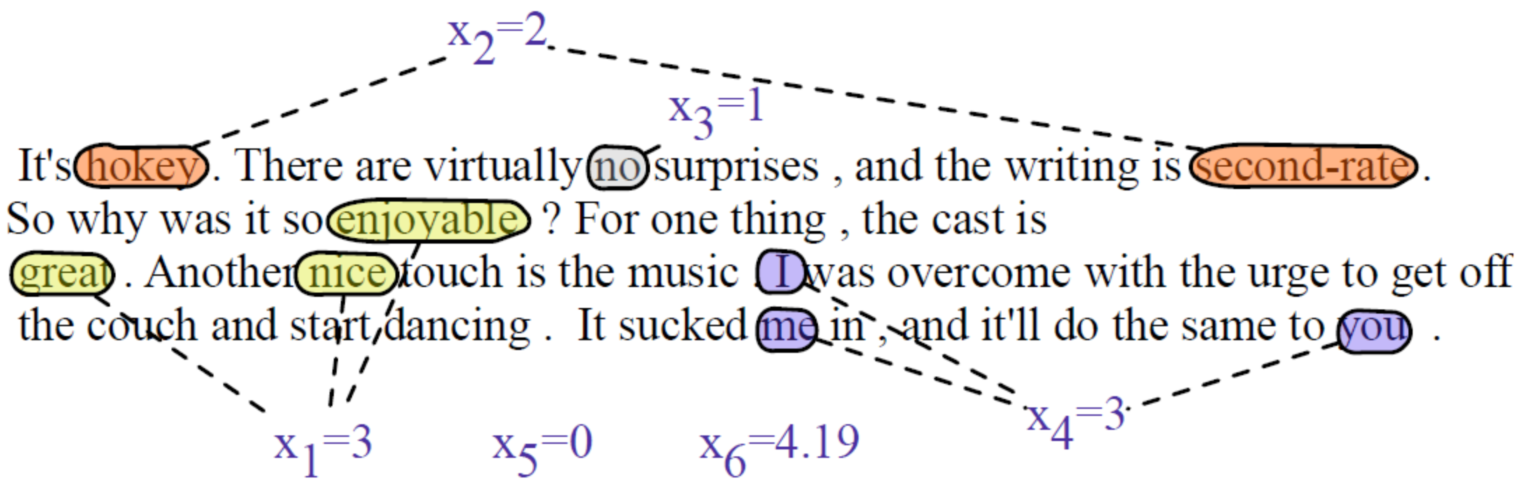
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{l} \text{if } w \cdot x + b > 0 \\ \text{if } w \cdot x + b \leq 0 \end{array}$$

Veamos un ejemplo: análisis de sentimientos (tomado del libro)

It's hokey . There are virtually no surprises , and the writing is second-rate. So why was it so enjoyable ? For one thing , the cast is great . Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you.

Es cursi. Prácticamente no hay sorpresas, y la escritura es de segunda categoría. Entonces, ¿por qué fue tan agradable? Por un lado, el elenco es genial. Otro buen detalle es la música. Me invadieron las ganas de levantarme del sofá y empezar a bailar. Me absorbió y te hará lo mismo a ti.



Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Suponga el vector de pesos -> $w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$, $b = 0.1$

$$
\begin{aligned}
p(y = 1|\boldsymbol{x}; \boldsymbol{w}) &= \sigma(\boldsymbol{wx} + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7]\ [3,2,1,3,0,4.19] + 0.1) \\
&= \sigma(0.833) = 0.70
\end{aligned}
$$

$$p(y = 0|\boldsymbol{x}; \boldsymbol{w}) = 1 - \sigma(\boldsymbol{wx} + b) = 0.30$$

Por lo tanto, $\hat{y} = 1$

De donde vienen los pesos $\boldsymbol{W}$?

Aprendizaje supervisado: sabemos el label correcto y para cada x, y nuestro clasificador por regresión logistica produce una estimación $\hat{y}$.

Queremos los valores de $\boldsymbol{w}$ y b que minimizen la distancia entre nuestra estimación y el valor real.

- Necesitamos un estimador de distancia: *una función de perdida* (loss) o una *función de costo*.

- Necesitamos un *algoritmo de optimización* para actualizar $\boldsymbol{w}$ y b para minimizar el loss.

