Introducción a Natural Language Processing

Modelos Generativos

Ronald Cárdenas Acosta

Setiembre, 2016

Outline

Probabilidades [Repaso]

- Definición frecuentista (clásica): $P(A) = lim_{N \rightarrow \infty} \frac{N_A}{N}$
  Frecuencia relativa de un evento A en un número infinito de pruebas
- Definición Subjetiva o Bayesiana: $P(A)$ es un grado de certidumbre
  Por ejemplo, le da sentido a P("mañana estará soleado")

Teorema de Bayes

Sean A y B dos variables randómicas dependientes entre sí

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Donde:

$$Posterior = \frac{Likelihood * Prior}{Evidencia}$$

- Posterior $P(A|B)$: Información sobre A inferida al conocer B.
- Likelihood $P(B|A)$: Chance de que B suceda luego de que A suceda.
- Prior $P(A)$: Información sobre A conocida de antemano
- Evidencia $P(B)$: Información observada, conocida.

Teorema de Bayes: Ejemplo

- Supongamos que tenemos dos variables: películas y libros.
- Tres tipos de película: Acción, Sci-fi, Romance
- Dos tipos de libros: Sci-fi, Romance
- Si se sabe que un objeto es de tipo sci-fi, ¿cuál es la probabilidad de que sea una película?

Teorema de Bayes: Ejemplo

$$P(pelicula|sci - fi) = \frac{P(scifi|pelicula) * P(pelicula)}{P(scifi)} \tag{1}$$

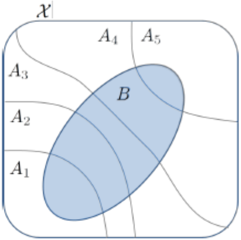
- Posterior: $P(pelicula|scifi)$
- Likelihood: $P(scifi|pelicula)$
- Prior: $P(movie)$
- Evidencia: $P(scifi)$

Ley de Probabilidad Total

Sea:

- X el espacio de posibles resultados de un experimento
- A,B: variables randómicas que representan subconjuntos de X
- A_i: valores que puede tomar A

$$P(B) = \sum_i P(B|A_i).P(A_i)P(B) = \sum_i P(B, A_i)$$



Probabilidad Total y Teorema de Bayes

Sean $A_1, ...A_i$ los valores que puede tomar A

$$P(A = A_i|B) = \frac{P(B, A_i)}{P(B)} = \frac{P(B|A_i).P(A_i)}{\sum_j P(B|A_j).P(A_j)}$$

El término $\sum_j P(B|A_j).P(A_j)$ se conoce como *función de partición* y es constante con respecto a los valores de A.

$$Z(A, B) = \sum_j P(B|A_j).P(A_j) = cte$$

Outline

Modelos Generativos

Clasificadores Generativos

Buscan aproximar la distribución de probabilidad $P(X, Y)$ que genere o imite el comportamiento de la data de entrenamiento.

- Se puede redefinir $P(X, Y)$ como $P(X, Y) = P(X|Y).P(Y)$
- Un clasificador generativo estima las distribuciones:
  - $P(Y)$: Prior de clases
  - $P(X|Y)$: Condicionales de clases

Clasificadores Generativos

- Se asume que la data se genera de acuerdo al proceso (independiente para cada muestra i):
  - Se muestrea una clase de la distribución Prior de clases: $y^i \sim P(Y)$
  - Se muestra una entrada de la correspondiente distribución Condicional de Clase: $x^i \sim P(X|Y = y^i)$

Outline

Clasificador Naive Bayes

Si se conociera la *verdadera* distribución $P(X, Y)$, el mejor posible clasificador (llamado Bayes óptimo), estaría definido por:

$$\hat{y} = argmax_{y \in Y} P(y|x)$$
$$\hat{y} = argmax_{y \in Y} \frac{P(x, y)}{P(x)}$$
$$\hat{y} = argmax_{y \in Y} \frac{P(x|y).P(y)}{P(x)}$$
$$\hat{y} = argmax_{y \in Y} P(x|y).P(y)$$

Donde se asume que $P(x)$ es constante con respecto de y.