Componentes del aprendizaje

Función de loss:

- **Cross-entropy loss**

Algoritmo de optimización:

- **Gradiente descendiente estocástico.**

Cross-entropy loss (I)

Objetivo: maximizar la probabilidad del label correcto $p(y|\boldsymbol{x})$

Como solo hay dos posibles salidas discretas (0 o 1) se puede ver como un experimento de *Bernoulli*. Podemos expresar la probabilidad $p(y|\boldsymbol{x})$ con una función de probabilidad:

$$p(y|x) = \hat{y}^{y}(1-\hat{y})^{1-y}$$

Nótese que:

- Si $y = 1$, la ecuación se simplifica a $\hat{y}$
- Si $y = 0$, la ecuación se simplifica a $1 - \hat{y}$

Cross-entropy loss (II)

Objetivo: maximizar la probabilidad del label correcto $p(y|\boldsymbol{x})$

Maximizar: $p(y|x) = \hat{y}^{y}(1-\hat{y})^{1-y}$

Por facilidad en los exponentes apliquemos $\log$

$$\log p(y|x) = \log[\hat{y}^{y}(1-\hat{y})^{1-y}]$$
$$\log p(y|x) = y\log\hat{y} + (1-y)\log(1-\hat{y})$$

Cross-entropy loss (II)

Objetivo: maximizar la probabilidad del label correcto $p(y|\boldsymbol{x})$

Maximizar: $p(y|x) = \hat{y}^y(1-\hat{y})^{1-y}$

Por facilidad en los exponentes apliquemos $\log$

$$\log p(y|x) = \log[\hat{y}^y(1-\hat{y})^{1-y}]$$
$$\log p(y|x) = y\log\hat{y} + (1-y)\log(1-\hat{y})$$

Ahora transformemos el objetivo de maximizar en uno de minimización. ¿Como?, Por qué? .

Minimizar facilita las cosas para el algoritmo de optimización.

Simplemente invirtamos el signo.

Cross-entropy loss (III)

Minimizar: $-\log p(y|x) = -[y\log \hat{y} + (1-y)\log(1-\hat{y})]$

Mas ampliamente conocido como:

$$L_{CE}(y,\hat{y}) = -[y\log \hat{y} + (1-y)\log(1-\hat{y})]$$

Cross-entropy loss!!, expandiendo $\hat{y}$:

$$L_{CE}(y,\hat{y}) = -\left[y\log \sigma(\boldsymbol{wx}+b) + (1-y)\log\left(1-\sigma(\boldsymbol{wx}+b)\right)\right]$$

Cross-entropy loss (IV) – Retomemos el ejemplo

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if “no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Suponga el vector de pesos -> $w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$, $b = 0.1$

$$p(y = 1|\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{wx} + b)$$

$$= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7]\ [3,2,1,3,0,4.19] + 0.1)$$

$$= \sigma(0.833) = 0.70$$

Cross-entropy loss (IV) – Retomemos el ejemplo

Escenario 1: Supongamos el label real como y=1, en este caso el loss debería ser pequeño porque nuestro modelo está cerca a lo correcto.

$$L_{CE}(y,\hat{y}) = -\left[y\log\sigma(\boldsymbol{wx}+b) + (1-y)\log\left(1-\sigma(\boldsymbol{wx}+b)\right)\right]$$
$$L_{CE}(y,\hat{y}) = -[\log\sigma(\boldsymbol{wx}+b)]$$
$$L_{CE}(y,\hat{y}) = -\log 0.7 = 0.36$$

Escenario 2: Supongamos el label real como y=0, en este caso el loss debería ser grande porque nuestro modelo está lejos de lo correcto.

$$L_{CE}(y,\hat{y}) = -\left[y\log\sigma(\boldsymbol{wx}+b) + (1-y)\log\left(1-\sigma(\boldsymbol{wx}+b)\right)\right]$$
$$L_{CE}(y,\hat{y}) = -[\log(1-\sigma(\boldsymbol{wx}+b))]$$
$$L_{CE}(y,\hat{y}) = -\log 0.3 = 1.2$$

Gradiente descendiente estocástico (I)

Objetivo: minimizar la función de perdida (loss).

Queremos los pesos $\boldsymbol{\theta} = (\boldsymbol{w}, b)$ que minimizan el loss sobre todos los ejemplos:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{argmin} \frac{1}{m} \sum_{i=1}^{m} L_{CE}(y^{(i)}, \hat{y}^{(i)})$$

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{argmin} \frac{1}{m} \sum_{i=1}^{m} L_{CE}(y^{(i)}, f(x^{(i)}; \boldsymbol{\theta}))$$

Gradiente descendiente estocástico (II)

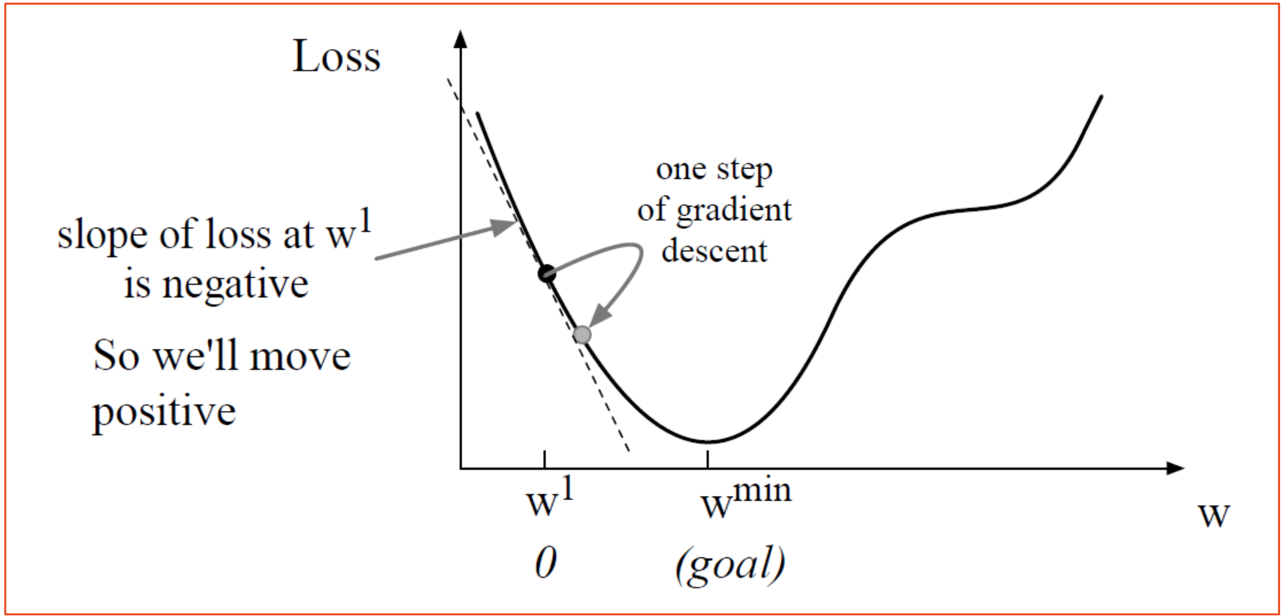
Objetivo: descender una montaña rápidamente.

Mirar 360 grados, encontrar la dirección de la pendiente más pronunciada, vaya por ese camino.

Gradiente descendiente estocástico (III)

Idea importante: la función de perdida (cross-entropy loss) en RL es convexa.

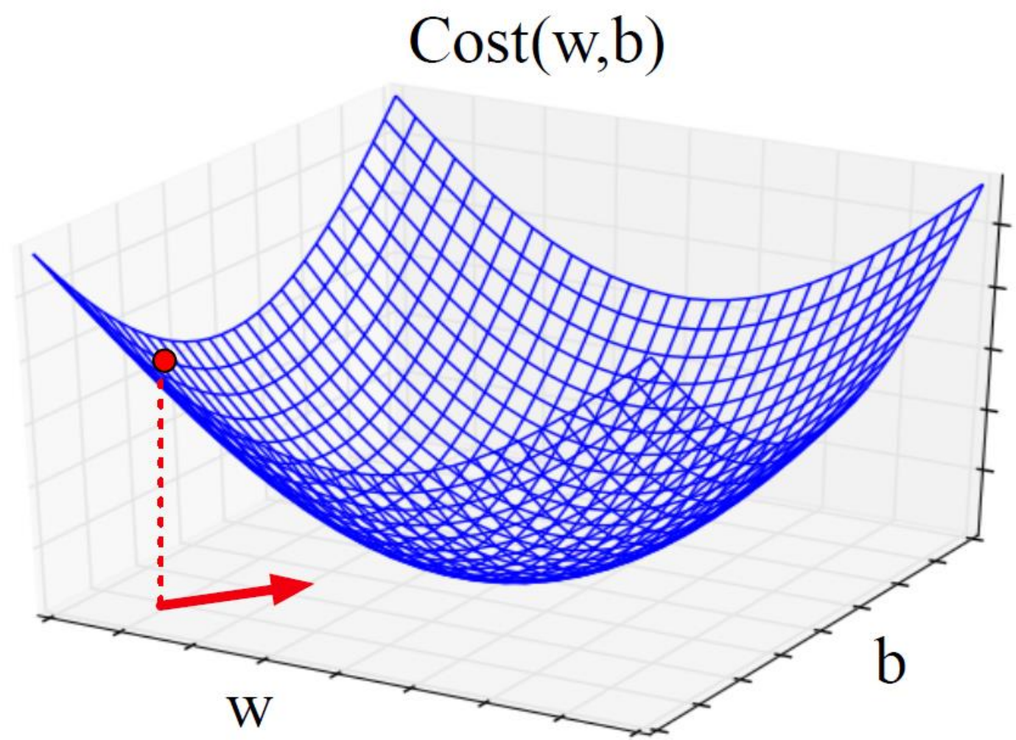
Una función convexa tiene solo un mínimo.
Que implica: el gradiente independiente de donde inicia va a encontrar el mínimo.