Clasificador Naive Bayes: Deducción

Sea $L(\hat{y}, y)$ la función de costo, el clasificador minimiza el costo esperado:

$$\hat{y} = argmin_y \mathbb{E}[L(\hat{y}, y)|X]$$

Donde

$$\mathbb{E}[L(\hat{y}, y)|X] = \sum_{y'} L(\hat{y}, y').P(y'|x)$$

Si $L(\hat{y}, y)$ es el error 0/1 (Hinge Loss), se tiene

$$\hat{y} = argmin_{y \in Y} \sum_{y' \in Y} L(\hat{y}, y').P(y'|x)$$

$$\hat{y} = argmin_{y \in Y}(1 - P(y|x))$$

$$\hat{y} = argmax_{y \in Y} P(y|x)$$

Clasificador Naive Bayes

- Debido a que se maximiza el *posterior*, el método de inferencia se denomida *Maximum a Posteriori*
- Si se considera $P(y)$ constante para todo y, el método se denomina *Maximum Likelihood*

$$\hat{y}_{ML} = argmax_{y \in Y} P(x|y)$$

- También es común plantear

$$\hat{y}_{MAP} = argmax_{y \in Y} log(P(x|y)) + log(P(y))$$
$$\hat{y}_{ML} = argmax_{y \in Y} log(P(x|y))$$

Entrenamiento e Inferencia

- **Entrenamiento:** Estimación de las distribuciones $P(Y)$ y $P(X|Y)$ usando el dataset $D = X, Y$.
  Se obtienen $\hat{P}(Y)$ y $\hat{P}(X|Y)$
- **Inferencia o Decoding:** Dada una nueva entrada x, predecir de acuerdo a:

$$\hat{y} = argmax_{y \in Y} \hat{P}(y).\hat{P}(x|y)$$

Naive Bayes: Consideraciones

Naive Bayes considera que las entradas $x^1, x^2, ..., x^N$ son *condicionalmente independientes dada la clase*

$$P(X|Y) = \prod_{i=1}^{N} P(X^i|Y)$$

- Esta consideración reduce el número de parámetros de $O(exp(N))$ a $O(N)$
- La estimación de $\hat{P}(X|Y)$ se hace más simple y eficiente para valores de N grandes.
- Reduce el riesgo de sobre-ajuste (over-fitting).
- Puede incrementar el riesgo de sub-ajuste (under-fitting) si la consideración en muy simplista para el problema.

Outline

Regularización en modelos generativos

Smoothing o suavizado

Mover masa de probabilidad de los parámetros con más evidencia hacia los parámetros con menos evidencia.

- Si un parámetro de la distribución condicional de clases ($P(X|Y)$ o likelihood) no fue visto durante entrenamiento, su cuenta será cero.
- El *smoothing* agrega cuentas a numerador y denominador para evitar divisiones entre ceros.
- Conocido como *additive smoothing*, *Laplace smoothing*
- Equivalente a definir la *evidencia* $P(X)$ como una distribución uniforme y usar *Maximum a Posteriori* en vez de *Maximum Likelihood*

Multinomial Naive Bayes

- Objetivo: clasificar un documento usando los tokens como características
- Sea $V = w_1, .., w_v$ el vocabulario, $Y = y_1, ..., y_C$ el conjunto de clases de los documentos.
- Representación del documento: $bag - of - words$
  El orden de los tokens es ignorado, se considera solo su frencuencia en el documento
- Se asocia a cada clase una distribución multinomial de tokens

Multinomial Naive Bayes

- Por simplicidad, sea L la longitud de todos los documentos.
- El proceso de generación de cada documento viene a ser:
  - $y \sim P(Y)$
  - Se genera cada token w_j en el documento x en forma secuencial $w_j \sim P(w_j|y)$
- La probabilidad de generar un documento entonces es:

$$P(x|y) = \prod_{l=1}^{L} P(w_l|y) = \prod_{j=1}^{V} P(w_j|y)^{n_j(x)}$$

Donde $n_j(x)$: frecuencia de token w_j en documento x

Multinomial Naive Bayes: Parámetros

- Se asume que los tokens son independientes dadas las clases (consideración de *Naive Bayes*)
- Los parámetros a estimar son:

$$\hat{P}(y_c) = \frac{|N_C|}{N}$$

$$\hat{P}(w_j|y_c) = \frac{\sum_{m \in N_c} n_j(x^m)}{\sum_{i=1}^{V} \sum_{m \in N_c} n_i(x^m)}$$

Donde N_c son los indices de los documentos de la clase c

Multinomial Naive Bayes: Parámetros

El smoothing se agrega solo al *likelihood* $P(w_j|y_c)$

$$\hat{P}(w_j|y_c) = \frac{\alpha + \sum_{m \in N_c} n_j(x^m)}{\alpha * V + \sum_{i=1}^{V} \sum_{m \in N_c} n_i(x^m)}$$

Donde V es el número total de tokens, y α es el parámetro de smoothing.