Dos dimensiones
(Un solo peso/parámetro)

Gradiente descendiente estocástico (IV)

Tres dimensiones
(Un peso y bias)



Gradiente descendiente estocástico (V)

El **gradiente** de una función de muchas variables es un vector apuntando en la dirección de mayor incremento de una función.

El **gradiente descendiente** es un algoritmo que encuentra el gradiente de la función de perdida (loss) y se mueve en la dirección opuesta.

¿Que es moverse? **– cambiar los valores de los parámetros $\boldsymbol{\theta} = (\boldsymbol{w}, b)$**, según:

$$w^{t+1} = w^t - \eta \frac{d}{dw} f(x^{(i)}; \boldsymbol{\theta})$$

η es la tasa de aprendizaje, entre más grande los cambios en los pesos son más grandes (**hiperparámetro**).

Gradiente descendiente estocástico (VI)

En la práctica nuestros modelos tienen muchos parámetros, el **gradiente** se suele definir con el operador nabla.

$$\nabla_{\theta} L(f(x;\theta),y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \end{bmatrix}$$

$$\boxed{\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta),y)}$$

Gradiente descendiente estocástico

¿Bueno, y cuál es la derivada de la función de perdida (loss)?

$$L_{\mathrm{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1-y) \log (1 - \sigma(w \cdot x + b))]$$

Pueden ver el proceso elegante de derivación en la sección 5.8 del libro (Speech and Language Processing (3rd ed. draft))

$$\frac{\partial L_{\mathrm{CE}}(\hat{y}, y)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j$$

GRADIENTE DESCENDIENTE ESTOCÁSTICO

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
  # where: L is the loss function
  #   f is a function parameterized by θ
  #   x is the set of training inputs $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$
  #   y is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \ldots, y^{(m)}$

$\theta \leftarrow 0$
repeat til done # see caption
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
      Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output $\hat{y}$?
      Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$) from the true outpu
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize l
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ

Entrenamiento en Batch. Mini-batch training (512,1024).

Por eso se llama estocástico.

Se definen un número de épocas – época es un completo por todos los ejemplos de entrenamiento.

Pequeño ejemplo: Un paso del gradiente descendiente (I)

Suponga un problema de clasificación de sentimientos.

“La película con elenco de tercera y bajo presupuesto, me logro sorprender, capturó toda mi atención y me hizo recordar esa niña feliz de los 80”

Suponga por simplicidad dos features extraídos de lexicones:

$$x_1 = 3 \; (\textit{número de palabras positivas en el lexicón})$$
$$x_2 = 2 \; (\textit{número de palabras negativas en el lexicón})$$

Asuma que inicializamos los pesos de la siguiente manera:

$$w_1 = w_2 = b = 0$$

Y la tasa de aprendizaje la fijamos en $\eta = 0.1$

La salida real es $y = 1$

Pequeño ejemplo: Un paso del gradiente descendiente (II)

$w_1 = w_2 = b = 0;$
$x_1 = 3; \quad x_2 = 2$

Calculamos la salida (predicción)

$$\sigma(\boldsymbol{wx} + b) = \sigma(0 + 0) = \sigma(0) = 0.5$$
$$\hat{y} = 0$$

Calculamos los gradientes

$$\nabla_{w,b} = \begin{bmatrix} \frac{\partial L_{\mathrm{CE}}(\hat{y},y)}{\partial w_1} \\ \frac{\partial L_{\mathrm{CE}}(\hat{y},y)}{\partial w_2} \\ \frac{\partial L_{\mathrm{CE}}(\hat{y},y)}{\partial b} \end{bmatrix} = \begin{bmatrix} (\sigma(w \cdot x + b) - y)x_1 \\ (\sigma(w \cdot x + b) - y)x_2 \\ \sigma(w \cdot x + b) - y \end{bmatrix} = \begin{bmatrix} (\sigma(0) - 1)x_1 \\ (\sigma(0) - 1)x_2 \\ \sigma(0) - 1 \end{bmatrix} = \begin{bmatrix} -0.5x_1 \\ -0.5x_2 \\ -0.5 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix}$$

Actualicemos los pesos de los parámetros

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta), y) \qquad \eta = 0.1;$$

$$\theta^1 = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} - \eta \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} .15 \\ .1 \\ .05 \end{bmatrix}$$

$$\mathrm{w}_1 = 0.15$$
$$\mathrm{w}_2 = 0.1$$
$$\mathrm{b} = 0.05$$

Gracias por la atención

¿Tiene alguna pregunta?